Technical Guide

Support for Allied Telesis Enterprise MIBs in AlliedWare Plus™

Feature Overview and Configuration Guide

Introduction

This guide describes SNMP Management Information Bases (MIBs) and managed objects supported by the AlliedWare Plus™ Operating System. The following topics are covered:

- "Allied Telesis Enterprise MIB" on page 7 describes the objects implemented in the Allied Telesis Enterprise MIB
- "Public MIBs" on page 140 describes the public MIBs supported by the AlliedWare Plus™ Operating System, and any variations from the standard implementation.

Contents

Products and software version that apply to this guide

This guide applies to all AlliedWare Plus™ products, running version **5.4.5-1** or later.

Related documents

The following documents give more information about the MIBs and SNMP on AlliedWare Plus products:

- the SNMP Feature Overview and Configuration Guide
- The product's Command Reference

These documents are available from the above links on our website at alliedtelesis.com.

About MIBs

A MIB is a collection of managed objects organized into a tree-like hierarchy of nodes in which the managed objects form the leaves. Within the tree, each node is identified by a non-negative integer identifier that is unique among the node's siblings. The address, or object identifier, of any node within the tree is expressed as a series of dot-delimited node identifiers that trace the path from the root of the tree to the node. For example, the object identifier for the sysDescr object is 1.3.6.1.2.1.1.1.

Note: This guide does not indicate which MIB objects are not-accessible (and therefore cannot be queried directly). Please consult the MIB files for that information.

About SNMP

A Network Management Station (NMS) uses a protocol known as Simple Network Management Protocol (SNMP) to query or change the values of objects in the MIB of managed devices.

A managed device uses SNMP to respond to queries from an NMS, and to send unsolicited alerts (traps) to an NMS in response to events.

Obtaining MIBs

You can download MIBs from the following locations:

Table 1: Obtaining MIBs

DOWNLOAD THIS MIB...	FROM THIS LOCATION...
Allied Telesis Enterprise MIB	The MIB files are available with the software files from the Support area at: www.alliedtelesis.com/support/software.
Public MIBs defined in RFCs	www.rfc-editor.org/rfc.html.
IANAifType-MIB	www.iana.org/assignments/ianaiftype-mib.

Loading MIBs

Individual MIBs define a portion of the total MIB for a device. For example, the MAU-MIB defines objects for managing IEEE 802.3 Medium Attachment Units (MAUs), and forms a sub-tree under mib-2 with the object identifier snmpDot3MauMgt (1.3.6.1.2.1.26).

All the objects within a MIB are assigned object identifiers relative to a parent object. Most MIBs import the object identifier of the parent object, along with other object identifiers, textual conventions, macros and syntax types from the MIBs where they are defined. This creates dependencies between MIBs.

Some network management stations and MIB compilers will generate errors if you load a MIB that depends on another MIB that has not already been loaded. To avoid these errors, we recommend that you load MIBs in the following order:

1. RFC 1212
   RFC 1239
   RFC 2257
   RFC 3410.
2. RFC1155-SMI (RFC 1155)
   SNMPv2-SMI (RFC 2578)
   SNMPv2-PDU (RFC 3416).
3. RFC1213-MIB (RFC 1213)
   RFC 1215
   SNMPv2-TC (RFC 2579)
   SNMPv2-CONF (RFC 2580).
4. IP-MIB (RFC 2011)
   TCP-MIB (RFC 2012)
   UDP-MIB (RFC 2013)
   IP-FORWARD-MIB (RFC 2096)
   SNMP-MPD-MIB (RFC 2572)
   RMON-MIB (RFC 2819)
   HCNUM-TC (RFC 2856)
   SNMP-FRAMEWORK-MIB (RFC 3411)
   SNMP-MPD-MIB (RFC 3412)
   SNMPv2-TM (RFC 3417)
   SNMPv2-MIB (RFC 3418)
   INET-ADDRESS-MIB (RFC 4001)
   IANAifType-MIB.
5. IF-MIB (RFC 2863)
   SNMP-TARGET-MIB (RFC 3413).
6. SNMP-COMMUNITY-MIB (RFC 2576)
   EtherLike-MIB (RFC 3635)
   MAU-MIB (RFC 3636)
   BRIDGE-MIB (RFC 4188)
   DISMAN-PING-MIB (RFC 4560)
   SNMP-NOTIFICATION-MIB (RFC 3413)

SNMP-PROXY-MIB (RFC 3413)
P-BRIDGE-MIB (RFC 2674)
Q-BRIDGE-MIB (RFC 2674)
RSTP-MIB (RFC 4318)
LLDP-MIB
LLDP-EXT-DOT1-MIB
LLDP-EXT-DOT3-MIB
LLDP-EXT-MED-MIB
POE-MIB
VRRPv3-MIB
AT-SMI-MIB.

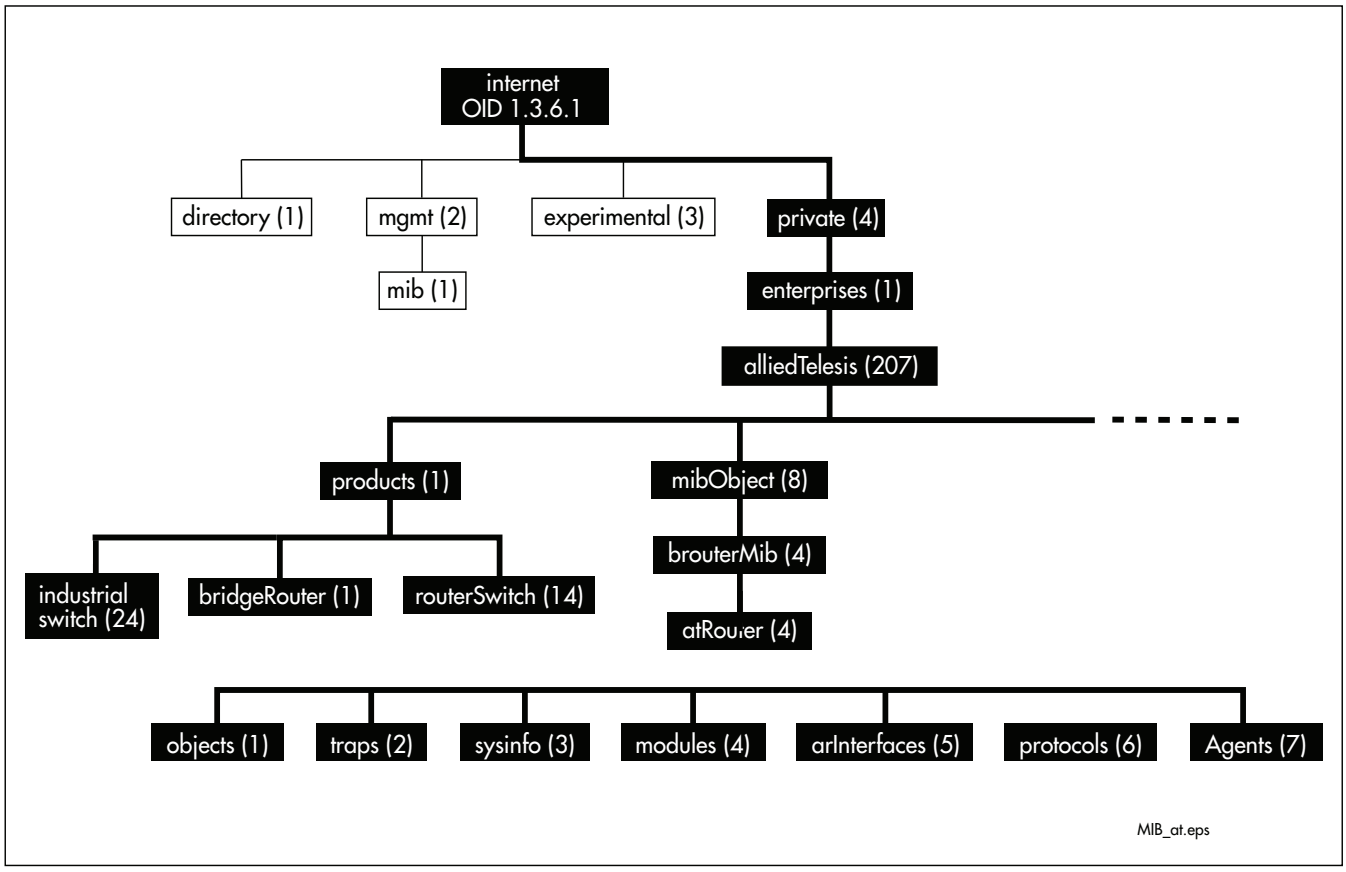
7. AT-PRODUCT-MIB
AT-BOARDS-MIB
AT-CHASSIS-MIB
AT-SYSINFO-MIB
AT-XEM-MIB
AT-ENVMONv2-MIB
AT-VCSTACK-MIB
AT-PORTINFO-MIB
AT-MIBVERSION-MIB
AT-VLAN-MIB
AT-USER-MIB
AT-RESOURCE-MIB
AT-LICENSE-MIB
AT-HHM-MIB
AT-LINKMON-MIB
AT-LINKTRAP-MIB
AT-ALMMON-MIB
AT-FIBER-MONITORING-MIB
AT-PLUGGABLES-DIAGNOSTICS-MIB
AT-TRIGGER-MIB
AT-LOOPPROTECT-MIB
AT-SETUP-MIB
AT-DNS-CLIENT-MIB
AT-NTP-MIB
AT-QOSv2-MIB
AT-PTP-MIB
AT-EPSRV2-MIB
AT-UDLD-MIB
AT-DHCPSN-MIB
AT-FILEV2-MIB
AT-LOG-MIB
AT-HHM-MIB
AT-ATMF-MIB
AT-G8032V2-MIB
AT-UFO-MIB
AT-MAC-NOTIFICATIONS-MIB

Allied Telesis Enterprise MIB

The Allied Telesis Enterprise MIB defines a portion of the MIB for managing Allied Telesis products and features that are not supported by public MIBs. Objects defined in this MIB reside in the private(4) subtree and have the object identifier **alliedTelesis** { enterprises 207 } OID 1.3.6.1.4.1.207.

This document describes only those portions of the Allied Telesis Enterprise MIB supported by the AlliedWare Plus™ Operating System. Figure 1 shows the structure of the Allied Telesis Enterprise MIB. Each component MIB is detailed in the following sections of this chapter.

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



AT-SMI-MIB the high level structure

AT-SMI-MIB defines the high-level structure and root objects of the Allied Telesis Enterprise MIB, see Table 2. The object identifier is **alliedTelesis** { enterprises 207 } OID 1.3.6.1.4.1.207. These objects are imported by other component MIBs of the Allied Telesis Enterprise MIB.

Table 2: AT Enterprise MIB - High Level Structure

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
alliedTelesis	{ enterprises 207 } 1.3.6.1.4.1.207	Root of the Allied Telesis Enterprise MIB under the private(4) node defined in RFC1155-SMI.
products	{ alliedTelesis 1 } 1.3.6.1.4.1.207.1	Sub-tree of all product OIDs. Described in “AT-PRODUCTS-MIB” on page 85.
bridgeRouter	{ products 1 } 1.3.6.1.4.1.207.1.1	Sub-tree of bridge product MIB objects (not applicable for AlliedWare Plus).
routerSwitch	{ products 14 } 1.3.6.1.4.1.207.1.2	Sub-tree for all router and switch product MIB objects.
industrialSwitch	{ products 24 } 1.3.6.1.4.1.207.1.24	Sub-tree for industrial switch product MIB objects.
mibObject	{ alliedTelesis 8 } 1.3.6.1.4.1.207.8	Sub-tree for all managed objects.
brouterMib	{ mibObject 4 } 1.3.6.1.4.1.207.8.4	Sub-tree of objects for managing bridges, routers, and switches.
atRouter	{ brouterMib 4 } 1.3.6.1.4.1.207.8.4.4	Sub-tree of objects for managing multiprotocol routers and switches.
objects	{ atRouter 1 } 1.3.6.1.4.1.207.8.4.4.1	Sub-tree of OIDs for boards, releases, interface types, and chips.
traps	{ atRouter 2 } 1.3.6.1.4.1.207.8.4.4.2	Sub-tree for generic traps (not applicable for AlliedWare Plus).
sysinfo	{ atRouter 3 } 1.3.6.1.4.1.207.8.4.4.3	Sub-tree of objects describing general system information.
modules	{ atRouter 4 } 1.3.6.1.4.1.207.8.4.4.4	Sub-tree of objects for monitoring and managing software features.
arInterfaces	{ atRouter 5 } 1.3.6.1.4.1.207.8.4.4.5	Sub-tree of objects describing boards, slots and physical interfaces.
protocols	{ atRouter 6 } 1.3.6.1.4.1.207.8.4.4.6	Sub-tree of OIDs for protocols.
atAgents	{ atRouter 7 } 1.3.6.1.4.1.207.8.4.4.7	Sub-tree of objects describing variations from standards.

AT-SMI-MIB objects listed by Object Identifier

Table 3 lists the major modules of the AT-SMI-MIB grouped by their object identifiers.

Table 3: AT-SMI-MIBs Listed by Object Identifier

MIB SECTION	OBJECT IDENTIFIER	DESCRIPTION
AT-SMI-MIB the high level structure	1.3.6.1.4.1.207	This section describes the structure of management information for the Allied Telesis Enterprise object, alliedTelesis { 1.3.6.1.4.1.207 }. See "AT-SMI-MIB the high level structure" on page 8.
AT-PRODUCTS-MIB	1.3.6.1.4.1.207.1	Object identifiers for Allied Telesis products. See "AT-PRODUCTS-MIB" on page 85.
AT-BOARDS-MIB	1.3.6.1.4.1.207.8.4.4.1.1	Object identifiers for boards, interface types, and chip sets. See "AT-BOARDS-MIB" on page 18.
AT-SYSINFO-MIB	1.3.6.1.4.1.207.8.4.4.3	Objects that describe generic system information and environmental monitoring. See "AT-SYSINFO-MIB" on page 121.
AT-XEM-MIB	1.3.6.1.4.1.207.8.4.4.3.11	Objects for managing XEMs (Expansion Modules). See "AT-XEM-MIB" on page 138.
AT-ENVMONv2-MIB	1.3.6.1.4.1.207.8.4.4.3.12	Objects and traps for monitoring fans, voltage rails, temperature sensors, and power supply bays. See "AT-ENVMONv2-MIB" on page 33.
AT-VCSTACK-MIB	1.3.6.1.4.1.207.8.4.4.3.13	Objects for managing Virtual Chassis Stacking (VCS). See "AT-VCSTACK-MIB" on page 131.
AT-PORTINFO-MIB	1.3.6.1.4.1.207.8.4.4.3.14	Objects for managing interface port objects such as transceivers. See "AT-PORTINFO-MIB" on page 86.
AT-MIBVERSION-MIB	1.3.6.1.4.1.207.8.4.4.3.15	Objects to display the last software release that contained changes to the support AT Enterprise MIB definition files. See "AT-MIBVERSION-MIB" on page 77.
AT-VLAN-MIB	1.3.6.1.4.1.207.8.4.4.3.16	Objects for managing VLANS. See "AT-VLAN-MIB" on page 136.
AT-USER-MIB	1.3.6.1.4.1.207.8.4.4.3.20	Objects for displaying information of users currently logged into a device, or configured in the Local User Data base of the device. See "AT-USER-MIB" on page 129.
AT-RESOURCE-MIB	1.3.6.1.4.1.207.8.4.4.3.21	Objects for displaying system hardware resource information. See "AT-RESOURCE-MIB" on page 112.
AT-LICENSE-MIB	1.3.6.1.4.1.207.8.4.4.3.22	Objects for managing software licenses on devices using AlliedWare Plus™ Operating System. See "AT-LICENSE-MIB" on page 60.
AT-CHASSIS-MIB	1.3.6.1.4.1.207.8.4.4.3.23	Objects for managing chassis based devices using AlliedWare Plus™ Operating System. See "AT-CHASSIS-MIB" on page 27.
AT-HHM-MIB	1.3.6.1.4.1.207.8.4.4.3.24	Objects for managing Hardware Health Monitoring (HHM). See "AT-HHM-MIB" on page 57.
AT-LINKTRAP-MIB	1.3.6.1.4.1.207.8.4.4.3.25	Objects for managing link traps. See "AT-LINKTRAP-MIB" on page 66.
AT-ALMMON-MIB	1.3.6.1.4.1.207.8.4.4.3.26	Objects for managing alarms. See "AT-ALMMON-MIB" on page 11.
AT-FIBER-MONITORING-MIB	1.3.6.1.4.1.207.8.4.4.3.27	Objects for managing fiber monitoring. See "AT-FIBER-MONITORING-MIB" on page 44.

Table 3: AT-SMI-MIBs Listed by Object Identifier (continued)

MIB SECTION	OBJECT IDENTIFIER	DESCRIPTION
AT-PLUGGABLE-DIAGNOSTICS-MIB	1.3.6.1.4.1.207.8.4.4.3.28	Objects for monitoring Optical Digital Diagnostics for pluggable devices. See “AT-PLUGGABLE-DIAGNOSTICS-MIB” on page 81.
AT-TRIGGER-MIB	1.3.6.1.4.1.207.8.4.4.4.53	Objects for managing triggers. See “AT-TRIGGER-MIB” on page 125.
AT-LOOPPROTECT-MIB	1.3.6.1.4.1.207.8.4.4.4.54	Objects for managing Allied Telesis Loop Protection. See “AT-LOOPPROTECT-MIB” on page 69.
AT-SETUP-MIB	1.3.6.1.4.1.207.8.4.4.4.500	Objects for managing software installation and configuration files. See “AT-SETUP-MIB” on page 113.
AT-DNS-MIB	1.3.6.1.4.1.207.8.4.4.4.501	Objects for managing Allied Telesis Domain Name System (DNS) Client Configuration. See “AT-DNS-MIB” on page 32.
AT-NTP-MIB	1.3.6.1.4.1.207.8.4.4.4.502	Objects for managing Allied Telesis Network Time Protocol (NTP) configuration. See “AT-NTP-MIB” on page 78.
AT-QOSv2-MIB	1.3.6.1.4.1.207.8.4.4.4.503	Objects for managing Allied Telesis Quality of Service (QoS) configuration. See “AT-QOSv2-MIB” on page 111.
AT-PTP-MIB	1.3.6.1.4.1.207.8.4.4.4.504	Objects for managing Allied Telesis Network Precision Time Protocol (PTP) configuration. See “AT-PTP-MIB” on page 91.
AT-EPSRv2-MIB	1.3.6.1.4.1.207.8.4.4.4.536	Objects for managing Allied Telesis Ethernet Automatic Protection Switching (EPSR). See “AT-EPSRv2-MIB” on page 41.
AT-DHCPSN-MIB	1.3.6.1.4.1.207.8.4.4.4.537	Objects for managing Allied Telesis Dynamic Host Configuration Protocol (DHCP) Snooping. See “AT-DHCPSN-MIB” on page 29.
AT-UDLD-MIB	1.3.6.1.4.1.207.8.4.4.4.550	Objects for managing Allied Telesis UniDirectional Link Detection (UDLD). See “AT-UDLD-MIB” on page 127.
AT-FILEv2-MIB	1.3.6.1.4.1.207.8.4.4.4.600	Objects for displaying and managing file content on local, stacked, and remote sources. See “AT-FILEv2-MIB” on page 46.
AT-LOG-MIB	1.3.6.1.4.1.207.8.4.4.4.601	Objects for listing log entries from the buffered and permanent logs. See “AT-LOG-MIB” on page 67.
AT-IP-MIB	1.3.6.1.4.1.207.8.4.4.4.602	Objects for Allied Telesis specific IP address management. See “AT-IP-MIB” on page 58.
AT-ATMF-MIB	1.3.6.1.4.1.207.8.4.4.4.603	Objects for managing ATMF. See “AT-ATMF-MIB” on page 13.
AT-G8032v2-MIB	1.3.6.1.4.1.207.8.4.4.4.604	Objects for managing Ethernet Protection Ring Switching (EPRS). See “AT-G8032v2-MIB” on page 56.
AT-UFO-MIB	1.3.6.1.4.1.207.8.4.4.4.605	Objects for Allied Telesis Upstream Forwarding Only (UFO) for private VLANs. See “AT-UFO-MIB” on page 128.
AT-LINKMON-MIB	1.3.6.1.4.1.207.8.4.4.4.606	Objects for Allied Telesis Link Monitoring for collecting and reporting link metrics. See “AT-LINKMON-MIB” on page 63.
AT-MAC-NOTIFICATION-MIB	1.3.6.1.4.1.207.8.4.4.4.607	Objects for Allied Telesis Link Monitoring for configuration of the MAC notification feature. See “AT-MAC-NOTIFICATION-MIB” on page 71.

Allied Telesis Enterprise MIBs in Alphabetical Order

AT-ALMMON-MIB

AT-ALMMON-MIB defines objects for managing and reporting device alarms, see Table 4. Objects in this group have the object identifier **atAlmMon** { sysinfo 26 } OID 1.3.6.1.4.1.207.8.4.4.3.26.

Table 4: Objects defined by the AT-ALMMON-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atAlmMon	{ sysinfo 26 } 1.3.6.1.4.1.207.8.4.4.3.26	The AT Alarm Monitoring v2 MIB for managing and reporting device alarms.
atAlmMonActionTable	{ atAlmMon 1 } 207.8.4.4.3.26.1	Table of information defining alarm monitoring inputs and consequent actions (i.e. fault LED and relay outputs), indexed by: ■ atAlmMonActionStackMemberId ■ atAlmMonActionIndex.
atAlmMonActionEntry	{ atAlmMonActionTable 1 } 207.8.4.4.3.26.1.1	A description and configuration of what to do for a specific monitored alarm.
atAlmMonActionStackMemberId	{ atAlmMonActionEntry 1 } 207.8.4.4.3.26.1.1.1	The index of the stack member of this alarm action. Read only.
atAlmMonActionIndex	{ atAlmMonActionEntry 2 } 207.8.4.4.3.26.1.1.2	The numeric identifier of this alarm action. Read only.
atAlmMonAlarmType	{ alAlmMonActionEntry 3 } 207.8.4.4.3.26.1.1.3	The type of alarm that this action monitors. Read only. Values can be: ■ externalPSU (1) ■ epsr (2) ■ contactInput (3) ■ portLinkDown (4) ■ loopDetect (5) ■ mainPse (6) ■ portPoeFailure (7) ■ temperature (8) ■ g8032 (9) ■ ufo (10)
atAlmMonAlarmTypeSelection	{ atAlmMonActionEntry 4 }	The 1-based index of the alarm of the particular type (as categorized by AlmMonAlarmType).
atAlmMonActionDescription	{ atAlmMonActionEntry 5 }	The description of this alarm monitoring entry. Read only.
atAlmMonActionUseRelay1	{ atAlmMonActionEntry 6 }	Indicates/controls whether or not this alarm monitor drives the first relay output. Read-write. Values can be: ■ Unused (1) ■ Used (2).
atAlmMonActionUseRelay2	{ alAlmMonActionEntry 7 }	Indicates/controls whether or not this alarm monitor drives the second relay output. Read-write. Values can be: ■ Unused (1) ■ Used (2).

Table 4: Objects defined by the AT-ALMMON-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atAlmMonActionUseRelay3	{ atAlmMonActionEntry 8 }	Indicates/controls whether or not this alarm monitor drives the third relay output. Read-write. Values can be: ■ Unused (1) ■ Used (2).
atAlmMonActionUseFaultLed	{ alAlmMonActionEntry 9 }	Indicates/controls whether or not this alarm monitor drives the fault LED. Read-write. Values can be: ■ Unused (1) ■ Used (2).
atAlmMonAbnormalState	{ atAlmMonActionEntry 10 }	Indicates/sets the abnormal (i.e., alarm active) state for a contact input. Only used for contactInput alarm monitors, ignored for all other types. Read-write. Values can be: ■ open (1) ■ closed (2).
atAlmMonActionState	{ atAlmMonActionEntry 11 }	Indicates the current state of this alarm monitor. Read only. Values can be: ■ Inactive (1) ■ Active (2).

AT-ATMF-MIB

The ATMF-MIB defines objects for managing the Allied Telesis Management Framework (AMF). Objects and triggers are shown diagrammatically in Figure 2 and Figure 3. Objects in this group have the object identifier **atmf** { modules 603 } OID 1.3.6.1.4.1.207.8.4.4.4.603.

Figure 2: The upper levels of the AT-ATMF-MIB sub-tree

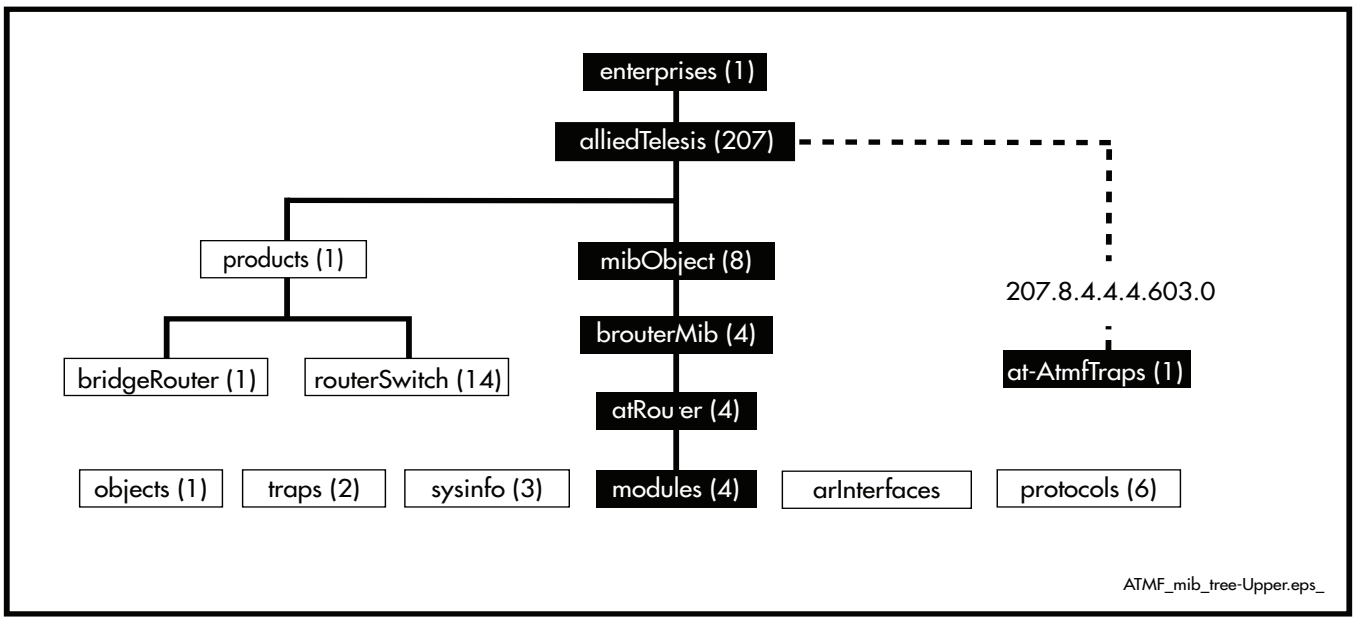


Figure 3: The lower levels of the AT-ATMF-MIB sub-tree

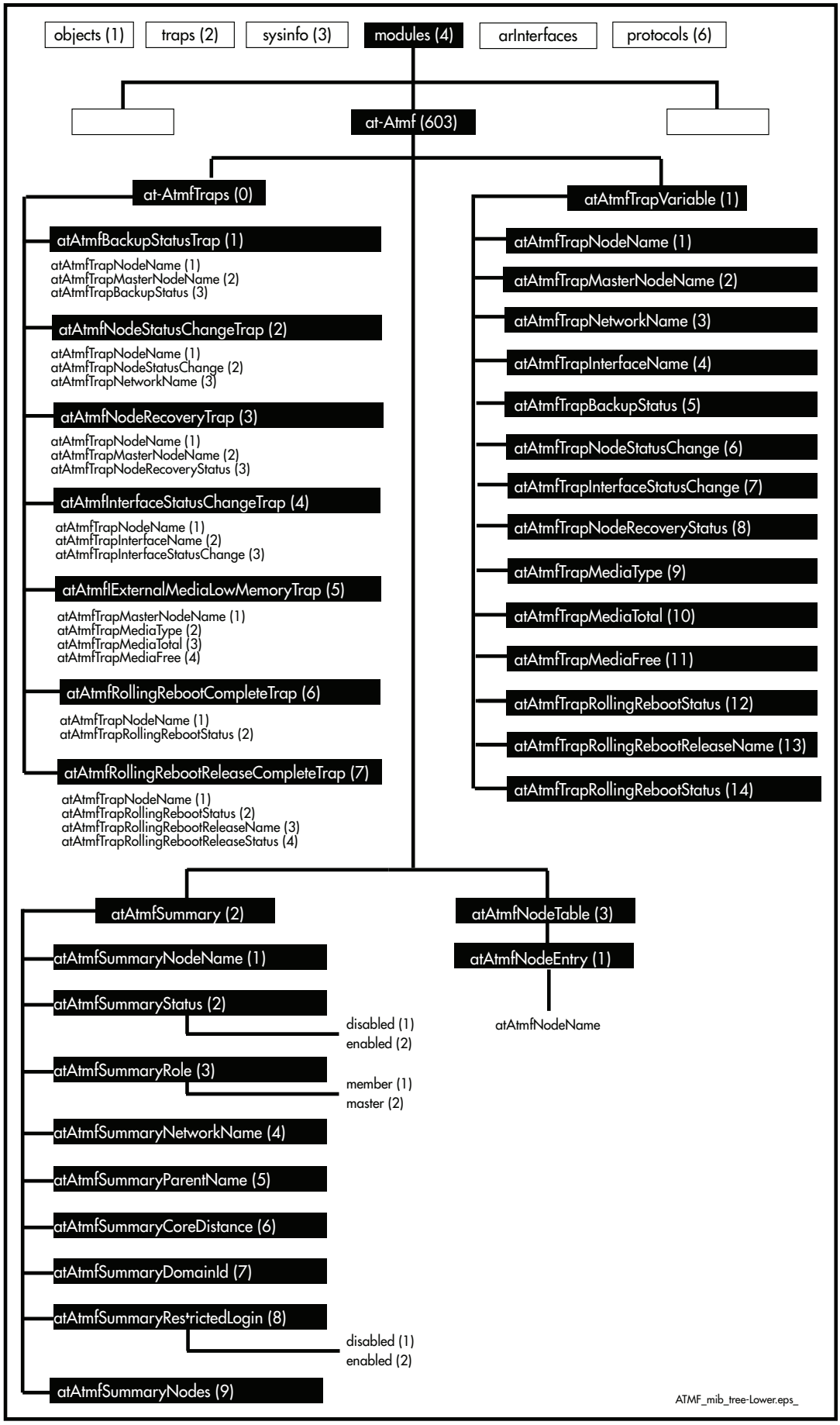


Table 5: AT-ATMF-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atmf	{ modules 603 } 1.3.6.1.4.1.207.8.4.4.4.603	Root of the Allied Telesis ATMF MIB under the private(4) node defined in RFC1155-SMI.
atAtmfTraps	{ atmf 0 } 1.3.6.1.4.1.207.8.4.4.4.603.0	Sub-tree of objects describing ATMF traps.
atAtmfBackupStatusTrap	{ atAtmfTraps 1 }	This trap is generated when an ATMF master attempts to backup a node's FLASH contents. It states whether the backup of an individual node, or all nodes, to a master node has passed or failed. Its objects are: ■ atAtmfTrapNodeName ■ atAtmfTrapMasterNodeName ■ atAtmfTrapBackupStatus.
atAtmfNodeStatusChangeTrap	{ atAtmfTraps 2 }	This trap is generated when an ATMF node joins or leaves the ATMF network. It states whether a node has <leftljoined> an ATMF network. Its objects are: ■ atAtmfTrapNodeName ■ atAtmfTrapNodeStatusChange ■ atAtmfTrapNetworkName.
atAtmfNodeRecoveryTrap	{ atAtmfTraps 3 }	This trap is generated when an attempt has been made to recover an ATMF node. It states whether an attempt to recover a node from the specified master has passed or failed. Its objects are: ■ atAtmfTrapNodeName ■ atAtmfTrapMasterNodeName ■ atAtmfTrapNodeRecoveryStatus.
atAtmfInterfaceStatusChangeTrap	{ atAtmfTraps 4 }	This trap is generated when an ATMF interface status change occurs. It states that an interface on a node has changed status to either blocking or forwarding. Its objects are: ■ atAtmfTrapNodeName ■ atAtmfTrapInterfaceName ■ atAtmfTrapInterfaceStatusChange.
atAtmfExternalMediaLowMemoryTrap	{ atAtmfTraps 5 }	This trap is generated when the available external storage on the ATMF master node falls below a nominated threshold. It states that the external USB or SD card storage on a master node has fallen below the designated threshold and specifies the total available memory <xxx MB> and the total free memory <xxx MB>. Its objects are: ■ atAtmfTrapMasterNodeName ■ atAtmfTrapMediaType ■ atAtmfTrapMediaTotal ■ atAtmfTrapMediaFree.

Table 5: AT-ATMF-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atAtmfRollingRebootCompleteTrap	{ atAtmfTraps 6 }	This trap is generated when the ATMF rolling reboot process has finished on a particular ATMF node. Nominally, it states that the ATMF rolling reboot, executed against the specified node, has returned a reboot status of either failed or passed. Its objects are: ■ atAtmfTrapNodeName ■ atAtmfTrapRollingRebootStatus.
atAtmfRollingRebootReleaseCompleteTrap	{ atAtmfTraps 7 }	This trap is generated when the ATMF rolling reboot process attempts to push a new software release to a specified ATMF node. Nominally, it states that the ATMF rolling reboot release process, executed from the specified node has returned a reboot status of either failed or passed, the name of the attempted release file and the release status of either “failed” or “passed”. Its objects are: ■ atAtmfTrapNodeName ■ atAtmfTrapRollingRebootStatus ■ atAtmfTrapRollingRebootReleaseName ■ atAtmfTrapRollingRebootReleaseStatus.
atAtmfTrapVariable	{ atmf 1 }	Sub-tree of objects describing ATMF traps.
atAtmfTrapNodeName	{ atAtmfTrapVariable 1 }	The ATMF trap node name.
atAtmfTrapMasterNodeName	{ atAtmfTrapVariable 2 }	The ATMF trap master node name.
atAtmfTrapNetworkName	{ atAtmfTrapVariable 3 }	The ATMF trap network name.
atAtmfTrapInterfaceName	{ atAtmfTrapVariable 4 }	The ATMF interface name, “Trap”.
atAtmfTrapBackupStatus	{ atAtmfTrapVariable 5 }	The status of the last trap backup attempt on either a specified ATMF node or all nodes in the ATMF network. Its objects are: ■ failed(1) ■ passed(2).
atAtmfTrapNodeStatusChange	{ atAtmfTrapVariable 6 }	An ATMF trap node has changed its status in the ATMF network. Its objects are: ■ left(1) ■ joined(2).
AtmfTrapInterfaceStatusChange	{ atAtmfTrapVariable 7 }	An ATMFtrap interface has changed its status. Its objects are: ■ blocking(1) ■ forwarding(2).
atAtmfTrapNodeRecoveryStatus	{ atAtmfTrapVariable 8 }	The status of the last recovery attempt. Its objects are: ■ failed(1) ■ passed(2).
atAtmfTrapMediaType	{ atAtmfTrapVariable 9 }	The media type resident on the ATMF node - USB or SD.
atAtmfTrapMediaTotal	{ atAtmfTrapVariable 10 }	The total memory available on the resident media, in MB.

Table 5: AT-ATMF-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atAtmfTrapMediaFree	{ atAtmfTrapVariable 11 }	The free memory available on the resident media, in MB. Each node has a maximum flash of 64MB.
atAtmfTrapRollingRebootStatus	{ atAtmfTrapVariable 12 }	The status of the last rolling reboot for a node. Its objects are: ■ failed(1) ■ passed(2).
atAtmfTrapRollingRebootReleaseName	{ atAtmfTrapVariable 13 }	The name of the last rolling reboot release
atAtmfTrapRollingRebootReleaseStatus	{ atAtmfTrapVariable 14 }	The release update status of the last rolling reboot for a node. Its objects are: ■ failed(1) ■ passed(2).
atAtmfTrapVariable	{ atmf 2 }	Sub-tree of objects describing ATMF summary table.
atAtmfSummaryNodeName	{ atAtmfSummary 1 }	The name assigned to a particular node
atAtmfSummaryStatus	{ atAtmfSummary 2 }	The Node's ATMF status.
atAtmfSummaryRole	{ atAtmfSummary 3 }	The role configured for this ATMF device, either Master or Member.
atAtmfSummaryNetworkName	{ atAtmfSummary 4 }	The ATMF network that a particular node belongs to.
atAtmfSummaryParentName	{ atAtmfSummary 5 }	The parent name of the node or 'none'.
atAtmfSummaryCoreDistance	{ atAtmfSummary 6 }	The ATMF core distance for this node.
atAtmfSummaryDomainId	{ atAtmfSummary 7 }	The ATMF domain Id for this node.
atAtmfSummaryRestrictedLogin	{ atAtmfSummary 8 }	The login for this ATMF device is restricted to only those devices that are designated ATMF Masters. Its objects are: ■ isabled(1) ■ enabled(2.)
atAtmfSummaryNodes	{ atAtmfSummary 9 }	The number of ATMF nodes known to this device.
atAtmfNodeTable	{ atmf 3 }	Sub-tree of objects describing ATMF Nodes.
atAtmfNodeName	{ atAtmfNodeTable 1 }	The name assigned to a particular node.
atAtmfControllerAreaTable	{ atmf 4 }	Sub-tree of objects describing the configured areas available to the ATMF controller.
atAtmfControllerAreaEntry	{ atAtmfControllerAreaTable 1 }	The area identifier assigned to a particular controller area.
atAtmfControllerAreaName	{ atAtmfControllerAreaEntry 2 }	The name assigned to a particular controller area.
atAtmfControllerAreaStatus	{ atAtmfControllerAreaEntry 3 }	The reachability status of a particular controller area.
atAtmfControllerAreaMemberCount	{ atAtmfControllerAreaEntry 4 }	The number of member nodes available in a particular controller area.

AT-BOARDS-MIB

AT-BOARDS-MIB defines object identifiers for components of Allied Telesis products—base CPU and expansion boards, interface types, and chip sets. Objects in this MIB have the object identifier **objects** { atRouter 1 } OID 1.3.6.1.4.1.207.8.4.4.1, and are organized into the following groups:

- Base CPU and expansion boards, see Table 6. These object identifiers are for use with the hrDeviceID object in the Host Resources MIB (see "Public MIBs" on page 140).
- Interface types, see Table 7.
- Chip sets, see Table 8.

Table 6: Object identifiers for base CPU and expansion boards

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
boards	{ objects 1 } 1.3.6.1.4.1.207.8.4.4.1.1	A subtree beneath which board ids are assigned.
pprx90024XT	{ boards 271 }	x900-24XT, Enhanced Gigabit Layer 3+ Expandable Switch, 24 x 10/100/1000BASE-T copper ports (RJ-45 connectors), 2 x 20 Gigabit expansion bays.
pprx90024XS	{ boards 272 }	x900-24XS, Enhanced Gigabit Layer 3+ Expandable Switch, 24 x 10/100/1000BASE-T copper ports (RJ-45 connectors), 2 x 20 Gigabit expansion bays.
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 10/100/1000BASE-T copper ports (RJ-45 connectors).
pprAtXum12TiN	{ boards 280 }	XEM-12T-N, Expansion Module, 12 x 10/100/1000BASE-T copper ports (RJ-45 connectors), NEBS compliant.
pprx90024XTN	{ boards 281 }	x900-24XT, Enhanced Gigabit Layer 3+ Expandable Switch, 24 x 10/100/1000BASE-T copper ports (RJ-45 connectors), 2 x 20 Gigabit expansion bays, NEBS compliant.
pprSwitchBladex908	{ boards 282 }	SwitchBlade x908, 8 Slot Layer 3 Switch Chassis.
pprXemStk	{ boards 286 }	AT-XEM-STK, 2 x high speed stacking ports.
pprx90012XTS	{ boards 288 }	AT-x900-12XT/S, Advanced Gigabit Layer 3+ Expandable Switch, 12 x combo ports (10/100/1000BASE-T copper or SFP), 1 x 30Gbps expansion bay.
pprAt9524TS	{ boards 290 }	x600-24Ts/XP, 24 x 10/100/1000BASE-T ports (RJ45 connectors), 4 x SFP (combo) ports.
pprAt9524TSXP	{ boards 291 }	x600-24Ts/XP, 24 x 10/100/1000BASE-T ports (RJ45 connectors), 4 x SFP (combo) ports, 2 x XFP ports.
pprAt9548TS	{ boards 294 }	x600-44Ts, 44 x 10/100/1000BASE-T ports, 4 x SFP ports.
pprAt9548TSXP	{ boards 295 }	x600-44Ts/XP, 44 x 10/100/1000BASE-T ports, 4 x SFP ports, 2 x XFP ports.
pprXem2XP	{ boards 306 }	XEM-2XP, Expansion Module, 2 x 10Gbe XFP port.
pprATStackXG	{ boards 307 }	x600, Expansion Module, Stacking.

Table 6: Object identifiers for base CPU and expansion boards (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
pprATEMXP	{ boards 308 }	x600, Expansion Module, 2 x 10G XFP ports.
pprATLBM	{ boards 309 }	x600, Expansion Module, loopback.
pprAtSBx8112	{ boards 316 }	AT-SBx8112, SwitchBlade x8112 chassis.
pprAtSBx81CFC400	{ boards 317 }	AT-SBx81CFC, Control Fabric Card for SwitchBlade x8112.
pprAtSBx81GP24	{ boards 318 }	AT-SBx81GP24, 24 x 1G PoE line card.
pprAtSBxPWRSYSAC	{ boards 320 }	AT-SBxPWR SYS/AC, system power supply unit for the SwitchBlade x8112 (AC input).
pprAtSBxPWRPOEAC	{ boards 321 }	AT-SBxPWR POE/AC, PoE power supply unit for the SwitchBlade x8112 (AC input).
pprAtSBxFAN12	{ boards 322 }	AT-SBxFAN12, fan tray for the SwitchBlade x8112.
pprAtPWR05DC	{ boards 323 }	AT-PWR05, DC power supply unit for SwitchBlade x908.
pprXem2XT	{ boards 325 }	XEM-2XT, Expansion Module, 2 x 10Gbe copper XEM port.
pprx60024TSPOE	{ boards 326 }	x600-24Ts-POE, 24 x 10/100/1000BASE-T PoE ports (RJ45 connectors), 4 x SFP (combo) ports.
pprx60024TSPOEPLUS	{ boards 327 }	x600-24Ts-POE+, 24 x 10/100/1000BASE-T PoE+ ports (RJ45 connectors), 4 x SFP (combo) ports.
pprx61048TsXPOEPlus	{ boards 331 }	x610-48Ts/X-POE+, 48 x 10/100/1000BASE-T PoE+ ports (RJ45 connectors), 2 x SFP (combo) ports, 2 x SFP+ ports.
pprx61048TsPOEPlus	{ boards 332 }	x610-48Ts-POE+, 48 x 10/100/1000BASE-T PoE+ ports (RJ45 connectors), 4 x SFP (combo) ports.
pprx61024TsXPOEPlus	{ boards 333 }	x610-24Ts/X-POE+, 24 x 10/100/1000BASE-T PoE+ ports (RJ45 connectors), 4 x SFP (combo) ports, 2 x SFP+ ports.
pprx61024TsPOEPlus	{ boards 334 }	x610-24Ts-POE+, 24 x 10/100/1000BASE-T PoE+ ports (RJ45 connectors), 4 x SFP (combo) ports.
pprPWR800	{ boards 336 }	AT-PWR800, 800W power supply unit.
pprPWR1200	{ boards 337 }	AT-PWR1200, 1200W power supply unit.
pprPWR250	{ boards 338 }	AT-PWR250, 250W power supply unit.
pprx61048TsX	{ boards 339 }	AT-x610-48Ts/X, 48 x 10/100/1000BASE-T ports (RJ45 connectors), 2 x SFP (combo) ports, 2 x SFP+ ports.
pprx61048Ts	{ boards 340 }	AT-x610-48Ts, 48 x 10/100/1000BASE-T ports (RJ45 connectors), 4 x SFP (combo) ports.
pprx61024TsX	{ boards 341 }	AT-x610-24Ts/X, 24 x 10/100/1000BASE-T ports (RJ45 connectors), 4 x SFP (combo) ports, 2 x SFP+ ports.
pprx61024Ts	{ boards 342 }	AT-x610-24Ts, 24 x 10/100/1000BASE-T ports (RJ45 connectors), 4 x SFP (combo) ports.
pprx61024SPX	{ boards 343 }	AT-x610-24SPs/X, 20 x 10/100/1000BASE SFP ports, 4 x SFP (combo) ports.
pprXem2XS	{ boards 350 }	AT-XEM-2XS, Expansion Module, 2 x 10GbE (SFP+) ports.
pprPWR250DC	{ boards 351 }	AT-PWR250DC, 250W DC power supply unit.
pprAtSBx81GT24	{ boards 352 }	AT-SBx81GT24, 24 x 1G copper line card.
pprAtSBx81GS24a	{ boards 353 }	AT-SBx81GS24a, 24 x 1G SFP line card.

Table 6: Object identifiers for base CPU and expansion boards (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
pprAtSBx81XS6	{ boards 354 }	AT-SBx81XS6, 6 x 10G SFP+ line card.
pprXem24T	{ boards 356 }	AT-XEM-24T, Expansion Module, 24 x 10/100/1000T (RJ Point 5) ports.
pprXem12Tv2	{ boards 358 }	AT-XEM-12Tv2, Expansion Module, 12 x 10/100/1000T (RJ-45) ports.
pprXem12Sv2	{ boards 359 }	AT-XEM-12Sv2, Expansion Module, 12 x 10/100/1000X SFP ports.
pprx2109GT	{ boards 367 }	AT-x210-9GT, 8xGigbit, 1xSFP/T.
pprx21016GT	{ boards 368 }	AT-x210-16GT, 14xGigbit, 2xcombo SFP/T.
pprx21024GT	{ boards 369 }	AT-x210-24GT, 20xGigbit, 4xcombo SFP/T.
pprx51028GTX	{ boards 370 }	AT-x510-28GTX, 24 10/100/1000Base-T ports and four 10Gb/s SFP+ ports.
pprx51028GPX	{ boards 371 }	AT-x510-28GPX, 24 10/100/1000Base-T ports, four 10 Gb/s SFP+ ports and PSE function available on pins 1/2 and 3/6 (Mode A) of every copper port.
pprx51028GSX	{ boards 372 }	AT-x510-28GSX, 24 100/1000 SFP ports and four 10 Gb/s SFP+ ports.
pprx51052GTX	{ boards 373 }	AT-x510-52GTX, 48 10/100/1000Base-T ports and four 10 Gb/s SFP+ ports.
pprx51052GPX	{ boards 374 }	AT-x510-52GPX, 48 10/100/1000Base-T ports, four 10 Gb/s SFP+ ports and PSE function available on pins 1/2 and 3/6 (Mode A) of every copper port.
pprAtSBx8106	{ boards 375 }	AT-SBx8106, SwitchBlade x8106 chassis.
pprAtSBxFAN06	{ boards 376 }	AT-SBxFAN06, fan tray for the SwitchBlade x8106.
pprAtSBx81CFC960	{ boards 377 }	AT-SBx81CFC960, Control Fabric Card for SwitchBlade x8100 Series chassis, four 10GbE SFP+ ports.
pprSBx81GT24a	{ boards 378 }	AT-SBx81GT24, line card.
pprSBx81GP24a	{ boards 379 }	AT-SBx81GP24, PoE+ line card.
pprSBx81CFC960v2	{ boards 380 }	AT-SBx81CFC960v2 Control Fabric Card for SwitchBlade x8100 Series chassis, four 10GbE SFP+ ports.
pprAtSBx81GT40	{ boards 381 }	AT-SBx81GT40, RJ point five line card.
pprSBx81XS16	{ boards 382 }	AT-SBx81XS16, SFP+ line card.
pprPWR100R	{ boards 384 }	AT-PWR100R, 100W power supply unit.
pprPWR250DCR	{ boards 385 }	AT-PWR250R-DC, 250W DC power supply unit.
pprx510DP52GTX	{ boards 386 }	AT-x510DP-52GTX, 48 10/100/1000Base-T ports and four 10 Gb/s SFP+ ports.
pprxIX528GPX	{ boards 387 }	AT-IX5-28GPX, 24 10/100/1000Base-T ports, four 10 Gb/s SFP+ ports and PSE function available on pins 1/2 and 3/6 (Mode A) of every copper port.
pprx93028GTX	{ boards 388 }	AT-x930-28GTX, 24 1000BASE-T ports, 4 x 10 Gb/s SFP+ ports.
pprx93028GPX	{ boards 389 }	AT-x930-28GPX, 24 1000BASE-T ports, 4 x 10 Gb/s SFP+ ports.
pprx93028GSTX	{ boards 390 }	AT-x930-28GSTX, 24 1000BASE-T ports (combo), 24 x 1000 SFP ports (combo), 4 x 10 Gb/s SFP+ ports.
pprx93052GTX	{ boards 391 }	AT-x930-52GTX, 48 10/100/1000BASE-T ports, 4 x 10 Gb/s SFP+ ports.

Table 6: Object identifiers for base CPU and expansion boards (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
pprx93052GPX	{ boards 392 }	AT-x930-52GPX, 48 10/100/1000BASE-T ports, 4 x 10 Gb/s SFP+ ports.
pprx31026FT	{ boards 393 }	AT-x310-26FT, 24 10/100 Base-T ports, two 10/100/1000Base-T ports, and four 10 Gb/s SFP+ ports.
pprx31050FT	{ boards 394 }	AT-x310-50FT, 48 10/100 Base-T ports, two 10/100/1000Base-T ports, and four 10 Gb/s SFP+ ports.
pprx31026FP	{ boards 395 }	AT-x310-26FP, 24 10/100 Base-T ports, two 10/100/1000Base-T ports, four 10 Gb/s SFP+ ports, plus PSE function available on pins 1/2 and 3/6 (Mode A) of every copper port.
pprx31050FP	{ boards 396 }	AT-x310-50FP, 48 10/100 Base-T ports, two 10/100/1000Base-T ports, four 10 Gb/s SFP+ ports, plus PSE function available on pins 1/2 and 3/6 (Mode A) of every copper port.
pprx23010GP	{ boards 405 }	AT-x230-10GP, 8 10/100/1000BASE-RJ-45 ports, 2 1000Base SFP ports.
pprx23018GP	{ boards 406 }	AT-x 230-18GP, 16 10/100/1000BASE RJ-45 ports and 2 1000BASE SFP ports.
pprx23028GP	{ boards 407 }	AT-x 230-28GP, 24 10/100/1000BASE RJ-45 ports and 4 1000BASE SFP ports.
pprIE2006GT	{ boards 410 }	IE200-6GT, L2+ managed industrial Switch with 4 x 10/100/1000T LAN ports and 2 x SFP uplinks (100/1000X). **Note:** that this is a single board device.
pprIE2006GP	{ boards 411 }	IE200-6GP, L2+ managed industrial Switch with 4 x 10/100/1000T LAN ports (with 802.3at PoE+) and 2 x SFP uplinks (100/1000X). **Note:** that this is a single board device.
pprIE2006GPW	{ boards 412 }	IE200-6GPW, L2+ managed industrial Switch with 4 x 10/100/1000T LAN ports (with 802.3at PoE+) and 2 x SFP uplinks (100/1000X) and 802.11bgn wireless. **Note:** that this is a single board device.
pprdc2552xs	{ boards 414 }	AT-DC2552XS/L3, Stackable 10 Gigabit Layer 3 switch with 48-port SFP+ slot, 4-port QSFP slot.
pprATStackQS	{ boards 419 }	AT-StackQS, 2 x QSFP+ expansion module, 40GBPS 2.5µs.
pprx51028GSXDC	{ boards 421 }	AT-x510-28GSX-80? 24 SFP ports, 4 SFP+ ports.
pprIE51028GSX	{ boards 422 }	AT-x510-28GSX, 24 SFP ports, 4 SFP+ ports.
pprAR3050S	{ boards 423 }	AT-AR3050S, Next-Generation Firewall, 2 x 1000X SFP / 2 x 10/100/1000T RJ-45 combo.
pprAR4050S	{ boards 426 }	AT-AR4050S, Next-Generation Firewall, 2 x 1000X SFP / 2 x 10/100/1000T RJ-45 combo.
pprIE2006FT	{ boards 429 }	AT-IE200-6FT, 4 × 10/100TX ports and 2 × 100/1000X SFP Industrial switch.
pprIE2006FP	{ boards 430 }	AT-IE200-6FP, 4 × 10/100TX ports (PoE+ support) and 2 × 100/1000X SFP Industrial switch.
pprx510DP28GTX	{ boards 431 }	AT-x510DP-28GTX, 24 10/100/1000BASE RJ-45 ports, 4 x SFP+ ports.
pprx510L28GT	{ boards 432 }	AT-x510L-28GT, 24 x 10/100/1000BASE RJ-45 ports, 4 x SFP+ ports.
pprx510L28GP	{ boards 434 }	AT-x510L-28GP, 24 x 10/100/1000BASE RJ-45 ports, 4 x SFP+ ports.

Table 6: Object identifiers for base CPU and expansion boards (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
pprx510L52GP	{ boards 435 }	AT-x510L-52GT, 48 x 10/100/1000BASE RJ-45 ports, 4 x SFP+ ports.
pprIE30012GT	{ boards 438 }	Industrial Layer 3 switch Gigabit 12-port IE300-12GT.
pprIE30012GP	{ boards 439 }	Industrial Ethernet Layer 3 Switch Gigabit 12-port IE300-12GT.
pprAtGS924MX	{ boards 443 }	AT-GS924MX, 24 x 10/100/1000T ports, 2 combo ports (10/100/1000T or 100/1000X SFP) and 2 X 10G SFP+ Stacking/User ports.
pprAtGS924MPX	{ boards 444 }	AT-GS924MPX, 24 x 10/100/1000T POE+ ports, 2 combo ports (10/100/1000T or 100/1000X SFP) and 2 X 10G SFP+ Stacking/User ports.
pprAtGS948MX	{ boards 445 }	AT-GS948MX, 48 x 10/100/1000T ports, 2 combo ports (10/100/1000T or 100/1000X SFP) and 2 X 10G SFP+ Stacking/User ports.
pprAtGS948MPX	{ boards 446 }	AT-GS948MPX, 48 x 10/100/1000T POE+ ports, 2 combo ports (10/100/1000T or 100/1000X SFP) and 2 X 10G SFP+ Stacking/User ports.
pprAR2050V	{ boards 454 }	AT-AR2050V 1 x GE WAN and 4 x 10/100/1000 LAN.
pprAR2010V	{ boards 455 }	AT-AR2010V 2 x 10/100/1000T RJ-45.
pprSH23010GP	{ boards 463 }	AT-x230-10GP, 8 x 10/100/1000BASE RJ-45 ports, 2 x 10/100/1000BASE SFP ports and PoE+.
pprSH23018GP	{ boards 464 }	AT-x 230-18GP, 16 x 10/100/1000BASE RJ-45 ports, 2 x 10/100/1000BASE SFP ports and PoE+.
pprSH23028GP	{ boards 465 }	AT-x 230-28GP, 24 x 10/100/1000BASE RJ-45 ports, 2 x 10/100/1000BASE SFP ports and PoE+.
pprSH2109GT	{ boards 466 }	AT-x210-9GT, 8 x 10/100/1000BASE RJ-45 ports, 1 x 10/100/1000BASE SFP port.
pprSH21016GT	{ boards 467 }	AT-x210-16GT, 14 x 10/100/1000BASE RJ-45 ports, 2 x 10/100/1000BASE combo ports.
pprSH21024GT	{ boards 468 }	AT-x210-24GT, 20 x 10/100/1000BASE RJ-45 ports, 4 x 10/100/1000BASE combo ports.
pprSH31026FT	{ boards 469 }	AT-x310-26FT, 24 x 10/100BASE RJ-45 ports, 2 x 10/100/1000BASE (combo) ports and 2 x stacking ports.
pprAtXS916MXT	{ boards 456 }	AT-GS916M 14 × 10/100/1000T ports, 2 × Combo ports (2 × 10/100/1000T or 100FX/1000X ports) 1 standard AC power supply in a compact form factor.
pprSH51028GTX	{ boards 459 }	AT-x510-28GTX features 24 x 10/100/1000T ports and 4 x 10G/1G SFP+ uplink ports with two internal load-sharing AC power supplies.
pprSH51052GTX	{ boards 460 }	AT- x510-52GTX stackable Gigabit edge switch features 48 x 10/100/1000T ports and 4 x 10G/1G SFP+ uplink ports with two internal load-sharing AC power supplies.
pprSH51028GPX	{ boards 461 }	AT-x510-28GPX features 24 x 10/100/1000T PoE+ ports and 4 x 10G/1G SFP+ uplink ports.
pprSH51052GPX	{ boards 462 }	AT-x510-52GPX stackable Gigabit edge switch features 48 x 10/100/1000T PoE+ ports and 4 x 10G/1G SFP+ uplink ports.
pprSH31050FT	{ boards 470 }	AT-x310-50FT, 48 x 10/100BASE RJ-45 ports, 2 x 10/100/1000BASE (combo) ports and 2 x stacking ports.

Table 6: Object identifiers for base CPU and expansion boards (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
pprSH31026FP	{ boards 471 }	AT-x310-26FP, 24 x 10/100BASE RJ-45 ports, 2 x 10/100/1000BASE (combo) ports and 2 x stacking ports.
pprSH31050FP	{ boards 472 }	AT-x310-50FP, 48 x 10/100BASE RJ-45 ports, 2 x 10/100/1000BASE (combo) ports and 2 x stacking ports.
pprSH23010GT	{ boards 473 }	AT-x230-10GT L3 switch with 8 x 10/100/1000T ports and 2 x 100/1000X SFP ports.
pprSH23018GT	{ boards 474 }	AT-x230-18GP L3 switch with 16 x 10/100/1000T PoE ports and 2 x 100/1000X SFP ports.
pprSH23028GT	{ boards 475 }	AT-x230-18GT L3 switch with 16 x 10/100/1000T ports and 2 x 100/1000X SFP ports.
pprAtFS980M9	{ boards 476 }	AT-FS980M/9, 8 x 10/100TX ports and 1 combo (10/100/1000T or 100/1000X SFP) uplink port.
pprAtFS980M9PS	{ boards 477 }	AT-FS980M/9PS, 8 x 10/100TX PoE+ ports and 1 combo (10/100/1000T or 100/1000X SFP) uplink port.
pprAtFS980M18	{ boards 478 }	AT-FS980M/18, 16 x 10/100TX ports and 2 combo (10/100/1000T or 100/1000X SFP) uplink ports.
pprAtFS980M18PS	{ boards 479 }	AT-FS980M/18PS, 16 x 10/100TX PoE+ ports and 2 combo (10/100/1000T or 100/1000X SFP) uplink ports.
pprAtFS980M28	{ boards 480 }	AT-FS980M/28, 24 x 10/100TX ports and 4 x 100/1000X SFP uplink/stacking ports.
pprAtFS980M28PS	{ boards 481 }	AT-FS980M/28PS, 24 x 10/100TX PoE+ ports and 4 x 100/1000X SFP uplink/stacking ports.
pprAtFS980M52	{ boards 482 }	AT-FS980M/52, 48 x 10/100TX ports and 4 x 100/1000X SFP uplink/stacking ports.
pprAtFS980M52PS	{ boards 483 }	AT-FS980M/52PS, 48 x 10/100TX PoE+ ports and 4 x 100/1000X SFP uplink/stacking ports.
pprSBx908G2	{ boards 484 }	AT-SBx908 GEN 2, High capacity Layer 3+ modular switch chassis with 8 x high speed expansion bays.
pprAtFan08	{ boards 486 }	AT-FAN08, Hot-swappable fan module for the SwitchBlade x908 Generation 2.
pprAtXem2QS4	{ boards 487 }	AT-XEM2-4QS, 4 x 40G QSFP+ ports.
pprAtXem2XS12	{ boards 488 }	AT-XEM2-12XS, 12 x 1G/10G SFP+ ports.
pprAtXem2XT12	{ boards 489 }	AT-XEM2-12XT, 12 x 1G/10G RJ45 ports.
pprx55018XTQ	{ boards 491 }	AT- x550-18XTQ, stackable 10 Gigabit Layer 3 switch is a high-performing and feature-rich solution, with 16 x 1G/10G copper Ethernet ports and 2 x 40G uplinks.
pprx55018XSQ	{ boards 492 }	AT-x550-18XSQ, stackable 10 Gigabit Layer 3 switch is a high-performing and feature-rich solution, with 16 x 1G/10G SFP+ ports and 2 x 40G uplinks.
pprx55018XSPQm	{ boards 493 }	AT-x550-18XSPQm, stackable 10 Gigabit Layer 3 switch is a high-performing and feature-rich solution, with 8 x 1G/10G SFP+ slots, 8 x 1G/2.5G/5G/10G PoE+ Ethernet ports, and 2 x 40G uplinks.
pprAtXem2CQ1	{ boards 494 }	AT-XEM2-1CQ, 1 x 100G QSFP28 port module.
pprx53028GTXm	{ boards 523 }	AT-x530-28GTXm, stackable Gigabit layer 3 switch is high-performing and feature-rich, with 24 x 100M/1G copper ports and 4 x 10G uplinks.

Table 6: Object identifiers for base CPU and expansion boards (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
pprx53028GPXm	{ boards 524 }	AT-x530-28GPXm, stackable Gigabit layer 3 switch is high-performing and feature-rich, with 24 x 100M/1G copper PoE+ ports and 4 x 10G uplinks.
pprAtGS970M28PS	{ boards 534 }	AT-GS970M/28PS, 24 x 10/100/1000T Gigabit PoE+ ports and 4 x 100/1000X SFP ports.
pprAtGS970M18PS	{ boards 535 }	AT-GS970M/18PS, Layer 3 managed switch 16 x 10/100/1000T Gigabit PoE+ ports and 2 x 100/1000X SFP ports.
pprAtGS970M10PS	{ boards 536 }	AT-GS970M/10PS, Layer 3 managed switch 8 x 10/100/1000T Gigabit PoE+ ports and 2 x 100/1000X SFP ports.
pprAtGS970M28	{ boards 537 }	AT-GS970M/28, Layer 3 managed switch 24 x 10/100/1000T Gigabit Ethernet ports and 4 x 100/1000X SFP ports.
pprAtGS970M18	{ boards 538 }	AT-GS970M/18, Layer 3 managed switch 16 x 10/100/1000T Gigabit Ethernet ports and 2 x 100/1000X SFP ports.
pprAtGS970M10	{ boards 539 }	AT-GS970M/10, Layer 3 managed switch 8 x 10/100/1000T Gigabit Ethernet ports and 2 x 100/1000X SFP ports.
pprAtIE34012GP	{ boards 540 }	AT-IE340-12GP-80, 8x 10/100/1000T, 4x 100/1000X SFP, Industrial Ethernet, Layer 3 Switch, PoE+ Support.
pprAtIE340L18GP	{ boards 541 }	AT-IE340L-18GP-80, 16x 10/100/1000T, 2x 1000X SFP, Industrial Ethernet, Layer 3 Switch, PoE+ Support.
pprAtIE34012GT	{ boards 542 }	AT-IE340-12GT-80, 8x 10/100/1000T, 4x 100/1000X SFP, Industrial Ethernet, Layer 3 Switch.
pprAtIE34020GP	{ boards 543 }	AT-IE340-20GP-80, 16x 10/100/1000T, 4x 100/1000X SFP, Industrial Ethernet, Layer 3 Switch, PoE+ Support.
pprIE21010GP	{ boards 544 }	AT-IE210L-10GP, 8x 10/100/1000T, 2x 100/1000X SFP, Industrial Ethernet, Layer 2 Switch, PoE+ Support.
pprIE21018GP	{ boards 545 }	AT-IE210L-18GP, 16x 10/100/1000T, 2x 100/1000X SFP, Industrial Ethernet, Layer 2 Switch, PoE+ Support.
pprAtXem2XTm12	{ boards 546 }	AT-XEM2-12XTm-B0y, 12 x 1/2.5/5/10G RJ45 ports.
pprAtXem2XSTm8	{ boards 547 }	AT-XEM2-8XSTm-B0y, 4 x 1/2.5/5/10G RJ45 ports and 4 x 1G/10G SFP+ ports.
pprx95028XTQm	{ boards 548 }	AT-950-28XTQm, 24 x 1/2.5/5/10G copper ports, 4 x 40G/100G uplink ports, and 1 x expansion bay.
pprx95028XSQ	{ boards 549 }	AT-950-28XSQ, 24 x 1/10G SFP+ ports, 4 x 40G/100G uplink ports, and 1 x expansion bay.
pprx230L17GT	{ boards 556 }	L3 switch with 16 x 10/100/1000T ports and 1 x 100/1000X SFP port.
pprx230L26GT	{ boards 557 }	L3 switch with 24 x 10/100/1000T ports and 2 x 100/1000X SFP ports.
pprx32010GH	{ boards 558 }	8-port 10/100/1000T PoE++ switch with 2 SFP ports, and 3 external PSU ports.
pprx32011GPT	{ boards 559 }	8-port 10/100/1000T PoE+ switch with 2 SFP ports, one AC adapter port(4), and one PoE-in port(5) (supporting PD and PoE pass-through)
pprAtXem2XS12V2	{ boards 560 }	2 x 10GbE (SFP+) ports?
pprAR1050V	{ boards 561 }	1 x GE WAN and 4 x 10/100/1000 LAN.

Table 6: Object identifiers for base CPU and expansion boards (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
pprAtSBx81GC40	{ boards 562 }	AT-SBx81GC40, line card for SBx8100.
pprx55018XSQV2	{ boards 563 }	AT-x550-18XSQ, stackable 10 Gigabit Layer 3 switch 16 x 1G/10G SFP+ ports and 2 x 40G uplinks.
pprx530L28GTX	{ boards 568 }	AT-x530L-28GTX, 24-port 10/100/1000T stackable switch with 4 SFP+ ports and 2 fixed power supplies.
pprx530L28GPX	{ boards 569 }	AT-x530L-28GPX, 24-port 10/100/1000T PoE+ stackable switch with 4 SFP+ ports and 2 fixed power supplies.
pprx530L52GTX	{ boards 570 }	AT-x530L-52GTX, 48-port 10/100/1000T stackable switch with 4 SFP+ ports and 2 fixed power supplies.
pprx530L52GPX	{ boards 571 }	AT-x530L-52GPX, 48-port .10/100/1000T PoE+ stackable switch with 4 SFP+ ports and 2 fixed power supplies
pprAtFS980M28DP	{ boards 573 }	AT-FS980M/28DP, 24-port 10/100TX PoE+ switch with 4 SFP uplinks and dual fixed AC power supply.
pprAtSBx81FAN12v2	{ boards 575 }	AT-x81FAN12, contains four fans, temperature sensors and controller board for SBx8106 chassis.
pprx95052XSQ	{ boards 577 }	AT-x950-52XSQ, 48 x SFP+, 4 x QSFP+/QSFP28.
pprAtGS980EM10H	{ boards 578 }	AT-GS980EM/10H, 8-port 10/100/1000T PoE++ switch with 2 SFP ports, and 3 external PSU ports.
pprAtGS980EM11PT	{ boards 579 }	AT-GS980EM/11PT, 8-port 10/100/1000 PoE+ switch with 2 SPF ports, one AC adapter port and one PoE-in port (supporting PD and PoE pass-through).
pprVAPA	{ boards 580 }	AT-VAPA, Virtual AMF Plus Appliance.
pprAtSBx81GP24v2	{ boards 581 }	AT-SBx81GP24 v2, 24-port 10/100/1000T PoE+ Ethernet line card.
pprx530L10GHXm	{ boards 582 }	AT-x530-10GHXm, 8 x Multigig POE++, 2 x SFP+, single PSU.
pprAtGS980MX10HSm	{ boards 586 }	AT-GS980MX/10HSm, 8 x Multigig POE++, 2 x SFP+, single PSU.
pprSBx81CFC960v2a	{ boards 588 }	AT-SBx81CFC960 v2, 960Gbps Controller fabric card with 4 x 1/10GbE(6) ports. Version 2 has 1G and 10G speeds.

Table 7: Object identifiers for interface types

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
iftypes	{ objects 3 }	A subtree beneath which interface type ids are assigned.
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 8: Object identifiers for chip sets

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
chips	{ objects 4 }	A subtree beneath which chip IDs are assigned.
chip68020Cpu	{ chips 1 }	MC68020 CPU.
chip68340Cpu	{ chips 2 }	MC68340 CPU.
chip68302Cpu	{ chips 3 }	MC68302 CPU.
chip68360Cpu	{ chips 4 }	MC68360 CPU.
chip860TCpu	{ chips 5 }	MPC860T CPU.
chipMips4kcCpu	{ chips 6 }	Dual MIPS CPU.
chipRtc1	{ chips 21 }	Real Time Clock v1.
chipRtc2	{ chips 22 }	Real Time Clock v2.
chipRtc3	{ chips 23 }	Real Time Clock v3.
chipRtc4	{ chips 24 }	Real Time Clock v4.
chipRam1mb	{ chips 31 }	1 MB RAM.
chipRam2mb	{ chips 32 }	2 MB RAM.
chipRam3mb	{ chips 33 }	3 MB RAM.
chipRam4mb	{ chips 34 }	4 MB RAM.
chipRam6mb	{ chips 36 }	6 MB RAM.
chipRam8mb	{ chips 38 }	8 MB RAM.
chipRam12mb	{ chips 42 }	12 MB RAM.
chipRam16mb	{ chips 46 }	16 MB RAM.
chipRam20mb	{ chips 50 }	20 MB RAM.
chipRam32mb	{ chips 62 }	32 MB RAM.
chipFlash1mb	{ chips 71 }	1 MB FLASH memory.
chipFlash2mb	{ chips 72 }	2 MB FLASH memory.
chipFlash3mb	{ chips 73 }	3 MB FLASH memory.
chipFlash4mb	{ chips 74 }	4 MB FLASH memory.
chipFlash6mb	{ chips 76 }	6 MB FLASH memory.
chipFlash8mb	{ chips 78 }	8 MB FLASH memory.
chipPem	{ chips 120 }	Processor Enhancement Module.

AT-CHASSIS-MIB

See Table 9 for the list of objects defined in the AT-CHASSIS-MIB. These objects manage chassis-based devices. Objects in this group have the object identifier **chassis** { sysinfo 23 } OID 1.3.6.1.4.1.207.8.4.4.3.23.

Table 9: Objects defined in AT-CHASSIS-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
chassis	{ sysinfo 23 } (1.3.6.1.4.1.207.8.4.4.3.23)	Trap notifications for chassis based devices.
chassisNotifications	{ chassis 0 }	List of traps (notifications) generated for the chassis.
chassisCardRoleChangeNotify	{ chassis 0.1 }	Notification generated when the Control Fabric Card's role changes.
chassisCardJoinNotify	{ chassis 0.2 }	Notification generated when a line card connects to the Control Fabric Card.
chassisCardLeaveNotify	{ chassis 0.3 }	Notification generated when a line card detaches from the Control Fabric Card.
slotNumber	{ chassis 0.4 }	The slot number of the line card or Control Fabric Card that has changed. Has the numeric range 1-12.
chassisRole	{ chassis 0.5 }	The Control Fabric Card's role in the chassis. Can be one of the following: ■ leaving (1) ■ discovering (2) ■ synchronizing (3) ■ standbyMember (4) ■ pendingMaster (5) ■ disabledMaster (6) ■ activeMaster (7).
chassisCardTable	{ chassis 1 }	A list of cards presented on the device.
chassisCardEntry	{ chassis 1.1 }	A table entry containing information about a card.
chassisCardSlot	{ chassis 1.1.1 }	The slot number that the card is installed in.
chassisCardBoardOID	{ chassis 1.1.2 }	The object ID used to identify the AT board type as defined in the AT Enterprise MIB.
chassisCardName	{ chassis 1.1.3 }	The name of the card; e.g. AT-SBx81CFC400, AT-SBx81CFC960 for controller cards; and AT-SBx81GP24, AT-SBx81GT24, AT-SBx81GS24, AT-SBx81XS6 for line cards; "unknown" is displayed for unsupported hardware.

Table 9: Objects defined in AT-CHASSIS-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
chassisCardState	{ chassis 1.1.4 }	The current state of the card. The valid state is in one of the following: ■ unknown(1), ■ configuring(2), ■ syncing(3), ■ online(4), ■ syncingFirmware(5), ■ joining(6), ■ incompatibleSW(7), ■ disabled(8), ■ initializing(9), ■ booting(10), ■ unsupportedHW(11), ■ provisioned(12).
chassisCardControllerState	{ chassis 1.1.5 }	The current state of the controller card, in addition to the card state. The valid state is in one of the following: ■ unknown(1) ■ active(2) ■ standby(3).
chassiCardSwVersion	{ chassis 1.1.6 }	The software version that the card is running. When feature ISSU (In Service Software Update) is supported, there are situations when a LIF card may run a different software version to that running on the active CFC.
chassismappingtable	{ chassis 2 }	A list of cards existing on the device.
chassismappingentry	{ chassis 2.1 }	Other SNMP SET/GET operations that refer to a card in a chassis using a single integer index, i.e. a 'node-ID'. This table entry maps a node-ID value to the VCS member-ID and slot number of the card. For example, node-ID 17 would map to card 2.5 (VCS member-ID 2, slot 5).
chassisNodeId	{ chassis 2.1.1 }	The node ID of a chassis card.
chassisVCSMemberId	{ chassis 2.1.2 }	The VCS member ID of the chassis that contains the card.
chassisSlotNumber	{ chassis 2.1.3 }	The chassis slot number that the card is installed in.
chassisNodeDisplayString	{ chassis 2.1.4 }	The name of the chassis card, e.g. 'card 1.5'.
chassisNodeStateString	{ chassis 2.1.5 }	The current state of the card located in the bay. If the card is present, then its state is displayed as “online”. If the bay is provisioned then the display will be “Provisioned”.

AT-DHCPSN-MIB

This MIB contains objects for displaying and managing DHCP snooping and ARP security information on the switch, see Table 10. The objects reside in the module **atDhcpsn** { modules 537 } OID 1.3.6.1.4.1.207.8.4.4.4.537, organized in the following groups:

- The DHCP Snooping Events group (atDhcpsnEvents) contains notifications (traps).
- The DHCP Snooping table (atDhcpsnVariablesTable) contains DHCP snooping information.
- The ARP Security table (atArpsecVariablesTable) contains ARP security information.

Table 10: Objects defined in AT-DHCPSN-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atDhcpsn	{ modules 537 } or 1.3.6.1.4.1.207.8.4.4.4.537	This MIB file contains definitions of managed objects for DHCP Snooping in AlliedWare Plus.
atDhcpsnEvents	{ atDhcpsn 0 }	DHCP Snooping notifications (traps).
atDhcpsnTrap	{ atDhcpsnEvents 1 }	DHCP Snooping violation notification.
atArpsecTrap	{ atDhcpsnEvents 2 }	DHCP Snooping ARP Security violation notification.
atDhcpsnVariablesTable	{ atDhcpsn 1 }	The DHCP Snooping table. This table contains rows of DHCP Snooping information.
atDhcpsnVariablesEntry	{ atDhcpsnVariablesTable 1 }	A set of parameters that describe the DHCP Snooping features.
atDhcpsnIfIndex	{ atDhcpsnVariablesEntry 1 }	Ifindex of the port that the packet was received on.
atDhcpsnVid	{ atDhcpsnVariablesEntry 2 }	VLAN ID of the port that the packet was received on.
atDhcpsnSmac	{ atDhcpsnVariablesEntry 3 }	Source MAC address of the packet that caused the trap.
atDhcpsnOpcode	{ atDhcpsnVariablesEntry 4 }	Opcode value of the BOOTP packet that caused the trap. Only bootpRequest(1) or bootpReply(2) is valid.
atDhcpsnCiaddr	{ atDhcpsnVariablesEntry 5 }	Ciaddr value of the BOOTP packet that caused the trap.
atDhcpsnYiaddr	{ atDhcpsnVariablesEntry 6 }	Yiaddr value of the BOOTP packet that caused the trap.
atDhcpsnGiaddr	{ atDhcpsnVariablesEntry 7 }	Giaddr value of the BOOTP packet that caused the trap.
atDhcpsnSiaddr	{ atDhcpsnVariablesEntry 8 }	Siaddr value of the BOOTP packet that caused the trap.
atDhcpsnChaddr	{ atDhcpsnVariablesEntry 9 }	Chaddr value of the BOOTP packet that caused the trap.

Table 10: Objects defined in AT-DHCPSN-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atDhcpsnVioType	{ atDhcpsnVariablesEntry 10 }	The reason that the trap was generated. ■ invalidBootp(1) indicates that the received BOOTP packet was invalid. For example, it is neither BootpRequest nor BootpReply. ■ invalidDhcpAck(2) indicates that the received DHCP ACK was invalid. ■ invalidDhcpRelDec(3) indicates the DHCP Release or Decline was invalid. ■ invalidIp(4) indicates that the received IP packet was invalid. ■ maxBindExceeded(5) indicates that if the entry was added, the maximum bindings configured for the port would be exceeded. ■ opt82InsertErr(6) indicates that the insertion of Option 82 failed. ■ opt82RxInvalid(7) indicates that the received Option 82 information was invalid.
atDhcpsnVioType (continued)		■ opt82RxUntrusted(8) indicates that Option 82 information was received on an untrusted port. ■ opt82TxUntrusted(9) indicates that Option 82 would have been transmitted out an untrusted port. ■ replyRxUntrusted(10) indicates that a BOOTP Reply was received on an untrusted port. ■ srcMacChaddrMismatch(11) indicates that the source MAC address of the packet did not match the BOOTP CHADDR of the packet. ■ staticEntryExisted(12) indicates that the static entry to be added already exists. ■ dbAddErr(13) indicates that adding an entry to the database failed.
atArpsecVariablesTable	{ atDhcpsn 2 }	The ARP Security table. This table contains rows of DHCP Snooping ARP Security information.
atArpsecVariablesEntry	{ atArpsecVariablesTable 1 }	A set of parameters that describe the DHCP Snooping ARP Security features.
atArpsecIfIndex	{ atArpsecVariablesEntry 1 }	Ifindex of the port that the ARP packet was received on.
atArpsecClientIP	{ atArpsecVariablesEntry 2 }	Source IP address of the ARP packet.
atArpsecSrcMac	{ atArpsecVariablesEntry 3 }	Source MAC address of the ARP packet.
atArpsecVid	{ atArpsecVariablesEntry 4 }	VLAN ID of the port that the ARP packet was received on.

Table 10: Objects defined in AT-DHCPSN-MIB (continued)

OBJECT		OBJECT IDENTIFIER	DESCRIPTION
	atArpsecVioType	{ atArpsecVariablesEntry 5 }	The reason that the trap was generated. ■ srcIpNotFound(1) indicates that the Sender IP address of the ARP packet was not found in the DHCP Snooping database. ■ badVLAN(2) indicates that the VLAN of the DHCP Snooping binding entry associated with the Sender IP address of the ARP packet does not match the VLAN that the ARP packet was received on. ■ badPort(3) indicates that the port of the DHCP Snooping binding entry associated with the Sender IP address of the ARP packet does not match the port that the ARP packet was received on. ■ srcIpNotAllocated(4) indicates that the CHADDR of the DHCP Snooping binding entry associated with the Sender IP address of the ARP packet does not match the Source MAC and/or the ARP source MAC of the ARP packet.

AT-DNS-MIB

AT-DNS-MIB contains definitions of managed objects for the Allied Telesis DNS Client Configuration. Objects in this group have the object identifier **atDns** { Modules 501 } OID 1.3.6.1.4.1.207.8.4.4.4.501. Table 11 lists the objects supported by the AlliedWare Plus™ Operating System.

Table 11: Objects defined in AT-DNS-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atDnsClient	{ atDns } or 1.3.6.1.4.1.207.8.4.4.4.501	MIB File for DNS Client Configuration.
atDNSServerIndexNext	{ atDnsClient 1 }	The next available value for the object 'atDNSServerIndex'. The value is used by a management application to create an entry in the 'atDNSServerTable'.
atDNSServerTable	{ atDnsClient 2 }	Table of information about the Domain Name System (DNS) Server configurations in the system, indexed by 'atDNSServerIndex'.
atDNSServerEntry	{ atDNSServerTable 1 }	Information about a single DNS Server Configuration.
atDNSServerIndex	{ atDNSServerEntry 1 }	The index corresponding to the particular DNS Server Configuration. When creating a new entry in the table, the value of this object must be equal to the value in the 'atDNSServerIndexNext'.
atDNSServerAddrType	{ atDNSServerEntry 2 }	The Internet Address Type of the 'atDNSServerAddr' object. Can be one of the following: ■ unknown (0) ■ ipv4 (1) - default ■ ipv6 (2) - not supported ■ ipv4z (3) - not supported ■ ipv6z (4) - not supported ■ dns (16) - not supported.
atDNSServerAddr	{ atDNSServerEntry 3 }	The IP Address of the DNS Server. When a new entry is created, this object is set to the default of '0.0.0.0' { '00000000'h }. The management application will change this to the desired value using a SET operation.
atDNSServerStatus	{ atDNSServerEntry 4 }	The status of the current entry (row). Can be one of the following: ■ active (1) ■ createAndGo (4) ■ destroy (6). To create a new entry the management application must set this object with value 'createAndGo (4)'. To delete an entry, the management application must set this object with value 'destroy (6)'. Once an entry is deleted, all subsequent entries in the table will be renumbered. The default is 1 (active).

AT-ENVMONv2-MIB

The AT Environment Monitoring v2 MIB contains objects for managing and reporting data relating to fans, voltage rails, temperature sensors and power supply units installed in the device, see Table 12. Objects in this group have the object identifier **EnvMonv2** { sysinfo 12 } OID 1.3.6.1.4.1.207.8.4.4.3.12.

Table 12: Objects defined in AT-ENVMONV2-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atEnvMonv2Notifications	{ atEnvMonv2 0 } or (1.3.6.1.4.1.207.8.4.4.3.12.0)	A collection of traps (notification) objects for monitoring fans, voltage rails, temperature sensors, and power supply bays.
atEnvMonv2FanAlarmSetNotify	{ atEnvMonv2Notifications 1 } or (1.3.6.1.4.1.207.8.4.4.3.12.0.1)	A notification that is generated when the monitored speed of a fan drops below its lower threshold. It returns the value of: ■ atEnvMonv2FanStackMemberId ■ atEnvMonv2FanBoardIndex ■ atEnvMonv2FanIndex ■ atEnvMonv2FanDescription ■ atEnvMonv2FanLowerThreshold ■ atEnvMonv2FanCurrentSpeed.
atEnvMonv2FanAlarmClearedNotify	{ atEnvMonv2Notifications 2 }	Notification generated when the monitored speed of a fan returns to an acceptable value, the fan having previously been in an alarm condition. It returns the value of: ■ atEnvMonv2FanStackMemberId ■ atEnvMonv2FanBoardIndex ■ atEnvMonv2FanIndex ■ atEnvMonv2FanDescription ■ atEnvMonv2FanLowerThreshold ■ atEnvMonv2FanCurrentSpeed.
atEnvMonv2VoltAlarmSetNotify	{ atEnvMonv2Notifications 3 }	Notification generated when the voltage of a monitored voltage rail, goes out of tolerance by either dropping below its lower threshold, or exceeding its upper threshold. It returns the value of: ■ atEnvMonv2VoltageStackMemberId ■ atEnvMonv2VoltageBoardIndex ■ atEnvMonv2VoltageIndex ■ atEnvMonv2VoltageDescription ■ atEnvMonv2VoltageUpperThreshold ■ atEnvMonv2VoltageLowerThreshold ■ atEnvMonv2VoltageCurrent (i.e. the voltage currently being measured).
atEnvMonv2VoltAlarmClearedNotify	{ atEnvMonv2Notifications 4 }	Notification generated when the voltage of a monitored voltage rail returns to an acceptable value, having previously been in an alarm condition. It returns the value of: ■ atEnvMonv2VoltageStackMemberId ■ atEnvMonv2VoltageBoardIndex ■ atEnvMonv2VoltageIndex ■ atEnvMonv2VoltageDescription ■ atEnvMonv2VoltageUpperThreshold ■ atEnvMonv2VoltageLowerThreshold ■ atEnvMonv2VoltageCurrent (i.e. the voltage currently being measured).

Table 12: Objects defined in AT-ENVMONV2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atEnvMonv2TempAlarmSetNotify	{ atEnvMonv2Notifications 5 }	Notification generated when a monitored temperature exceeds its upper threshold. It returns the value of: ■ atEnvMonv2TemperatureStackMemberId ■ atEnvMonv2TemperatureBoardIndex ■ atEnvMonv2TemperatureIndex ■ atEnvMonv2TemperatureDescription ■ atEnvMonv2TemperatureUpperThreshold ■ atEnvMonv2TemperatureCurrent.
atEnvMonv2TempAlarmClearedNotify	{ atEnvMonv2Notifications 6 }	Notification generated when a monitored temperature returns to an acceptable value, having previously been in an alarm condition. It returns the value of: ■ atEnvMonv2TemperatureStackMemberId ■ atEnvMonv2TemperatureBoardIndex ■ atEnvMonv2TemperatureIndex ■ atEnvMonv2TemperatureDescription ■ atEnvMonv2TemperatureUpperThreshold.
atEnvMonv2PsbAlarmSetNotify	{ atEnvMonv2Notifications 7 }	Notification generated when a monitored parameter of a power supply bay device goes out of tolerance. It returns the value of: ■ atEnvMonv2PsbSensorStackMemberId ■ atEnvMonv2PsbSensorBoardIndex ■ atEnvMonv2PsbSensorIndex ■ atEnvMonv2PsbSensorType ■ atEnvMonv2PsbSensorDescription.
atEnvMonv2PsbAlarmClearedNotify	{ atEnvMonv2Notifications 8 }	Notification generated when a monitored parameter of a power supply bay device returns to an acceptable value, having previously been in an alarm condition. It returns the value of: ■ atEnvMonv2PsbSensorStackMemberId ■ atEnvMonv2PsbSensorBoardIndex ■ atEnvMonv2PsbSensorIndex ■ atEnvMonv2PsbSensorType ■ atEnvMonv2PsbSensorDescription.
atEnvMonv2ContactInputOpenNotify	{ atEnvMonv2Notifications 9 }	Notification generated when a monitored contact input opens. It returns the value of: ■ atEnvMonv2ContactInputStackMemberId ■ atEnvMonv2ContactInputBoardIndex ■ atEnvMonv2ContactInputIndex ■ atEnvMonv2ContactInputDescription.
atEnvMonv2ContactInputCloseNotify	{ atEnvMonv2Notifications 10 }	Notification generated when a monitored contact input closes. It returns the value of: ■ atEnvMonv2ContactInputStackMemberId ■ atEnvMonv2ContactInputBoardIndex ■ atEnvMonv2ContactInputIndex ■ atEnvMonv2ContactInputDescription.
atEnvMonv2ExternalPSUAlarmSetNotify	{ atEnvMonv2Notifications 11 }	Notification generated when supply potential of a monitored external power supply is not present. It returns the value of: ■ atEnvMonv2ExternalPSUStackMemberId ■ atEnvMonv2ExternalPSUBoardIndex ■ atEnvMonv2ExternalPSUIndex ■ atEnvMonv2ExternalPSUDescription.

Table 12: Objects defined in AT-ENVMONV2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atEnvMonv2ExternalPSUAlarmClearedNotify	{ atEnvMonv2Notifications 12 }	Notification generated when supply potential of a monitored external power supply returns to an acceptable level, having previously been in alarm condition. It returns the value of: ■ atEnvMonv2ExternalPSUStatusStackMemberId ■ atEnvMonv2ExternalPSUStatusBoardIndex ■ atEnvMonv2ExternalPSUStatusIndex ■ atEnvMonv2ExternalPSUStatusDescription.
atEnvMonv2TemIpCriticalSetNotify	{ atEnvMonv2TemperatureStackMemberid 13 }	A notification generated when a monitored temperature exceeds its upper threshold.
atEnvMonv2TempCriticalClearedNotify	{ atEnvMonv2TemperatureStackMemberid 14 }	A notification generated when a monitored temperature returns to an acceptable value, having previously been in an alarm condition.
atEnvMonv2TempCriticalLevel1SetNotify	{ atEnvMonv2TemperatureStackMemberid 15 }	A trap generated when a monitored temperature exceeds the level 1 threshold.
atEnvMonv2TempCriticalLevel1ClearedNotify	{ atEnvMonv2TemperatureStackMemberid 16 }	A trap generated when a monitored temperature returns to an acceptable value, having previously been in critical temperature level 1 condition.
atEnvMonv2TempCriticalLevel2SetNotify	{ atEnvMonv2TemperatureStackMemberid 17 }	A trap generated when a monitored temperature exceeds the level 2 threshold.
atEnvMonv2TempCriticalLevel2ClearedNotify	{ atEnvMonv2TemperatureStackMemberid 18 }	A trap generated when a monitored temperature returns to an acceptable value, having previously been in critical temperature level 2 condition.
atEnvMonv2PsbPresence ChangeNotify	{ atEnvMonv2PsbSensorStackMemberid 19 }	A notification generated when an insertion or removal event is detected on a power supply bay device.
atEnvMonv2FanTable	{ EnvMonv2 1 } (1.3.6.1.4.1.207.8.4.4.3.12.1)	Table of information about fans installed in the device that have their fan speeds monitored by environment monitoring hardware, indexed by: ■ atEnvMonv2FanStackMemberId ■ atEnvMonv2FanBoardIndex ■ atEnvMonv2FanIndex.
atEnvMonv2FanEntry	{ atEnvMonv2FanTable 1 }	Description, current speed, lower threshold speed and current status of a single fan.
atEnvMonv2FanStackMemberId	{ atEnvMonv2FanEntry 1 }	Index of the stack member hosting this fan.
atEnvMonv2FanBoardIndex	{ atEnvMonv2FanEntry 2 }	Index of the board hosting this fan in the board table.
atEnvMonv2FanIndex	{ atEnvMonv2FanEntry 3 }	Numeric identifier of this fan on its host board.
atEnvMonv2FanDescription	{ atEnvMonv2FanEntry 4 }	Description of this fan.
atEnvMonv2FanCurrentSpeed	{ atEnvMonv2FanEntry 5 }	Current speed of this fan in revolutions per minute.
atEnvMonv2FanLowerThreshold	{ atEnvMonv2FanEntry 6 }	Minimum acceptable speed of the fan in revolutions per minute.

Table 12: Objects defined in AT-ENVMONV2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atEnvMonv2FanStatus	{ atEnvMonv2FanEntry 7 }	Whether this fan is currently in an alarm condition. The values can be: ■ Failed. Means that the current speed is too low. ■ Good. Means that the current speed is acceptable.
atEnvMonv2VoltageTable	{ atEnvMonv2 2 }	Table of information about voltage rails in the device that are monitored by environment monitoring hardware, indexed by: ■ atEnvMonv2VoltageStackMemberId ■ atEnvMonv2VoltageBoardIndex ■ atEnvMonv2VoltageIndex.
atEnvMonv2VoltageEntry	{ atEnvMonv2VoltageTable 1 }	Description, current value, upper & lower threshold settings and current status of a single voltage rail.
atEnvMonv2VoltageStackMemberId	{ atEnvMonv2VoltageEntry 1 }	Index of the stack member hosting this voltage sensor.
atEnvMonv2VoltageBoardIndex	{ atEnvMonv2VoltageEntry 2 }	Index of the board hosting this voltage sensor in the board table.
atEnvMonv2VoltageIndex	{ atEnvMonv2VoltageEntry 3 }	Numeric identifier of this voltage rail on its host board.
atEnvMonv2VoltageDescription	{ atEnvMonv2VoltageEntry 4 }	Description of this voltage rail.
atEnvMonv2VoltageCurrent	{ atEnvMonv2VoltageEntry 5 }	Current reading of this voltage rail in millivolts.
atEnvMonv2VoltageUpperThreshold	{ atEnvMonv2VoltageEntry 6 }	Maximum acceptable reading of this voltage rail in millivolts.
atEnvMonv2VoltageLowerThreshold	{ atEnvMonv2VoltageEntry 7 }	Minimum acceptable reading of this voltage rail in millivolts.
atEnvMonv2VoltageStatus	{ atEnvMonv2VoltageEntry 8 }	Whether this voltage rail is currently in an alarm condition. Possible values are: ■ outOfRange (1) - means that the current reading is outside the threshold range. ■ inRange (2) - means that the current reading is acceptable.
atEnvMonv2TemperatureTable	{ atEnvMonv2 3 }	Table of information about temperature sensors in the device that are monitored by environment monitoring hardware, indexed by: ■ atEnvMonv2TemperatureStackMemberId ■ atEnvMonv2TemperatureBoardIndex ■ atEnvMonv2TemperatureIndex ■ atEnvMonv2TemperatureDescription ■ atEnvMonv2TemperatureCurrent ■ atEnvMonv2TemperatureUpperThreshold ■ atEnvMonv2TemperatureStatus.
atEnvMonv2TemperatureEntry	{ atEnvMonv2TemperatureTable 1 }	Description, current value, upper threshold setting and current status of a single temperature sensor.
atEnvMonv2TemperatureStackMemberId	{ atEnvMonv2TemperatureEntry 1 }	Index of the stack member hosting this temperature sensor.
atEnvMonv2TemperatureBoardIndex	{ atEnvMonv2TemperatureEntry 2 }	Index of the board hosting this temperature sensor in the board table.
atEnvMonv2TemperatureIndex	{ atEnvMonv2TemperatureEntry 3 }	Numeric identifier of this temperature sensor on its host board.

Table 12: Objects defined in AT-ENVMONV2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atEnvMonv2TemperatureDescription	{ atEnvMonv2TemperatureEntry 4 }	Description of this temperature sensor.
atEnvMonv2TemperatureCurrent	{ atEnvMonv2TemperatureEntry 5 }	Current reading of this temperature sensor in degrees Celsius.
atEnvMonv2TemperatureUpperThreshold	{ atEnvMonv2TemperatureEntry 6 }	Maximum acceptable reading for this temperature sensor in degrees Celsius.
atEnvMonv2TemperatureStatus	{ atEnvMonv2TemperatureEntry 7 }	Whether this temperature sensor is currently in an alarm condition. Can be: ■ outOfRange (1) - means that the current reading is outside the threshold range. ■ inRange (2) - means that the current reading is acceptable.
atEnvMonv2PsbObjects	{ atEnvMonv2 4 }	Collection of objects for monitoring power supply bays in the system and any devices that are installed. It contains the following objects: ■ atEnvMonv2PsbTable ■ atEnvMonv2PsbSensorTable.
atEnvMonv2PsbTable	{ atEnvMonv2PsbObjects 1 }	Table of information about power supply bays in the system, indexed by: ■ atEnvMonv2PsbHostStackMemberId ■ atEnvMonv2PsbHostBoardIndex ■ atEnvMonv2PsbHostSlotIndex ■ atEnvMonv2PsbHeldBoardIndex ■ atEnvMonv2PsbHeldBoardId ■ atEnvMonv2PsbDescription.
atEnvMonv2PsbEntry	{ atEnvMonv2PsbTable 1 }	Description and current status of a single power supply bay device.
atEnvMonv2PsbHostStackMemberId	{ atEnvMonv2PsbEntry 1 }	Index of the stack member hosting this power supply bay.
atEnvMonv2PsbHostBoardIndex	{ atEnvMonv2PsbEntry 2 }	Index of the board hosting this power supply bay in the board table.
atEnvMonv2PsbHostSlotIndex	{ atEnvMonv2PsbEntry 3 }	Index of this power supply bay slot on its host board. This index is fixed for each slot, on each type of board.
atEnvMonv2PsbHeldBoardIndex	{ atEnvMonv2PsbEntry 4 }	Index of a board installed in this power supply bay. This value corresponds to atEnvMonv2PsbSensorBoardIndex for each sensor on this board. A value of 0 indicates that a board is either not present or not supported.
atEnvMonv2PsbHeldBoardId	{ atEnvMonv2PsbEntry 5 }	Type of board installed in this power supply bay. The values of this object are taken from the pprXxx object IDs under the boards sub-tree in the parent MIB. A value of 0 indicates that a board is either not present or not supported.
atEnvMonv2PsbDescription	{ atEnvMonv2PsbEntry 6 }	Description of this power supply bay.
atEnvMonv2PsbSensorTable	{ atEnvMonv2PsbObjects 2 }	Table of information about environment monitoring sensors on devices installed in power supply bays, indexed by: ■ atEnvMonv2PsbSensorStackMemberId ■ atEnvMonv2PsbSensorBoardIndex ■ atEnvMonv2PsbSensorIndex.

Table 12: Objects defined in AT-ENVMONV2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atEnvMonv2PsbSensorEntry	{ atEnvMonv2PsbSensorTable 1 }	Description and current status of the sensor on a device installed in a power supply bay.
atEnvMonv2PsbSensorStackMemberId	{ atEnvMonv2PsbSensorEntry 1 }	Index of the stack member hosting this sensor.
atEnvMonv2PsbSensorBoardIndex	{ atEnvMonv2PsbSensorEntry 2 }	Index of the board hosting this sensor in the board table.
atEnvMonv2PsbSensorIndex	{ atEnvMonv2PsbSensorEntry 3 }	Index of this power supply bay environmental sensor on its host board.
atEnvMonv2PsbSensorType	{ atEnvMonv2PsbSensorEntry 4 }	Type of environmental variable this sensor detects. One of: ■ psbSensorTypeInvalid(0) ■ fanSpeedDiscrete(1) ■ temperatureDiscrete(2) ■ voltageDiscrete(3).
atEnvMonv2PsbSensorDescription	{ atEnvMonv2PsbSensorEntry 5 }	Description of this power supply bay environmental sensor.
atEnvMonv2PsbSensorStatus	{ atEnvMonv2PsbSensorEntry 6 }	Whether this environmental sensor is currently in an alarm condition. One of: ■ failed (1) - the device is in a failure condition ■ good (2) - the device is functioning normally. ■ notPowered (3) - a PSU is installed, but not powered up.
atEnvMonv2PsbSensorReading	{ atEnvMonv2PsbSensorEntry 7 }	An indication of whether this environmental sensor is currently reading a value for the monitored device. A value of 'no' indicates that there is no current reading, 'yes' indicates that the monitored device is supplying a reading. ■ no ■ yes.
atEnvMonv2Traps	{ atEnvMonv2 5 } or (1.3.6.1.4.1.207.8.4.4.3.12.5)	Note that objects under this portion of the tree have been deprecated, and replaced by objects under the tree portion 207.8.4.4.3.12.0.
atEnvMonv2FaultLedTable	{ atEnvMonv2 6 }	Table detailing any LED fault indications on the device, indexed by: ■ atEnvMonv2FaultLedStackMemberId.
atEnvMonv2FaultLedEntry	{ atEnvMonv2FaultLedTable 1 }	Information pertaining to a given fault LED.
atEnvMonv2FaultLedStackMemberId	{ atEnvMonv2FaultLedEntry 1 }	Index of the stack member hosting this fault LED.
atEnvMonv2FaultLed1Flash	{ atEnvMonv2FaultLedEntry 2 }	Indicates whether a fault LED is currently showing a system failure by flashing once. Values can be: ■ heatsinkFanFailure (1) - indicates that one or more heatsink fans have failed, or are operating below the recommended speed ■ noFault (2).

Table 12: Objects defined in AT-ENVMONV2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atEnvMonv2FaultLed2Flashes	{ atEnvMonv2FaultLedEntry 3 }	Indicates whether a fault LED is currently showing a system failure by flashing twice. Values can be: ■ chassisFanFailure (1) - indicates that one or both of the chassis fans are not installed, or the fans are operating below the recommended speed ■ noFault (2).
atEnvMonv2FaultLed3Flashes	{ atEnvMonv2FaultLedEntry 4 }	Indicates whether a fault LED is currently showing a system failure by flashing three times. Values can be: ■ sensorFailure (1) - indicates that the ability to monitor temperature or fans has failed ■ noFault (2).
atEnvMonv2FaultLed4Flashes	{ atEnvMonv2FaultLedEntry 5 }	Indicates whether a fault LED is currently showing a system failure by flashing four times. Values can be: ■ xemInitialisationFailure (1) - indicates that a XEM failed to initialise or is incompatible ■ noFault (2).
atEnvMonv2FaultLed5Flashes	{ atEnvMonv2FaultLedEntry 6 }	Indicates whether a fault LED is currently showing a system failure by flashing five times. Values can be: ■ alarmMonitorAlarm (1) - indicates that the Alarm Monitor has detected one or more fault conditions. ■ noFault (2).
atEnvMonv2FaultLed6Flashes	{ atEnvMonv2FaultLedEntry 7 }	Indicates whether a fault LED is currently showing a system failure by flashing six times. Values can be: ■ temperatureFailure (1) - indicates that the device's temperature has exceeded the recommended threshold ■ noFault (2).
atEnvMonv2ContactInputTable	{ atEnvMonv2 7 } (1.3.6.1.4.1.207.8.4.4.3.12.7)	Table of information about contact inputs available in the device that are monitored by environment monitoring hardware, indexed by: ■ atEnvMonv2ContactInputStackMemberId ■ atEnvMonv2ContactInputBoardIndex ■ atEnvMonv2ContactInputIndex.
atEnvMonv2ContactInputEntry	{ atEnvMonv2ContactInputTable 1 }	The description and current state of a contact input.
atEnvMonv2ContactInputStackMemberId	{ atEnvMonv2ContactInputEntry 1 }	Index of the stack member hosting this input contact.
atEnvMonv2ContactInputBoardIndex	{ atEnvMonv2ContactInputEntry 2 }	Index of the board hosting this input contact in the board table.
atEnvMonv2ContactInputIndex	{ atEnvMonv2ContactInputEntry 3 }	The numeric identifier of this contact input on its host board.
atEnvMonv2ContactInputDescription	{ atEnvMonv2ContactInputEntry 4 }	The description of this contact input.
atEnvMonv2ContactInputState	{ atEnvMonv2ContactInputEntry 5 }	Current state of the input contact - closed (1) or open (2).

Table 12: Objects defined in AT-ENVMONV2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atEnvMonv2ContactIOutputTable	{ atEnvMonv2 8}	Table of information about contact outputs available in the device that are managed by environment monitoring hardware, indexed by: ■ atEnvMonv2ContactOutputStackMemberId ■ atEnvMonv2ContactOutputBoardIndex ■ atEnvMonv2ContactOutputIndex.
atEnvMonv2ContactOutputEntry	{ atEnvMonv2ContactOutputTable 1 }	Table of information for contact input and output entries.
atEnvMonv2ContactOutputStackMemberId	{ atEnvMonv2ContactInputEntry 1 }	Index of the stack member hosting this output contact.
atEnvMonv2ContactOutputBoardIndex	{ atEnvMonv2ContactInputEntry 2 }	Index of the board hosting this contact output in the board table.
atEnvMonv2ContactOutputIndex	{ atEnvMonv2ContactInputEntry 3 }	The numeric identifier of this contact output on its host board.
atEnvMonv2ContactOutputDescription	{ atEnvMonv2ContactInputEntry 4 }	The description of this contact output.
atEnvMonv2ContactOutputState	{ atEnvMonv2ContactInputEntry 5 }	Current state of the output contact - closed (1) or open (2).
atEnvMonv2ExternalPSUTable	{ atEnvMonv2 9 }	Table of information about external power supply status monitored by environment monitoring hardware, indexed by: ■ atEnvMonv2ExternalPSUStatusStackMemberId ■ atEnvMonv2ExternalPSUStatusBoardIndex ■ atEnvMonv2ExternalPSUStatusIndex.
atEnvMonv2ExernalPSUEntry	{ atEnvMonv2ExternalPSUStatusTable 1 }	Table of external PSU status with descriptions and current status of an external power supply.
atEnvMonv2ExternalPSUStackMemberId	{ atEnvMonv2ExternalPSUEntry 1 }	Index of the stack member hosting this external power supply.
atEnvMonv2ExternalPSUBoardIndex	{ atEnvMonv2ExternalPSUEntry 2 }	Index of the board hosting this external power supply in the board table.
atEnvMonv2ExternalPSUIndex	{ atEnvMonv2ExternalPSUEntry 3 }	The numeric identifier of this external power supply on its host board.
atEnvMonv2ExternalPSUDescription	{ atEnvMonv2ExternalPSUEntry 4 }	The description of this external power supply.
atEnvMonv2ExternalPSUState	{ atEnvMonv2ExternalPSUEntry 5 }	Current state of the output contact - failed (1) or good (2).

AT-EPSRv2-MIB

The EPSRv2 Group-MIB defines objects for managing Epsrv2 objects and triggers, see Figure 4 and Table 13. Objects in this group have the object identifier **Epsrv2** { modules 536 } OID 1.3.6.1.4.1.207.8.4.4.4.536.

Figure 4: The AT-EPSRv2 MIB sub-tree

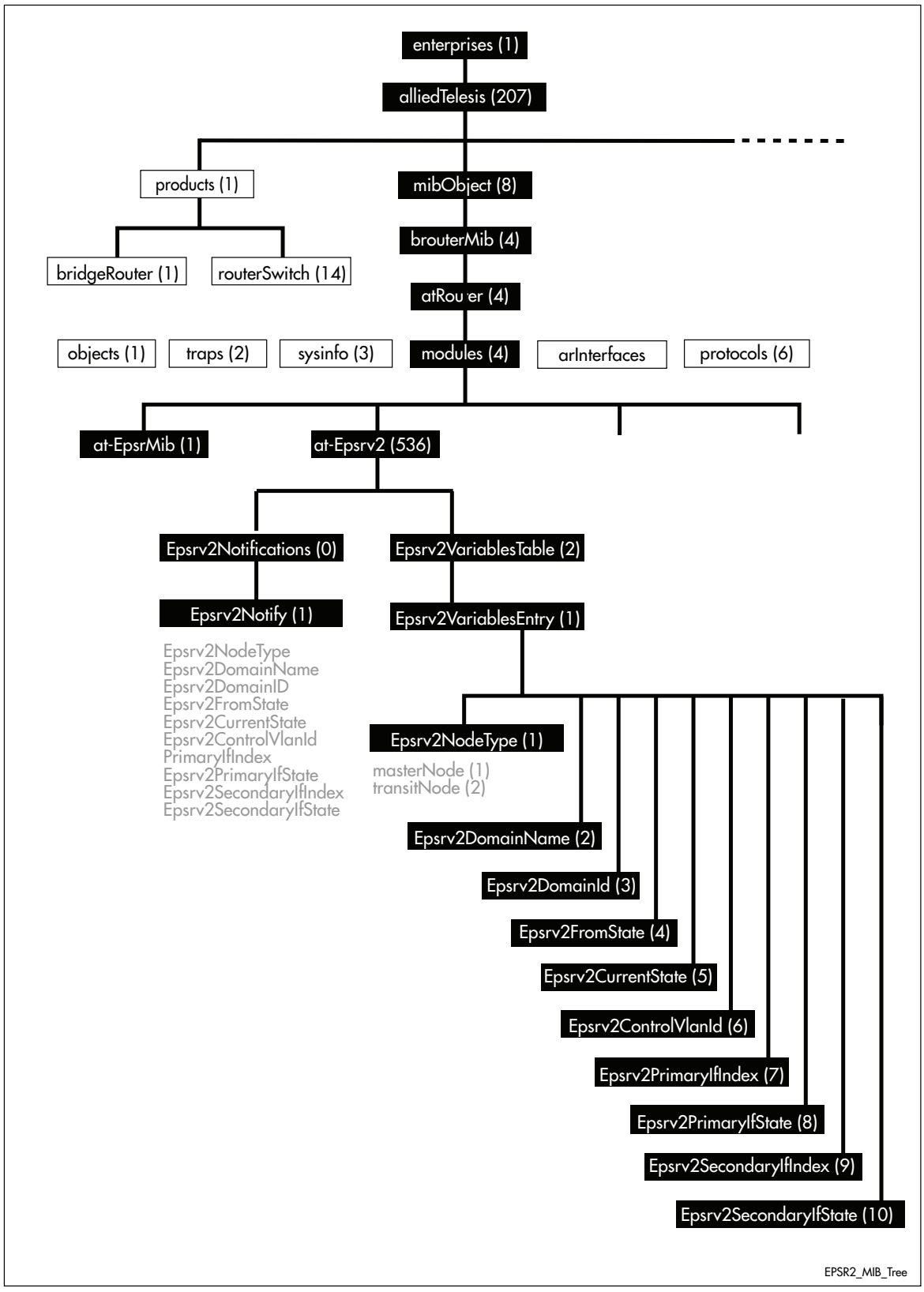


Table 13: atEpsrv2Objects Defined in the AT-EPSRV2 MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
{ at-Epsrv2 }	{ modules 536 } 1.3.6.1.4.1.207.8.4.4.4.536	The root of the Epsrv2 object sub tree.
{ atEpsrv2Notifications }	{ at-Epsrv2 0 }	EPSRv2 Master/Transit node state transition notification.
{ atEpsrv2Notify }	{ atEpsrv2Notifications 1 }	EPSR Master/Transit node state transition trap. Note that there is a one to one relationship between nodes and domains.
{ Epsrv2NodeType }	{ atEpsrv2VariablesEntry 1 }	The EPSR node type: either master or transit.
{ atEpsrv2DomainName }	{ atEpsrv2VariablesEntry 2 }	The name of the EPSR domain.
{ atEpsrv2DomainID }	{ atEpsrv2VariablesEntry 3 }	The ID of the EPSR domain.
{ Epsrv2FromState }	{ atEpsrv2VariablesEntry 4 }	The previous state of the EPSR domain.
{ Epsrv2Current State }	{ atEpsrv2VariablesEntry 5 }	The current state of the EPSR domain.
{ Epsrv2ControlVlanId }	{ atEpsrv2VariablesEntry 6 }	The VLAN identifier for the control VLAN.
{ Epsrv2PrimaryIfIndex }	{ atEpsrv2VariablesEntry 7 }	The IfIndex of the primary interface.
{ atEpsrv2PrimaryIfState }	{ atEpsrv2VariablesEntry 8 }	The current state of the primary interface.
{ atEpsrv2SecondaryIfIndex }	{ atEpsrv2VariablesEntry 9 }	The IfIndex of the secondary interface.
{ atEpsrv2SecondaryIfState }	{ atEpsrv2VariablesEntry 10 }	The state of the secondary interface.
{ atEpsrv2VariablesTable }	{ at-Epsrv2 2 }	The enterprise Epsrv2VariablesTable.
{ atEpsrv2VariablesEntry }	{ atEpsrv2VariablesTable 1 }	Contains entries within the enterprise atEpsrv2VariablesTable.
{ atEpsrv2NodeType }	{ atEpsrv2VariablesEntry 1 }	The EPSR domain node type: either ■ master (1) ■ transit (2).
{ atEpsrv2DomainName }	{ Epsrv2NodeType 2 }	The name of the EPSR domain.
{ atEpsrv2DomainID }	{ Epsrv2NodeType 3 }	The ID of the EPSR domain.
{ atEpsrv2FromState }	{ Epsrv2NodeType 4 }	The previous state of the EPSR domain.
{ atEpsrv2Current State }	{ Epsrv2NodeType 5 }	The current state of the EPSR domain.
{ atEpsrv2ControlVlanId }	{ Epsrv2NodeType 6 }	The VLAN identifier for the control VLAN.
{ Epsrv2PrimaryIfIndex }	{ Epsrv2NodeType 7 }	The IfIndex of the primary interface.
{ atEpsrv2PrimaryIfState }	{ Epsrv2NodeType 8 }	The current state of the primary interface.
{ atEpsrv2SecondaryIfIndex }	{ Epsrv2NodeType 9 }	The IfIndex of the secondary interface.
{ atEpsrv2SecondaryIfState }	{ Epsrv2NodeType 10 }	The state of the secondary interface.
TEXTUAL CONVENTIONS		
{ atEpsrv2NodeState }		The trap states that can be advertised for an EPSR domain node. The following states are defined: ■ idle (1) ■ complete (2) ■ failed (3) ■ linksUp (4) ■ linksDown (5) ■ preForward (6) ■ unknown (7).

Table 13: atEpsrv2Objects Defined in the AT-EPSRV2 MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
{ atEpsrv2InterfaceState }		The trap states that can be advertised for an EPSR interface. The following states are defined: ■ unknown (1) ■ down (2) ■ blocked (3) ■ forward (4).

AT-FIBER-MONITORING-MIB

See Table 14 for the list of objects defined in the AT-HHM-MIB. These objects manage definition and notification definitions for Fiber Monitoring. Objects in this group have the object identifier **atFiberMon** { sysinfo 27 } OID 1.3.6.1.4.1.207.8.4.4.3.27.

Table 14: Objects defined in AT-FIBER-MONITORING-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atFiberMon	{ sysinfo 27 } 1.3.6.1.4.1.207.8.4.4.3.27	This MIB contains managed objects definition and notification definitions for AT Fiber Monitoring.
atFiberMonNotifications	{ atFiberMon 0 }	Notification objects for Fiber Monitoring alarms.
atFiberMonAlarmSetNotify	{ atFiberMonNotifications 1 }	A notification generated when the monitored received optical power of an SFP goes outside its alarm threshold.
atFiberMonAlarmClearedNotify	{ atFiberMonNotifications 2 }	A notification generated when the monitored received optical power of an SFP returns to an acceptable value. This can occur because the power has returned to its previous level or the comparison baseline has adjusted to the new level.
atFiberMonTrapVariable	{ atFiberMon 1 }	Notification variable.
atFiberMonReading	{ atFiberMonTrapVariable 1 }	The current received optical power reading for the port specified in the trap (x0.0001mW).
atFiberMonConfigTable	{ atFiberMon 2 }	It contains the objects used to configure the operation of fiber monitoring.
atFiberMonConfigEntry	{ atFiberMonConfigTable 1 }	Entry for the table. It contains configurable options for fiber monitoring for an interface.
atFiberMonIfname	{ atFiberMonConfigEntry 1 }	The name of the interface where the pluggable is in.
atFiberMonEnable	{ atFiberMonConfigEntry 2 }	Enable or disable fiber monitoring.
atFiberMonInterval	{ atFiberMonConfigEntry 3 }	Fiber monitoring polling interval in seconds.
atFiberMonSensitivity	{ atFiberMonConfigEntry 4 }	The sensitivity for fiber monitoring. Acceptable mode strings and value ranges: ■ low - 2dB ■ medium - 1dB ■ high - 0.5dB ■ highest - 0.0025mW ■ fixed <25 - 65535> - units of 0.0001mW ■ relative <0.00 - 10.00> - units of dB. Examples: ■ If you want to set the sensitivity to medium level, then you enter word **medium**. ■ If you want to set it to a fixed value, then you enter string **fixed 1000**. ■ If you want to set to a relative value, then you enter string **relative 1.20**.

Table 14: Objects defined in AT-FIBER-MONITORING-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atFiberMonBaseline	{ atFiberMonConfigEntry 5 }	The baseline value for fiber mornitoring. Acceptable mode string and value ranges: ■ average <12-150> - number of readings used to calculate moving average. ■ fixed <1-65535> - fixed baseline value in units of 0.0001mW. For example: ■ If you want to use a moving average of 12 readings as the baseline, you enter **average 12**. ■ If you want to use a fixed value, though it is not recommended, you enter **fixed 500**.
atFiberMonInterval	{ atFiberMonConfigEntry 6 }	Actions to take when an alarm occurs: Acceptable values are: 1. Log a message only. 2. Send snmp trap and log. 3. Shut down the interface and log. 4. Send trap, shutdown and log.
atFiberMonStateTable	{ atFiberMon 3 }	This table contains the information of fiber monitoring state for SFPs.
atFiberMonStateEntry	{ atFiberMonStateTable 1 }	Entry for fiber monitoring state table. It contains state values for a fiber channel.
atFiberMonIfindex	{ atFiberMonStateEntry 1 }	The index of the interface where the pluggable is in.
atFiberMonChannel	{ atFiberMonStateEntry 2 }	The channel number of the pluggable.
atFiberMonIlfDescription	{ atFiberMonStateEntry 3 }	Interface description. Normally it would be the name of the interface.
atFiberMonActualBaseline	{ atFiberMonStateEntry 4 }	The calculated baseline value for the link, in units of 0.0001mW.
atFiberMonThreshold	{ atFiberMonStateEntry 5 }	Threshold value for alarm to occur, in units of 0.0001mW.
atFiberMonReadingHistory	{ atFiberMonStateEntry 6 }	The reading history. It is a string containing last 12 readings.
atFiberMonMinReading	{ atFiberMonStateEntry 7 }	The minimum reading of rx power, in units of 0.0001mW.
atFiberMonMaxReading	{ atFiberMonStateEntry 8 }	The maximum reading of rx power, in units of 0.0001mW.
atFiberMonLastReading	{ atFiberMonStateEntry 9 }	The most recent reading of rx power, in units of 0.0001mW. -1 is returned if no readings have been taken yet.

AT-FILEv2-MIB

This MIB contains objects for displaying and managing file content of:

- Flash.
- SD cards.
- NVS on local and stacked devices.
- Copying, moving and deleting files from local, stacked and remote sources.
- USB storage devices.

The objects reside in the module **atFilev2** { modules 600 }, OID 1.3.6.1.4.1.207.8.4.4.4.600, organized in the following groups:

- The file operation devices - object for various devices supported for file operations.
- The SD card table - information about the SD Cards configured on the device.
- The File Info Table - information about all files, including pathnames, that are present on the device.
- The USB storage device table - information about the USB storage device configured on the device.

To see how to use these MIB objects to upgrade to a new software version and boot configuration file, see the procedure “Copy a File to or from a TFTP Server” in the SNMP Feature Overview and Configuration Guide.

Table 15: Objects defined in AT-FILEv2-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atFilev2	{ modules 600 } 1.3.6.1.4.1.207.8.4.4.4.600	MIB containing objects for listing and managing files.
atFilev2FileOperation	{ atFilev2 3 }	Collection of file operation objects available for configuration, to enable copying, moving and deleting files.
atFilev2SourceStackID	{ atFilev2Operation 1 }	Specifies the Stack ID of the source file. Set an integer corresponding to the stack ID of the stack member to use as the source. For devices that are not capable of being stacked, set with the value 1. This value is ignored if the source device is set to TFTP.

Table 15: Objects defined in AT-FILEv2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atFilev2SourceDevice	{ atFilev2Operation 2 }	Specifies the source device for the file to be copied. Valid values are 1 to 5. Set a value that corresponds with the various devices, as below: ■ Flash - default ■ Card ■ NVS ■ TFTP ■ USB. For moving files you cannot use TFTP as source or destination. For deleting files, the source cannot be TFTP. You must fully configure all required parameters before an operation can commence. Where a TFTP operation is configured, an IP address must also be set via atFilev2TftpIPAddr. To copy a file from TFTP to Flash, use 4 for source and 1 for destination.
atFilev2SourceFilename	{ atFilev2Operation 3 }	Specifies the **filename** of the source file to copy, move or delete. Include any path as required, but the storage type is not necessary. For example, to copy the file **latest.cfg** from the backupconfigs/routers directory on the TFTP server, you would set: backupconfigs/routers/latest.cfg.
atFilev2DestinationStackID	{ atFilev2Operation 4 }	Specifies the Stack ID for the destination file. For devices that are not capable of being stacked, set with the value 1. This value is ignored if the destination device is set to TFTP, or if a deletion operation is carried out.
atFilev2DestinationDevice	{ atFilev2Operation 5 }	Specifies the destination device for the files to be copied into. Valid values are 1 to 5. Set a value that corresponds with the various devices, as below: ■ Flash - default ■ Card ■ NVS ■ TFTP ■ USB. For copying files, you may use any combination of devices for the source and destination, except for copying from TFTP to TFTP. For moving files you cannot use TFTP as source or destination. For deleting files, this object is ignored. You must fully configure all required parameters before an operation can commence. Where a TFTP operation is configured, an IP address must also be set via atFilev2TftpIPAddr. To copy a file from TFTP to Flash, use 4 for source and 1 for destination.

Table 15: Objects defined in AT-FILEv2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atFilev2DestinationFilename	{ atFilev2Operation 6 }	Specifies the destination **filename** of the file to be copied or moved. Include any path as required, but the storage type is not necessary. The destination **filename** does not need to be the same as the source **filename**, and this object is ignored for file deletion operations. For example, to copy a release file from the TFTP server to the backup release directory on Flash, you would set:backuprelease/latest.rel. **Note:** If the destination is set to Flash, card or NVS, any file at the destination that shares the destination **filename** will be overwritten by a move or copy operation.

Table 15: Objects defined in AT-FILEv2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atFilev2CopyBegin	{ atFilev2Operation 7 }	Represents the status of the copy file operation, in the form of octet string. A read on this object can return several possible values, depending on the current status of the system and the various file operation objects: ■ idle - There is no file operation in progress and all required objects have been set correctly. Setting a '1' to this object will begin the file copy. ■ Error codes: [1-7] - A copy operation cannot be started until these errors are resolved. See below for key. ■ [action]ing x [--> y] - A file operation is currently in progress. You cannot start another operation while the object is returning this value. ■ [action] x [--> y] success - The last copy, move or delete operation was successfully completed. ■ [action] x [--> y] failure: [err] - The last copy, move or delete operation failed, with the error message attached. Common failures include lack of space on the destination file system, incorrect source file names or communication errors with remote services. Upon reading a success or failure message, the message will be cleared and the next read will result in either an 'idle' message or an 'Error codes' message if not all required objects have been correctly set. If the read returned 'idle', a new file operation can now be started. Following are possible values returned as Error codes for file copy: ■ atFilev2SourceDevice has not been set ■ atFilev2SourceFilename has not been set ■ atFilev2DestinationDevice has not been set ■ atFilev2DestinationFilename has not been set ■ atFilev2SourceDevice and atFilev2DestinationDevice are both set to TFTP ■ the combination of source device, stackID and filename is the same as the destination device, stackID and filename (i.e. it is not valid to copy a file onto itself) ■ TFTP IP address has not been set and TFTP has been set for one of the devices. TFTP IP address has not been set and TFTP has been set for one of the devices Provided all above requirements are met, immediately upon executing the SNMP set, the device will indicate that it was a success. The actual file copy itself will be started and continue on the device until it has completed. For large files, operations can take several minutes to complete. Subsequent reads of the object will return one of messages shown in the first table, to allow for tracking of the progress of the copy operation.

Table 15: Objects defined in AT-FILEv2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atFilev2MoveBegin	{ atFilev2Operation 8 }	Represents the status of the move file operation, in the form of octet string. A read on this object can return several possible values, depending on the current status of the system and the various file operation objects: ■ idle - There is no file operation in progress and all required objects have been set correctly. Setting a '1' to this object will begin the file move. ■ Error codes: [1-6] - A move operation cannot be started until these errors are resolved. See below for key. ■ [action]ing x [--> y] - A file operation is currently in progress. You cannot start another operation while the object is returning this value. ■ [action] x [--> y] success - The last copy, move or delete operation was successfully completed. ■ [action] x [--> y] failure: [err] - The last copy, move or delete operation failed, with the error message attached. Common failures include lack of space on the destination file system, incorrect source file names or communication errors with remote services. Upon reading a success or failure message, the message will be cleared and the next read will result in either an 'idle' message or an 'Error codes' message if not all required objects have been correctly set. If the read returned 'idle', a new file operation can now be started. Following are possible values returned as Error codes for file move: ■ atFilev2SourceDevice has not been set ■ atFilev2SourceFilename has not been set ■ atFilev2DestinationDevice has not been set ■ atFilev2DestinationFilename has not been set ■ either atFilev2SourceDevice or atFilev2DestinationDevice are set to TFTP ■ the combination of source device, stackID and filename is the same as the destination device, stackID and filename (i.e. it is not valid to move a file onto itself. Provided all above requirements are met, immediately upon executing the SNMP set, the device will indicate that it was a success. The actual file move itself will be started and continue on the device until it has completed. For large files, operations can take several minutes to complete. Subsequent reads of the object will return one of messages shown in the first table, to allow for tracking of the progress of the move operation.

Table 15: Objects defined in AT-FILEv2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atFilev2DeleteBegin	{ atFilev2Operation 9 }	Represents the status of the delete file operation, in the form of octet string. A read on this object can return several possible values, depending on the current status of the system and the various file operation objects: ■ idle - There is no file operation in progress and all required objects have been set correctly. Setting a '1' to this object will begin the file deletion. ■ Error codes: [1-3] - A delete operation cannot be started until these errors are resolved. See below for key. ■ [action]ing x [--> y] - A file operation is currently in progress. You cannot start another operation while the object is returning this value. ■ [action] x [--> y] success - The last copy, move or delete operation was successfully completed. ■ [action] x [--> y] failure: [err] - The last copy, move or delete operation failed, with the error message attached. Common failures include lack of space on the destination file system, incorrect source file names or communication errors with remote services. Upon reading a success or failure message, the message will be cleared and the next read will result in either an 'idle' message or an 'Error codes' message if not all required objects have been correctly set. If the read returned 'idle', a new file operation can be started. File deletion operations ignore the values set in the atFilev2DestinationStackID, atFilev2DestinationDevice and atFilev2DestinationFilename objects. The file deletion operation is equivalent to the CLI 'delete force [file]' command, so it is possible to delete any normally-protected system files, such as the currently configured boot release. Following are possible values returned as Error codes for file move: ■ atFilev2SourceDevice has not been set ■ atFilev2SourceFilename has not been set ■ atFilev2SourceDevicehas not been set to TFTP. Provided all above requirements are met, immediately upon executing the SNMP set, the device will indicate that it was a success. The actual file move itself will be started and continue on the device until it has completed. For large files, operations can take several minutes to complete. Subsequent reads of the object will return one of messages shown in the first table, to allow for tracking of the progress of the move operation.
atFilev2Flash_1	{ atFilev2Operation 10 }	Represents the Flash operation device object.
atFilev2Card_2	{ atFilev2Operation 11 }	Represents the Card operation device object.
atFilev2Nvs_3	{ atFilev2Operation 12 }	Represents the NVS operation device object.

Table 15: Objects defined in AT-FILEv2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atFilev2Tftp_4	{ atFilev2Operation 13 }	Represents the TFTP operation device object.
atFilev2TftpIPAddr	{ atFilev2Tftp_4 1 }	The IP address of the TFTP server that is to be used for the file copy process. This IP Address needs to be reachable from the device, or the file copy will fail.
atFilev2Usb	{ atFilev2Operation 15 }	Represents the USB storage device operation device object.
atFilev2SDcardTable	{ atFilev2 4 }	The SD Card table, containing information related to SD Cards. Indexed by: ■ atFilev2SDcardStackMemberId.
atFilev2SDcardEntry	{ atFilev2SDcardTable 1 }	Data pertaining to an SD Card instance.
atFilev2SDcardStackMemberId	{ atFilev2SDcardEntry 1 }	The index of the stack member hosting this SD Card.
atFilev2SDcardPresence	{ atFilev2SDcardEntry 2 }	This object indicates whether or not an SD Card is inserted into a slot. Possible values are: ■ notPresent (1) ■ present (2).
atFilev2InfoTable	{ atFilev2 5 }	The SD Card table, containing information related to SD Cards. Indexed by: ■ atFilev2SDcardStackMemberId.
atFilev2InfoEntry	{ atFilev2InfoTable 1 }	An entry in the list of files, containing information about a single file.
atFilev2InfoFilepath	{ atFilev2InfoEntry 1 }	The full path and name of the file. Files are sorted in alphabetical order and any filepath that is longer than 112 characters will not be displayed due to SNMP Object Identifier length limitations.
atFilev2InfoFileSize	{ atFilev2InfoEntry 2 }	The size of the file in bytes.
atFilev2InfoFileCreationTime	{ atFilev2InfoEntry 3 }	File creation time in the form <MMM DD YYYY HH:MM:SS>. For example, Sep 7 2008 06:07:54.
atFilev2InfoFileIsDirectory	{ atFilev2InfoEntry 4 }	This object will return the value TRUE if the entry is a directory, or FALSE if it is not.
atFilev2InfoFileIsReadable	{ atFilev2InfoEntry 5 }	This object will return the value TRUE if the file is readable, or FALSE if it is not.
atFilev2InfoFileIsWriteable	{ atFilev2InfoEntry 6 }	This object will return the value TRUE if the file is writeable, or FALSE if it is not.
atFilev2InfoFileIsExecutable	{ atFilev2InfoEntry 7 }	This object will return the value TRUE if the file is executable, or FALSE if it is not.
atFilev2USBMediaTable	{ atFilev2 6 }	The USB storage device table, containing information related to USB storage devices.
atFilev2USBMediaEntry	{ atFilev2USBMediaTable 1 }	Data pertaining to a USB storage device instance.
atFilev2USBMediaStackMemberId	{ atFilev2USBMediaEntry 1 }	The index of the stack member hosting this USB media. For devices that are not capable of being stacked, this object will always return the value 1.

Table 15: Objects defined in AT-FILEv2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atFilev2USBMediaPresence	{ atFilev2USBMediaEntry 2 }	This object indicates whether or not a USB storage device is inserted in a slot. Possible values are: ■ notPresent (1) ■ present (2).
atFilev2FileViewer	{ atFilev2 7 }	The set of objects in this section allow for a view of the files on this managed device that is reminiscent of the view that a shell user on a device would have, with the concepts of current working directory, changing directories and viewing files within a directory all supported. This section obsoletes the first two attempts at creating a view of files for the reason that performance and functionality of the original views was not sufficient. In particular, trying to create a view of all files in the device, sorted by path and file name, proved to be very difficult when scaling considerations were taken into account. **Note:** In order to use this group of objects the ability to write MIB variables is required.
atFilev2FileViewerStackId	(atFilev2FileViewer 1 }	The stack ID of the stack member for which files will be displayed in the FileViewer table. For devices that are not capable of being stacked, this variable will always read as 1, and will cause an error on being written to with any value other than 1. Write this variable with the stack ID of the stack member for which a view of files is required. If the stack member doesn't exist, an error will be returned. For a chassis switch, it corresponds to the card ID. **Note:** The other variables specifying the files to view will not be altered by changing the stack ID, which means that the file view table could be empty if a non-existant device or path has been referenced previously.

Table 15: Objects defined in AT-FILEv2-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atFilev2FileViewerDevice	{ atFilev2FileViewer 2 }	The file system device for which files will be displayed in the FileViewer table. The values supported for this variable are identical to the values for other variables in the MIB, although not all values will actually result in the display of files. The different devices and whether they will result in the display of files are: ■ Flash - Onboard Flash ■ Card - Removable SD card ■ NVS - Onboard battery backed RAM ■ FTP - not supported ■ USB - Removable USB media. **Note:** Setting this variable to a unsupported value will result in an error, but setting to a value that is supported but on a device that doesn't contain that type of device will not. However, no files will be displayed in the File Viewer table in this case.
atFilev2FileViewerCurrentPath	{ atFilev2FileViewer 3 }	The file system path for which files will be displayed in the FileViewer table. This path will always read as a full pathname starting with the '/' character. Setting this variable will specify a new directory for which files will be displayed. The path specified must be the full path, relative setting of path does not work. Only paths with invalid characters in them will cause an error, paths specifying non-existent directories will be accepted, but no files will be displayed in the File Viewer table in this case.
atFilev2FileViewerTable	{ atFilev2FileViewer 4 }	The file system path for which files will be displayed in the FileViewer table. This path will always read as a full pathname starting with the '/' character. Setting this variable will specify a new directory for which files will be displayed. The path specified must be the full path, relative setting of path does not work. Only paths with invalid characters in them will cause an error, paths specifying non-existent directories will be accepted, but no files will be displayed in the File Viewer table in this case.
atFilev2FileViewerEntry	{ atFilev2FileViewerTable 1 }	An entry in the list of files, containing information about a single file.
atFilev2FileViewerName	{ atFilev2FileViewerEntry 1 }	The name of the file. Files are sorted in alphabetical order, and any name that is longer than 112 characters will not be displayed due to SNMP OID length limitations.
atFilev2FileViewerSize	{ atFilev2FileViewerEntry 2 }	The size of the file in bytes.
atFilev2FileViewerCreationTime	{ atFilev2FileViewerEntry 3 }	File creation time in the form <MMM DD YYYY HH:MM:SS>. For example, Sep 7 2008 06:07:54.
atFilev2FileViewerIsDirectory	{ atFilev2FileViewerEntry 4 }	Returns TRUE if the entry is a directory, FALSE otherwise.

Table 15: Objects defined in AT-FILEv2-MIB (continued)

OBJECT		OBJECT IDENTIFIER	DESCRIPTION
	atFilev2FileViewerIsReadable	{ atFilev2FileViewerEntry 5 }	Returns TRUE if the file is readable, FALSE otherwise.
	atFilev2FileViewerIsWriteable	{ atFilev2FileViewerEntry 6 }	Returns TRUE if the file is writeable, FALSE otherwise.
	atFilev2FileViewerIsExecutable	{ atFilev2FileViewerEntry 7 }	Returns TRUE if the file is executable, FALSE otherwise.

AT-G8032v2-MIB

These objects manage Ethernet Protection Ring Switching (EPRS). Objects in this group have the object identifier **atG8032v2** { modules 604 }, OID 1.3.6.1.4.1.207.8.4.4.4.604. See Table 16 for the list of objects defined in the AT-G8032v2-MIB.

Table 16: Objects defined in AT-G8032v2-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atG8032v2	{ modules 604 } 1.3.6.1.4.1.207.8.4.4.4.604	This MIB contains managed objects for EPRS.
TEXTUAL CONVENTIONS		
AtG8032v2InstanceState		Defines the EPRS Instance states that are sent in G8032 State Notification Traps: ■ unknown(1), ■ init(2), ■ idle(3), ■ protection(4), ■ manualSwitch(5), ■ forcedSwitch(6), ■ pending(7)
atG8032v2Notifications	{ atG8032v2 0 }	Object notifictions
atG8032v2InstanceNotify	{ atG8032v2Notifications 1 }	G8032 ERP Instance state transition notification.
atG8032v2SystemAlarmNotify	{ atG8032v2Notifications 2 }	G8032 ERP system alarm transition notification. Indicates whether any ERP instance is in a state that is considered to be an alarm condition.
atG8032v2NotificationVariable	{ atG8032v2 1 }	Notification variables.
atG8032v2NotificationInstanceName	{ atG8032v2NotificationVariable 1 }	Assigned name of the G8032 ERP Instance.
atG8032v2NotificationInstanceFromState	{ atG8032v2NotificationVariable 2 }	Defined state that a G8032 ERP instance is transitioning from.
atG8032v2NotificationInstanceCurrentState	{ atG8032v2NotificationVariable 3 }	Defined current state that a G8032 ERP instance is transitioning to.
atG8032v2NotificationSystemAlarmState	{ atG8032v2NotificationVariable 4 }	Has value of 1 (true) if one or more G8032 ERP instance(s) are in alarm state, else has value of 2 (false).

AT-HHM-MIB

See Table 17 for the list of objects defined in the AT-HHM-MIB. These objects manage Hardware Health Monitoring notification generated log messages. Objects in this group have the object identifier **atwhealthMon** { sysinfo 24 } OID 1.3.6.1.4.1.207.8.4.4.3.24.

Table 17: Objects defined in AT-HHM-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
athwhealthMon	{ sysinfo 24 } 1.3.6.1.4.1.207.8.4.4.3.24	The AT Hardware Health Monitoring MIB.
atHhmNotifications	{ atHwHealthMon 0 }	Objects for notifications.
atHhmLogMessage	{ atHhmNotifications 1 }	A notification generated when Hardware Health Monitoring generates a new log message.
atHhmNotificationVariables	{ atHwHealthMon 1 }	Objects for log messages.
atHhmLogMessage	{ atHhmNotificationVariables 1 }	The most recent log message generated by Hardware Health Monitoring.

AT-IP-MIB

This MIB contains objects for Allied Telesis specific IP address management, see Table 18. The objects reside in the module **atIpMib** { modules 602 }, OID 1.3.6.1.4.1.207.8.4.4.4.602.

Table 18: Objects defined in AT-IP-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atIpMib	{ modules 602 } 1.3.6.1.4.1.207.8.4.4.4.602	MIB containing objects for IP addressing management.
AtIpAddressAssignmentType	Textual Convention	Object containing conditional coded values for the IP address assignment type being applied to the interface, referred to by objects in this MIB. The possible values and explanation are: ■ notSet (0) - indicates that the IP address assignment type has not yet been configured. This value can only ever be read. ■ primary (1) - indicates that the address is a primary IP address; only one primary address is allowed per interface. ■ secondary (2) - indicates that the address is a secondary IP address; any number of secondary IP addresses may be applied.
AtIpAddressTable	{ atIpMib 1 }	A table containing mappings between primary or secondary IP addresses, and the interfaces they are assigned to. Indexed by: ■ atIpAddressAddrType ■ atIpAddressAddr.
AtIpAddressEntry	{ AtIpAddressTable 1 }	Information about the address mapping for a particular interface.
atIpAddressAddrType	{ AtIpAddressEntry 1 }	An indication of the IP version of 'atIpAddressAddr'.
atIpAddressAddr	{ AtIpAddressEntry 2 }	The IP address to which this entry's addressing information pertains. The address type of this object is specified in object 'atIpAddressAddrType'.
atIpAddressPrefixLen	{ AtIpAddressEntry 3 }	An integer, specifying the prefix length of the IP address represented by this entry.
atIpAddressLabel	{ AtIpAddressEntry 4 }	The name assigned to the IP address represented by this entry.
atIpAddressIfIndex	{ AtIpAddressEntry 5 }	The index value that uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index corresponds to the interface identified by the same value of the IF-MIB's ifIndex.
atIpAddressAssignmentType	{ AtIpAddressEntry 6 }	The IP address assignment type for this entry (primary or secondary), as described in the Textual Convention 'AtIpAddressAssignmentType'.

Table 18: Objects defined in AT-IP-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atIpAddressRowStatus	{ AtIpAddressEntry 7 }	The current status of the IP address entry. The following values may be returned when reading this object: ■ active (1) The IP address is currently mapped to an interface and is valid. ■ notReady (3) The IP address is currently partially configured and is not mapped to an interface. The following values may be written to this object: ■ active (1) An attempt will be made to map the IP address to the configured interface. ■ createAndWait (5) An attempt will be made to create a new IP address entry. ■ destroy (6) The IP address setting will be removed from the device. An entry cannot be made active until its atIpAddressPrefixLen, atIpAddressIfIndex and atIpAddressAssignmentType objects have been set to valid values.

AT-LICENSE-MIB

The AT-LICENSE-MIB contains objects for managing the AlliedWare Plus™ Operating System software licenses: listing applied software licenses, adding new licenses and deleting existing licenses, see Table 19. The objects in this group have the object identifier **license** { sysinfo 22 }, OID 1.3.6.1.4.1.207.8.4.4.3.22, and are organized in the following groups:

- Base Software License Table - a table containing the installed base software licenses on the device.
- Installed Software License Table - a list of installed software licenses; used also to remove software license from the device.
- Available Software Features Table.
- LicenseNew - Objects used to install a new license.
- LicenseStackRemove - Objects used to remove a license across a stack of devices.

Table 19: Objects defined in AT-LICENSE-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
license	{ sysinfo 22 } 1.3.6.1.4.1.207.8.4.4.3.22	MIB containing objects for listing applied software licenses, adding new licenses, and deleting existing licenses.
baseLicenseTable	{ license 1 }	Table containing information about base software licenses installed on a device. Indexed by: ■ baseLicenseStkId.
baseLicenseEntry	{ baseLicenseTable 1 }	Information about a single license installed on the device.
baseLicenseStkId	{ baseLicenseEntry 1 }	The stack member ID of the device hosting the license.
baseLicenseName	{ baseLicenseEntry 2 }	The name of the base license.
baseLicenseQuantity	{ baseLicenseEntry 3 }	The number of licenses issued for this entry.
baseLicenseType	{ baseLicenseEntry 4 }	The type of base license issued.
baseLicenseIssueDate	{ baseLicenseEntry 5 }	The date of issue of the base license.
baseLicenseExpiryDate	{ baseLicenseEntry 6 }	The expiry date of the base license.
baseLicenseFeatures	{ baseLicenseEntry 7 }	The feature set that this license enables, in the format of an octet string. Each bit in the returned octet string represents a particular feature that can be license-enabled. The bit position within the string maps to the feature entry with the same index, in licenseFeatureTable: ■ binary '1' indicates that the feature is included in the license ■ binary '0' indicates that the feature is not included in the license.
licenseTable	{ license 2 }	Table containing information about software licenses installed on the device. Indexed by: ■ licenseStackId ■ licenseIndex.
licenseEntry	{ licenseTable 1 }	Information about a single installed software license on the device.
licenseStackId	{ licenseEntry 1 }	The stack member ID of the device hosting the license.
licenseIndex	{ licenseEntry 2 }	The index number of the license entry.

Table 19: Objects defined in AT-LICENSE-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
licenseName	{ licenseEntry 3 }	The name of the license.
licenseCustomer	{ licenseEntry 4 }	The name of the customer of the license.
licenseQuantity	{ licenseEntry 5 }	The number of licenses issued for this entry.
licenseType	{ licenseEntry 6 }	The type of license issued.
licenseIssueDate	{ licenseEntry 7 }	The date of issue of the license.
licenseExpiryDate	{ licenseEntry 8 }	The expiry date of the license.
licenseFeatures	{ licenseEntry 9 }	The feature set that this license enables, in the format of octet string. Each bit in the returned octet string represents a particular feature that can be license-enabled. The bit position within the string maps to the feature entry with the same index, in licenseFeatureTable: ■ binary '1' indicates that the feature is included in the license ■ binary '0' indicates that the feature is not included in the license.
licenseRowStatus	{ licenseEntry 10 }	The current status of the license. The following values may be returned when reading this object: ■ active (1) - the license is currently installed and valid ■ notInService (2) - the license has expired or is invalid The following value may be written to this object: ■ destroy (6) - the license will be removed from the device. This may result in some features being disabled. **Note:** a stacked device that has a license deleted may not be able to rejoin the stack after reboot, unless the license is also deleted on all other devices in the stack.
licenseFeatureTable	{ license 3 }	Table containing all available Software Features. A feature must be license-enabled to be utilized on the device.
licenseFeatureEntry	{ licenseFeatureTable 1 }	Information about a single feature that must be license-enabled in order to be utilized on the device.
licenseFeatureIndex	{ licenseFeatureEntry 1 }	The index number of the feature which must be license-enabled.
licenseFeatureName	{ licenseFeatureEntry 2 }	The name of the feature under licensing control.
licenseFeatureStkMembers	{ licenseFeatureEntry 3 }	The set of stack members on which the feature is enabled, in the format of an octet string. Each bit in the string maps to an individual stacking member, e.g. bit one represents stacking member one, bit two represents stacking member two. ■ a bit value of '1' indicates that the applicable feature is enabled on the matching device ■ a bit value of '0' indicates that the feature is disabled.
licenseNew	{ license 4 }	Group of objects available for updates, used when installing a new software license on the device.
licenseNewStackId	{ licenseNew 1 }	The ID of the stacking member upon which the new license is to be installed. The value zero (0) indicates that the license should be applied to all stack members.
licenseNewName	{ licenseNew 2 }	The name of the new license to be installed.

Table 19: Objects defined in AT-LICENSE-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
licenseNewKey	{ licenseNew 3 }	The key for the new license to be installed.
licenseNewInstall	{ licenseNew 4 }	Used to install new licenses. Values can be: ■ true (1) ■ false (2). To commence installation, a valid license name and key must first have been set via the licenseNewName and licenseNewKey respectively. This object should then be set to the value true (1). If either the license name or key is invalid, the write operation will fail. Once installed, the software modules affected by any newly enabled features will automatically be restarted. Note that a stacked device that has a new license installed on it may not be able to rejoin the stack after reboot, unless the license is also added to all other devices in the stack. When read, the object will always return the value false (2).
licenseNewInstallStatus	{ licenseNew 5 }	The current status of the last license installation request. One of the following values is returned when reading this object: ■ idle (1) ■ processing (2) ■ success (3) ■ failed (4). When a stack license installation operation is complete the first read of this object will return either a success (3) or a failure (4) indication. Subsequent reads of this object will then return an idle (1) indication.
licenseStackRemove	{ license 5 }	Group of objects used when removing a software license across a stack of devices.
licenseStackRemoveName	{ licenseStackRemove 1 }	The name of the license to be removed from all devices across the stack, on which the license currently exists.
licenseStackRemoveExecute	{ licenseStackRemove 2 }	When set to the value true (1), the system will attempt to remove the named license from all devices across the stack on which the license currently exists. All devices in a stack must be from the same product family and the named license must activate the same feature set on all devices.
licenseStackRemoveStatus	{ licenseStackRemove 3 }	The current status of the last requested stack license removal request. One of the following values is returned when reading this object: ■ idle (1) ■ processing (2) ■ success (3) ■ failed (4). When a stack license removal operation is complete the first read of this object will return either a success (3) or failure (4) indication. Subsequent reads of this object will then return an idle (1) indication.

AT-LINKMON-MIB

See Table 20 for the list of objects defined in the AT-LINKMON-MIB. This contains objects for link monitoring used for collecting and reporting link metrics. Objects in this group have the object identifier **atLinkMon** { modules 606 } OID 1.3.6.1.4.1.207.8.4.4.4.606.

Table 20: Objects defined in AT-LINKMON-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atLinkMon	{ modules 606 } 1.3.6.1.4.1.207.8.4.4.4.606	The AT-LINKMON-MIB contains objects for link monitoring for collecting and reporting link metrics.
atLinkMon 1	{ atLinkMonProbeTable 1 }	A table of information describing the configured Link Monitoring probes. Upon device reboot, probe OID's will be re-assigned based on the alphabetical naming of probes in the startup configuration.
atLinkMonProbeID	{ atLinkMonProbeEntry 1 }	ID of the probe.
atLinkMonProbeName	{ atLinkMonProbeEntry 2 }	Name of the probe.
atLinkMonProbeType	{ atLinkMonProbeEntry 3 }	Type of probe, for example ICMP PING or HTTP GET.
atLinkMonProbeIPVersion	{ atLinkMonProbeEntry 4 }	IP version of the probe, for example IPv4 or IPv6.
atLinkMonProbeDestination	{ atLinkMonProbeEntry 5 }	The destination of the probe.
atLinkMonProbeEgressIf	{ atLinkMonProbeEntry 6 }	The interface that the probe packets should egress.
atLinkMonProbeEgressIfValid	{ atLinkMonProbeEntry 7 }	Indicates if the interface that the probe packets egress is valid.
atLinkMonProbeSource	{ atLinkMonProbeEntry 8 }	The source IP address or interface.
atLinkMonProbeSourceValid	{ atLinkMonProbeEntry 9 }	Indicates if the source IP address or interface is valid. This is only applicable for ICMP probes.
atLinkMonProbeDSCP	{ atLinkMonProbeEntry 10 }	The DSCP value to use when sending the packet. This is only applicable for ICMP probes.
atLinkMonProbePacketSize	{ atLinkMonProbeEntry 11 }	The size of a probe packet. This is only applicable for ICMP probes.
atLinkMonProbeInterval	{ atLinkMonProbeEntry 12 }	The number of milliseconds between sending out each probe. The range for an ICMP Ping probe is 100-10000. The range for an HTTP Get probe is 30000-3600000.
atLinkMonProbeSampleSize	{ atLinkMonProbeEntry 13 }	The number of probe results to use when calculating the latency and jitter metrics. This is only applicable for ICMP probes.
atLinkMonProbeEnabled	{ atLinkMonProbeEntry 14 }	Whether the probe is enabled or disabled. If it is enabled, then the device will attempt to send probes if the link is up. If it is disabled, then no probes are sent.
atLinkMon 2	{ atLinkMonProbeDetailTable 1 }	A table of information describing the latest Link Monitoring probe details.

Table 20: Objects defined in AT-LINKMON-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atLinkMonProbeDetailProbes Sent	{ atLinkMonProbeDetailEntry 1 }	The number of probes that have been sent.
atLinkMonProbeDetailProbe DetailLastTxTime	{ atLinkMonProbeDetailEntry 2 }	The time that the last probe was sent.
atLinkMonProbeDetailProbe DetailLastRxTime	{ atLinkMonProbeDetailEntry 3 }	The time that the device last successfully received a probe.
atLinkMon 3	{ atLinkMonProbeLatestMetricsTable }	A table of information describing the latest Link Monitoring probe metrics.
atLinkMonProbeMetric Latency	{ atLinkMonProbeMetricsEntry 1 }	The average latency based on the last samples in ms.
atLinkMonProbeLatestMetrics Jitter	{ atLinkMonProbeMetricsEntry 2 }	The average jitter based on the last samples in ms.
atLinkMonProbeLatestMetrics PktLoss	{ atLinkMonProbeMetricsEntry 3 }	The percentage of probes lost based on the last 100 probes. The value is given in tenths of a percent.
atLinkMonProbeLatestMetrics CnscPktLoss	{ atLinkMonProbeMetricsEntry 4 }	The number of consecutive probes that have been lost. A positive value means consecutive probes lost. A negative value means consecutive probes received.
atLinkMon 4	{ atLinkMonProbeHistoryTable }	A table of information describing historical Link Monitoring probe metrics.
atLinkMonProbeHistoryID	{ atLinkMonProbeHistoryEntry 1 }	The ID of the probe history collection.
atLinkMonProbeHistoryProbe Name	{ atLinkMonProbeHistoryEntry 2 }	The name of the probe the metrics are sampled from.
atLinkMonProbeLatestHistroy Interval	{ atLinkMonProbeHistoryEntry 3 }	The interval at which the metric data is sampled.
atLinkMonProbeHistory Buckets	{ atLinkMonProbeHistoryEntry 4 }	The maximum number of historical samples that re kept.
atLinkMonProbeHistoryLast SmpIID	{ atLinkMonProbeHistoryEntry 5 }	The ID of the latest sample recorded. This is always increasing to distinguish new samples.
atLinkMonProbeHistoryLast SmplTime	{ atLinkMonProbeMetricsEntry 6 }	The timestamp of the last sample.
atLinkMon 5	{ atLinkMonSampleTable }	A table of information describing the Link Monitoring metrics of a historical probe sample.
atLinkMonSampleBucket	{ atLinkMonSampleEntry 1 }	The bucket to read the sample data from. The first bucket has the oldest available sample. A bucket can have no data in it if less samples have been taken than there are buckets.
atLinkMonSampleLatency Sum	{ atLinkMonSampleLatencySumEntry 2 }	The sum of latency metric values calculated during the sample interval. Divide by the latency count for an average latency during the interval.
atLinkMonSampleLatency Count	{ atLinkMonSampleCountEntry 3 }	The number of latency metric values calculated in the sum.

Table 20: Objects defined in AT-LINKMON-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atLinkMonSampleJitterSum	{ atLinkMonSampleEntry 4 }	Sum of jitter metric values calculated during the sample interval. Divide by the jitter count for an average jitter during the interval.
atLinkMonSampleJitterCount	{ atLinkMonSampleEntry 5 }	The number of jitter metric values calculated in the sum.
atLinkMonSamplePktLoss Sum	{ atLinkMonSampleEntry 6 }	The sum of the packet loss metric values recorded during the sample interval. The value is given in tenths of a percent. Divide by packets transmitted for the average of the packet loss metric during the interval.
atLinkMonSamplePktsTx	{ atLinkMonSampleEntry 7 }	The number of probes sent during the sample interval.
atLinkMonSamplePktsRx	{ atLinkMonSampleEntry 8 }	The number of probes sent that received a reply during the same interval.

AT-LINKTRAP-MIB

See Table 21 for the list of objects defined in the AT-LINKTRAP-MIB. This contains objects for link up and down traps. Objects in this group have the object identifier **atLinkTrap** { sysinfo 25 } OID 1.3.6.1.4.1.207.8.4.4.3.25.

Table 21: Objects defined in AT-LINKTRAP-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atLinkTrap	{ sysinfo 25 } 1.3.6.1.4.1.207.8.4.4.3.25	The AT-LINKTRAP-MIB contains objects for link up and down traps.
atLinkDown	{ atLinkTrap 1 }	A trap generated when an interface is linked down.
atLinkUp	{ atLinkTrap 2 }	A trap generated when an interface is linked up.

AT-LOG-MIB

The AT Log MIB contains objects for listing log entries from the buffered and permanent logs, see Table 22. The object identifier **log** { modules 601 }, OID 1.3.6.1.4.1.207.8.4.4.4.601, organized in the following groups:

- Log Table - objects containing the information from log messages issued by the system, ordered from oldest to newest entry.
- Log Options - contains objects used to set up the log options configuration.

Table 22: Objects defined in AT-LOG-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
log	{ modules 601 } 1.3.6.1.4.1.207.8.4.4.4.601	MIB containing objects for listing log entries from the buffered and permanent logs.
logTable	{ log 1 }	A list of log entries from the source specified in the 'logSource' object. The list is ordered from oldest entry to newest entry. Indexed by: ■ logIndex.
logEntry	{ logTable 1 }	Information about a single log entry, from the source specified in the 'logSource' object.
logIndex	{ logEntry 1 }	An index integer. This index is not directly tied to any specific log entry. Over time, the log will grow larger and eventually older entries will be removed from the log.
logDate	{ logEntry 2 }	The date of the log entry. Data resides in the format octet string, in the form YYYY MMM DD, e.g. 2008 Oct 9.
logTime	{ logEntry 3 }	The time of the log entry. Data resides in the format octet string, in the form HH:MM:SS, e.g. 07:15:04.
logFacility	{ logEntry 4 }	The syslog facility that generated the log entry, in the format octet string. See the reference manual for more information.
logSeverity	{ logEntry 5 }	The severity level of the log entry, in the format octet string. Severities are given below: ■ emerg (Emergency, system is unusable) ■ alert (Action must be taken immediately) ■ crit (Critical conditions) ■ errr (Error conditions) ■ warning (Warning conditions) ■ notice (Normal, but significant, conditions) ■ info (Informational messages) ■ debug (Debug-level messages).
logProgram	{ logEntry 6 }	The program that generated the log entry, in the format octet string. See the reference manual for more information.
logMessage	{ logEntry 7 }	The message of the new log entry, in the format octet string.
logOptions	{ log 2 }	Contains objects used to set up the required log options configuration.
logSource	{ logOptions 1 }	An integer indicating the source from which the log entries are retrieved. The valid values are: ■ 1 - Buffered log (default) ■ 2 - Permanent log. This information is used when retrieving the logTable objects, and also specifies the log to be cleared when the 'clearLog' object is set.

Table 22: Objects defined in AT-LOG-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
logAll	{ logOptions 2 }	An integer indicating whether to display all log entries in the logTable objects, or not. The valid values are: ■ 0 - to display only the most recent log messages. This is the default. ■ 1 - to show all available log entries. **Note:** Choosing to display all log entries may result in delays of several seconds when accessing the logTable objects.
clearLog	{ logOptions 3 }	An integer indicating whether to clear the log that is specified by the 'logSource' object. Valid values are: ■ 0 - do not clear log ■ 1 - clear log.

AT-LOOPPROTECT-MIB

The atLoopProtect-MIB, see Figure 5 and Table 23, defines objects for managing Loop Protection objects and triggers. Objects in this group have the object identifier **atLoopProtect** { modules 4 }, OID 1.3.6.1.4.1.207.8.4.4.4.54.

Figure 5: The ATLoopProtect MIB Sub-tree

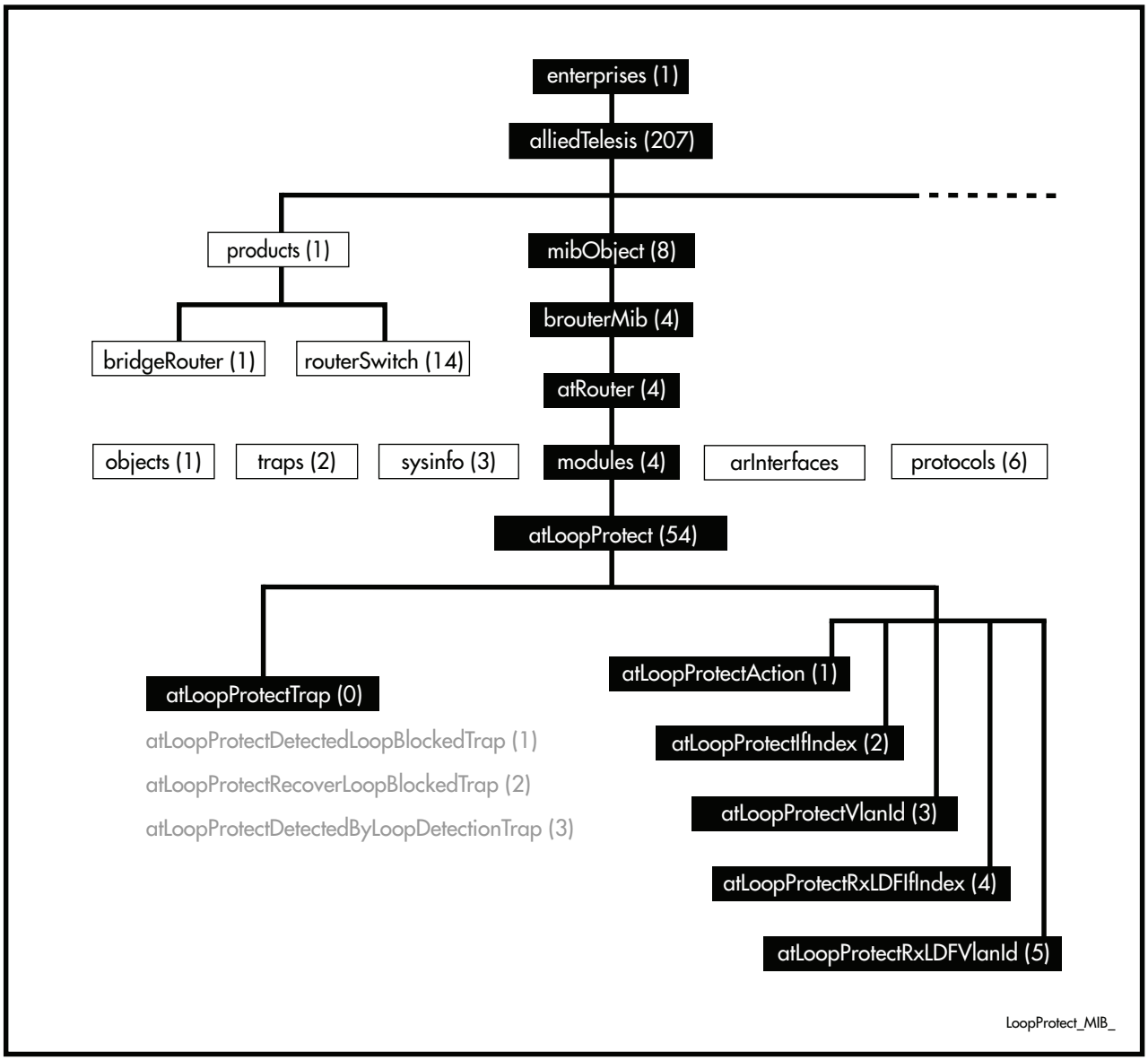


Table 23: Objects Defined in the AT-Loop Protect MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
{ atLoopProtect }	{ modules 54 } 1.3.6.1.4.1.207.8.4.4.4.54	The root of the Loop Protect object sub tree.
{ atLoopProtectTrap }	{ atLoopProtect0 }	The Loop Protection node state transition trap. List of traps (notifications) generated for Loop Protection.
{ atLoopProtectDetected LoopBlockedTrap }	{ atLoopProtectTrap1 }	Notification generated when the Loop Protection feature blocks an interface with a loop. The following bindings are associated with this trap: ■ atLoopProtectIfIndex ■ atLoopProtectVlanId ■ atLoopProtectAction.
{ atLoopProtectRecover LoopBlockedTrap }	{ atLoopProtectTrap2 }	Notification generated when the Loop Protection feature restores a blocked interface back to normal operation. The following bindings are associated with this trap: ■ atLoopProtectIfIndex ■ atLoopProtectVlanId ■ atLoopProtectAction.
{ atLoopProtectDetected ByLoopDetectionTrap }	{ atLoopProtectTrap3 }	Notification generated when the Loop Protection feature detects a loop by Loop Detection method. The following bindings are associated with this trap: ■ atLoopProtectIfIndex ■ atLoopProtectVlanId ■ atLoopProtectRxLDFIfIndex ■ atLoopProtectRxLDFVlanId.
{ atLoopProtectAction }	{ atLoopProtect1 }	The Action for the Loop Protection feature. The following values are defined: ■ atLoopProtectAction-LearnDisable (0) ■ atLoopProtectAction-LearnEnable (1) ■ atLoopProtectAction-PortDisable (2) ■ atLoopProtectAction-PortEnable (3) ■ atLoopProtectAction-LinkDown (4) ■ atLoopProtectAction-LinkUp (5) ■ atLoopProtectAction-VlanDisable (6) ■ atLoopProtectAction-VlanEnable (7).
{ atLoopProtectIfIndex }	{ atLoopProtect2 }	The interface on which the loop was detected.
{ atLoopProtectVlanId }	{ atLoopProtect3 }	The VLAN ID on which the loop was detected.
{ atLoopProtectRxLDFIfIndex }	{ atLoopProtect4 }	The interface on which the loop detection frame was received.
{ atLoopProtectRxLDFVlanId }	{ atLoopProtect5 }	The VLAN ID on which the loop detection frame was received.

AT-MAC-NOTIFICATION-MIB

AT-MAC-NOTIFICATION-MIB defines objects for configuration of the MAC notification feature. MAC notification is a mechanism to inform monitoring devices when there are MAC addresses learned or removed from the forwarding table of the monitored devices, see Table 24. Objects in this group have the object identifier **atMac** in { modules 607 }, OID 1.3.6.1.4.1.8.4.4.4.607

Table 24: Objects defined by the AT-MAC-NOTIFICATION-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atMac	{ modules 607 } (1.3.6.1.4.1.207.8.4.4.4.607)	This MIB file contains definitions of managed objects for the MAC notifications module.
atMacNotifications	{ atMac 0 }	This is for configuration of the MAC notifications.
atMacChangeNotification	{ atMacNotifications 1 }	This notification is sent if atMacChangeNotificationEnabled is true when: 1. there are enough MAC change events to fully occupy a maximum size SNMP trap message containing 23 events 2. the time elapsed from when the first MAC address change was queued is greater than the wait time denoted by atMacChangeNotificationInterval object.
atMacMoveNotification	{ atMacNotifications 2 }	This notification is sent when a MAC address is moved between two interfaces.
atMacThresholdNotification	{ atMacNotifications 3 }	This notification is sent when atMacUtilizationRate exceeds or equals to the atMacThresholdLimit for a given atMacUtilizationInstanceId.
atMacChangeObjects	{ atMac 1 }	This is for configuration of the MAC changes.
atMacChangeFeatureEnabled	{ atMacChangeObjects 1 }	Indicates whether the MAC change feature is currently running on the device. Setting this object to false(2) will disable the MAC change feature globally, thus disabling the feature at each interface. Setting this object to true(1) will start the MAC change feature if it is not already running. Once the mac-change feature is enabled, whether the feature is running at each interface is controlled by the atMacChangeIfTable.
atMacChangeNotificationEnabled	{ atMacChangeObjects 2 }	Indicates whether the MAC change notification is enabled. Setting this object to false(2) will disable the sending of atMacChangeNotification globally, but it will still archive the events in the atMacHistoryTable. Setting this object to true(1) will enable the sending of atMacChangeNotification globally, and MAC change events will be archived in the atMacHistoryTable.

Table 24: Objects defined by the AT-MAC-NOTIFICATION-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atMacChangeNotificationInterval	{ atMacChangeObjects 3 }	This object specifies the maximum interval of time between a MAC change event being queued and the generation of an atMacChangeNotification by the device. If the value of this object is equal to 0, the device will send atMacChangeNotification and archive the MAC change events in the atMacHistoryTable as soon as MAC addresses are learnt or removed. If the value of this object is greater than 0, the device will wait for a period of time equal to the value of this object before generating the atMacChangeNotification and archiving the MAC change events in the atMacHistoryTable.
atMacAddressesLearnt	{ atMacChangeObjects 4 }	Indicates the number of MAC addresses learnt by the device in MAC change events.
atMacAddressesRemoved	{ atMacChangeObjects 5 }	Indicates the number of MAC addresses removed from the forwarding table in MAC change events.
atMacChangeNotificationsSent	{ atMacChangeObjects 6 }	Indicates the number of atMacChangeNotification sent out by the device.
atMacHistTableMaxLength	{ atMacChangeObjects 7 }	The upper limit on the number of entries that the atMacHistoryTable may contain. A value of 0 will prevent any history from being retained.When this table is full, the oldest entry will be deleted and a new one will be created.
atMacHistoryTable	{ atMacChangeObjects 8 }	This table will archive the MAC change events generated by this device. The MAC change events are archived here even if the atMacChangeNotification is not actually sent.
atMacHistoryEntry	{ atMacHistoryTable 1 }	A MAC change message that was generated by this device. Each entry is indexed by a message index
atMacHistIndex	{ atMacHistoryEntry 1 }	An index that uniquely identifies a MAC change event previously generated by the device. This index starts at 1 and increases by one when a MAC change event occurs. When the index would increase beyond the maximum integer value of 4294967295, the agent will clear the existing events, and the indices restart from 1.

Table 24: Objects defined by the AT-MAC-NOTIFICATION-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atMacHistMacChangeMsg	{ atMacHistoryEntry 2 }	This object contains the information of a MAC change notification event. It consists of several tuples packed together in the format of '<tuple1><tuple2>...<tupleN><END>'. Each tuple consist of 11 octets in the format of '<operation><VLAN><MAC><dot1dBasePort> ' where <operation> is of size 1 octet and supports the values: 0 - End of MIB object. (Used only by <END> marker) 1 - MAC learnt. 2 - MAC removed. <VLAN> is VLAN number of the VLAN which the MAC address belonged to and has size of 2 octet. <MAC> is the Layer 2 MAC Address and has size of 6 octets. <dot1dBasePort> is the value of dot1dBasePort for the interface from which the MAC address is learnt or removed, and has size of 2 octets. The <END> marker is of size 1 octet and is always set to 0 so that it is identifiable from a regular MAC change operation.
atMacHistTimestamp	{ atMacHistoryEntry 3 }	The value of sysUpTime when the atMacChangeNotification containing the information denoted by the atMacHistMacChangeMsg object in this entry was generated.
atMacChangeIfTable	{ atMacChangeObjects 9 }	This table enables or disables the generation of MAC change events at each interface when MAC addresses are learnt or removed.
atMacChangeIfEntry	{ atMacChangeIfTable 1 }	Each entry contains the configuration for enabling the generation of MAC change events at each interface that supports this feature.
atMacAddrLearntNotificationEnabled	{ atMacChangeIfEntry 1 }	Indicates whether this interface is enabled to generate MAC change events when it learns a new MAC address. This variable has no effect when the value of atMacChangeFeatureEnabledobjectisfalse(2). Setting this object to true(1) enables the generation of MAC change events when this interface learns a new MAC address. Setting this object to false(2) disables the generation of MAC change events when this interface learns a new MAC address.

Table 24: Objects defined by the AT-MAC-NOTIFICATION-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atMacAddrRemovedNotificationEnabled	{ atMacChangeIfEntry 2 }	Indicates whether this interface is enabled to generate MAC change events when a MAC address which it learned previously is removed from the forwarding table. This variable has no effect when the value of atMacChangeFeatureEnabledobjectisfalse(2). Setting this object to true(1) enables the generation of MAC change events when a MAC address which this interface learned previously is removed from the forwarding table. Setting this object to false(2) disables the generation of MAC change events when a MAC address which this interface learned previously is removed from the forwarding table.
atMacMoveObjects	{ atMac 2 }	This is for configuration of the MAC addresses move information.
atMacMoveNotificationEnabled	{ atMacMoveObjects 1 }	Specifies whether atMacMoveNotification will or will not be sent when the MAC addresses are moved between the interfaces. Setting this object to true(1) enables the sending of atMacMoveNotification by the device. Setting this object to false(2) disables the sending of atMacMoveNotification by the device.
atMacMoveAddress	{ atMacMoveObjects 2 }	Indicates the MAC address that is moved between atMacMoveFromPortId and atMacMoveToPortId on atMacMoveVlanId.This object is instantiated only when atMacMoveNotificationEnabled is true(1) and a MAC address is moved between the interfaces.
atMacMoveVlanId	{ atMacMoveObjects 3 }	Indicates the VLAN on which the atMacMoveAddress is moved from atMacMoveFromPortId to atMacMoveToPortId.This object is instantiated only when atMacMoveNotificationEnabled is true(1) and a MAC address is moved between the interfaces.
atMacMoveFromPortId	{ atMacMoveObjects 4 }	The value of dot1dBasePort for the bridge port from which the atMacMoveAddress is moved to atMacMoveToPortId on atMacMoveVlanId.This object is instantiated only when atMacMoveNotificationEnabled is true(1) and a MAC address is moved between the interfaces.
atMacMoveToPortId	{ atMacMoveObjects 5 }	The value of dot1dBasePort for the bridge port to which the atMacMoveAddress is moved from atMacMoveFromPortId on atMacMoveVlanId.This object is instantiated only when atMacMoveNotificationEnabled is true(1) and a MAC address is moved between the interfaces.

Table 24: Objects defined by the AT-MAC-NOTIFICATION-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atMacMoveTime	{ atMacMoveObjects 6 }	The value of sysUpTime when a atMacMoveAddress is moved between atMacMoveFromPortId and atMacMoveToPortId.This object is instantiated only when atMacMoveNotificationEnabled is true(1) and a MAC address is moved between the interfaces.
atMacThresholdObjects	{ atMac 3 }	This is for configuration of the MAC Threshold.
atMacThresholdNotificationEnabled	{ atMacThresholdObjects 1 }	Specifies whether atMacThresholdNotification will or will not be sent when the forwarding table utilization exceeds or equals to atMacThresholdLimit. Setting this object to true(1) enables the sending of atMacThresholdNotification by the device. Setting this object to false(2) disables the sending of atMacThresholdNotification by the device.
atMacThresholdLimit	{ atMacThresholdObjects 2 }	Indicate the threshold limit of the forwarding table utilization.
atMacThresholdInterval	{ atMacThresholdObjects 3 }	The time interval at which forwarding table utilization is compared against atMacThresholdLimit.
atMacUtilizationTable	{ atMacThresholdObjects 4 }	atMacUtilizationTable specifies the forwarding table utilization information. This table is instantiated only when atMacThresholdEnabled is true(1). Entries in this table are updated at the end of every atMacThresholdInterval.
atMacUtilizationEntry	{ atMacUtilizationTable 1 }	A conceptual row containing forwarding table utilization maintained by switching engine (identified by atMacUtilizationInstanceId). Each switching engine managed by this MIB module can have at least one entry in this table.
atMacUtilizationInstanceId	{ atMacUtilizationEntry 1 }	Represents the instance of the physical mac address forwarding table.
atMacUtilizationVcsMemberId	{ atMacUtilizationEntry 2 }	Represents the Virtual Chassis Stacking member of the board of the physical mac address forwarding table, in a VCS environment. Otherwise it always represents a stand alone unit itself.
atMacUtilizationBayId	{ atMacUtilizationEntry 3 }	Represents the physical position of the board of the physical mac address forwarding tables.
atMacUtilizationEntries	{ atMacUtilizationEntry 4 }	Indicates the number of entries present in the forwarding table for the given atMacUtilizationInstanceId calculated at the end of atMacThresholdInterval.

Table 24: Objects defined by the AT-MAC-NOTIFICATION-MIB

OBJECT		OBJECT IDENTIFIER	DESCRIPTION
	atMacUtilizationRate	{ atMacUtilizationEntry 5 }	Indicates the utilization of the forwarding table for the given atMacUtilizationInstanceId calculated at the end of atMacThresholdInterval.
	atMacUtilizationTime	{ atMacUtilizationEntry 6 }	Indicates the sysUptime at which the atMacUtilizationRate is updated.

AT-MIBVERSION-MIB

The AT-MIBVERSION-MIB contains an object to display the last software release that contained changes to the supported AT Enterprise MIB definition files, see Table 25. Objects in this group have the object identifier **atMibsetVersion** { sysinfo 15 }, OID 1.3.6.1.4.1.207.8.4.4.3.15.

Table 25: Object defined in AT-MIBVERSION-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atMibVersion	{ sysinfo 15 } 1.3.6.1.4.1.207.8.4.4.3.15	This object returns a five digit integer which indicates the last software release that contained changes to the supported AT Enterprise MIB definition files. For example, If the currently loaded software release on the device is 5.3.1-0.3 but the Enterprise MIBs have not changed since 5.3.1-0.1, then the value returned will be 53101.

AT-NTP-MIB

This MIB contains objects for managing the Allied Telesis Network Time Protocol (NTP) configuration, see Table 26. The objects reside in the module **atNtp** { modules 502 }, OID 1.3.6.1.4.1.207.8.4.4.4.502, organized in the following groups:

- NTP Peer/Server Table - a table containing information on the Network Time Protocol (NTP) peers or server configurations in the system.
- Associations Table - a list of installed software; used also to remove software from the device.
- Status Table - objects in this group are not supported.

Table 26: Objects defined in AT-NTP-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atNtp	{ modules 502 } 1.3.6.1.4.1.207.8.4.4.4.502	MIB containing objects for configuring NTP.
atNtpPeerIndexNext	{ atNtp 6 }	The next available index number to be used for object 'atNtpPeerIndex'.
atNtpPeerTable	{ atNtp 7 }	Table containing information on the Network Time Protocol (NTP) peers or server configurations in the system. Indexed by: ■ atNtpPeerIndex.
atNtpPeerEntry	{ atNtpPeerTable 1 }	Information about a single NTP server or peer configuration.
atNtpPeerIndex	{ atNtpPeerEntry 1 }	The index number corresponding to a particular NTP server or peer configuration in the system. To create a new entry, the value of this object should be the same as that of the value of atNtpPeerIndexNext object, otherwise the entry creation will fail.
atNtpPeerNameAddr	{ atNtpPeerEntry 2 }	The host name, or the IP address of the NTP peer. When a new row (entry) is created, this object is set with a default of '0.0.0.0', and the management application should change it to a desired value by using a SET operation.
atNtpPeerMode	{ atNtpPeerEntry 3 }	The mode of the peer. Can be one of the following: ■ server (1) ■ peer (2) - default.
atNtpPeerPreference	{ atNtpPeerEntry 4 }	The values in this object specifies whether this peer is the preferred one. Valid values are 0 to 2: ■ 0 - unknown - default ■ 1 - not preferred ■ 2 - preferred. When the value is 'not preferred' (1) NTP chooses the peer with which to synchronize the time on the local system. If the object is set to 'preferred' (2) NTP will choose the corresponding peer to synchronize the time with.

Table 26: Objects defined in AT-NTP-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atNtpPeerVersion	{ atNtpPeerEntry 5 }	The NTP version the peer supports. Can be one of the following: ■ 0 - unknown - default ■ 1 - version 1 ■ 2 - version 2 ■ 3 - version 3 ■ 4 - version 4.
atNtpPeerKeyNumber	{ atNtpPeerEntry 6 }	The authentication key number. Default number is 0.
atNtpPeerRow Status	{ atNtpPeerEntry 7 }	The current status of this peer entry. The following values may be returned when reading this object: ■ active (1) - this value is returned on reading of this entry. ■ createAndGo (4) - this value is set by the management application when creating a new entry. ■ destroy (6) - value set by the management application when deleting the entry. When an entry is deleted, all subsequent entries in the table will be re-indexed.
atNtpAssociationTable	{ atNtp 10 }	Table containing information on the Network Time Protocol (NTP) associations. Indexed by: ■ atNtpAssociationIndex.
atNtpAssociationEntry	{ atNtpAssociationTable 1 }	Information about a single NTP server or peer configuration.
atNtpAssociationIndex	{ atNtpAssociationEntry 1 }	The index number corresponding to a particular NTP server or peer configuration in the system. To create a new entry, the value of this object should be the same as that of the value of atNtpPeerIndexNext object, otherwise the entry creation will fail.
atNtpAssociationPeerAddr	{ atNtpAssociationEntry 2 }	The host name, or the IP address of the NTP peer. When a new row (entry) is created, this object is set with a default of '0.0.0.0', and the management application should change it to a desired value by using a SET operation.
atNtpAssociationStatus	{ atNtpAssociationEntry 3 }	The status of this association. Can be one of the following: ■ master (synced) ■ master (unsynced) ■ selected ■ candidate ■ configured ■ unknown.
atNtpAssociationConfigured	{ atNtpAssociationEntry 4 }	The value in this object specifies whether the association is from configuration or not. Value can be: ■ configured ■ dynamic.
atNtpAssociationRefClkAddr	{ atNtpAssociationEntry 5 }	The IP Address for the reference clock.

Table 26: Objects defined in AT-NTP-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atNtpAssociationStratum	{ atNtpAssociationEntry 6 }	The stratum of the peer clock.
atNtpAssociationPoll	{ atNtpAssociationEntry 7 }	The time between NTP requests from the device to the server, in seconds.
atNtpAssociationReach	{ atNtpAssociationEntry 8 }	An integer that indicates the reachability status of the peer.
atNtpAssociationDelay	{ atNtpAssociationEntry 9 }	The round trip delay between the device and the server.
atNtpAssociationOffset	{ atNtpAssociationEntry 10 }	The difference between the device clock and the server clock.
atNtpAssociationDisp	{ atNtpAssociationEntry 11 }	The lowest measure of error associated with peer offset, based on delay, in seconds.
atNtpStatus	{ atNtp 11 }	Group of objects containing system status information. The objects in this group are not supported.
atNtpSysClockSync	{ atNtpStatus 1 }	Indicates whether the system clock is synchronized.
atNtpSysStratum	{ atNtpStatus 2 }	The stratum of the local clock.
atNtpSysReference	{ atNtpStatus 3 }	A display string, indicating the current synchronization source.
atNtpSysFrequency	{ atNtpStatus 4 }	The actual clock frequency, a numeric value in ‘Hz’ units
atNtpSysPrecision	{ atNtpStatus 5 }	A signed integer indicating the precision of the system clock, in seconds. The value is rounded to the next larger power of two. For example, a 50-Hz (20 ms) or 60-Hz (16.67 ms) power-frequency clock would be assigned the value -5 (31.25 ms), while a 1000-Hz (1 ms) crystal-controlled clock would be assigned the value -9 (1.95 ms).
atNtpSysRefTime	{ atNtpStatus 6 }	Indicates the local time when the local clock was last updated. If the local clock has never been synchronized, the value is zero. The values are in the format of an Octet string.
atNtpSysClkOffset	{ atNtpStatus 7 }	Indicates the offset of the local clock relative to the server clock, in milliseconds.
atNtpSysRootDelay	{ atNtpStatus 8 }	The total round trip delay, in milliseconds, to the primary reference source at the root of the synchronization subnet.
atNtpSysRootDisp	{ atNtpStatus 9 }	The maximum error, in milliseconds, relative to the primary reference source at the root of the synchronization subnet.

AT-PLUGGABLE-DIAGNOSTICS-MIB

This MIB contains objects for monitoring Optical Digital Diagnostics for pluggable devices, see Table 27. The objects identifier for this MIB is **atPluggableDiag** { sysinfo 28 }, OID 1.3.6.1.4.1.207.8.4.4.3.28, see "AT-SYSINFO-MIB" on page 121.

Table 27: Objects defined in AT-PLUGGABLE-DIAGNOSTICS-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atPluggableDiag	{ sysinfo (28) } 1.3.6.1.4.1.207.8.4.4.3.28	The AT-Pluggable Diagnostics MIB contains objects to retrieve the operational diagnostic information from installed SFP, SFP+, XFP and QSFP modules.
atPluggableDiagTable	{ atPluggableDiag 1 }	A table containing all operational diagnostics information that can be retrieved from connected SFP devices.
atPluggableDiagTempTable	{ atPluggableDiagTempTable 1 }	Information about temperature parameters such as: TempStatusReading, TempCurrentAlarm, TempAlarmMax etc.
atPluggableDiagTempIfIndex	{ atPluggableDiagTempEntry 1 }	The index of the interface for the pluggable.
atPluggableDiagTempChannel	{ atPluggableDiagTempEntry 2 }	The transceiver channel of the pluggables. QSFPs have 4 channels, other modules may have only 1 channel.
atrPluggableDiagTempStatusReading	{ atPluggableDiagTempEntry 3 }	The current temperature of the device e.g. 25.4 degrees Celsius.
atPluggableDiagTempCurrentAlarm	{ atPluggableDiagTempEntry 4 }	The current temperature alarm reading has 3 values: '**High'** (the current temperature status has exceeded AlarmMax), '**Low'** (the current temperature is lower than AlarmMin), '**---'** (the current temperature is within the acceptable range).
atPluggableDiagTempAlarmMax	{ atPluggableDiagTempEntry 5 }	The maximum temperature alarm threshold value.
atPluggableDiagTempAlarmMin	{ atPluggableDiagTempEntry 6 }	The minimum temperature alarm threshold value.
atPluggableDiagTempCurrentWarning	{ atPluggableDiagTempEntry 7 }	The current temperature warnings have 3 values: '**High'** (the current temperature warning status has exceeded CurrentWarningMax), '**Low'** (the current temperature status is lower than CurrentWarningMin), '**---'** (the current temperature is within the acceptable range).

Table 27: Objects defined in AT-PLUGGABLE-DIAGNOSTICS-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atPluggableDiagTempWarningMax	{ atPluggableDiagTempEntry 8 }	The maximum temperature warning threshold value.
atPluggableDiagTempWarningMin	{atPluggableDiagTempEntry 9 }	The minimum temperature warning threshold value.
atPluggableDiagVccTable	{ atPluggableDiagTable 2 }	A table of supply voltage measurements taken from the installed pluggable devices.
atPluggableDiagVccEntry	{ atPluggableDiagVccTable 1 }	A table of supply voltage parameters for installed pluggables, such as: VccStatusReading, VccCurrentWarning etc.
atPluggableDiagVccIfIndex	{ atPluggableDiagVccEntry 1 }	The index of the interface for the pluggable.
atPluggableDiagVccChannel	{ atPluggableDiagVccEntry 2 }	The transceiver channel of the pluggables. QSFPs have 4 channels, other modules may have only 1 channel.
atPluggableDiagVccStatusReading	{ atPluggableDiagVccEntry 3 }	The supply voltage status reading.
atPluggableDiagVccCurrentAlarm	{ ATPLUGGABLEDIAGVCCENTRY 4 }	The current supply voltage alarm reading.
atPluggableDiagVccAlarmMax	{ atPluggableDiagVccEntry 5 }	The maximum supply voltage alarm threshold value.
atPluggableDiagVccAlarmMin	{ atPluggableDiagVccEntry 6 }	The minimum supply voltage alarm threshold value.
atPluggableDiagVccCurrentWarning	{ atPluggableDiagVccEntry 7 }	The current supply voltage warnings.
atPluggableDiagVccWarningMax	{ atPluggableDiagVccEntry 8 }	The maximum supply voltage warning threshold value.
atPluggableDiagVccWarningMin	{ atPluggableDiagVccEntry 9 }	The minimum supply voltage warning threshold value.
atPluggableDiagTxBiasTable	{ atPluggableDiagTable 3 }	A table of information containing TxBias (mA) parameter that can be retrieved from connected SFP devices.
atPluggableDiagTxBiasEntry	{ atPluggableDiagTxBiasTable 1 }	The information about TxBias (mA) parameters of pluggables such as: Channel, StatusReading CurrentAlarm, AlarmMax, etc.
atPluggableDiagTxBiasIfIndex	{ atPluggableDiagTxBiasEntry 1 }	The index of the interface for the pluggable.
atPluggableDiagTxBiasChannel	{ atPluggableDiagTxBiasEntry 2 }	The transceiver channel of the pluggables. QSFPs have 4 channels, other modules may have only 1 channel.
atPluggableDiagTxBiasStatusReading	{ atPluggableDiagTxBiasEntry 3 }	The present TxBias status reading (mA).
atPluggableDiagTxBiasCurrentAlarm	{ atPluggableDiagTxBiasEntry 4 }	The current TxBias (mA) alarm reading.

Table 27: Objects defined in AT-PLUGGABLE-DIAGNOSTICS-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atPluggableDiagTxBiasAlarmMax	{ atPluggableDiagTxBiasEntry 5 }	The maximum TxBias (mA) alarm threshold value.
atPluggableDiagTxBiasAlarmMin	{ atPluggableDiagTxBiasEntry 6 }	The minimum TxBias (mA) alarm threshold value.
atPluggableDiagTxBiasCurrentWarning	{ atPluggableDiagTxBiasEntry 7 }	The current TxBias (mA) warnings.
atPluggableDiagTxBiasWarningMax	{ atPluggableDiagTxBiasEntry 8 }	The maximum TxBias (mA) warning threshold value.
atPluggableDiagTxBiasWarningMin	{ atPluggableDiagTxBiasEntry 9 }	The minimum TxBias (mA) warning threshold value.
atPluggableDiagTxPowerTable	{ atPluggableDiagTable 4 }	A table of information containing TxPower (mW) parameters fthat can be retrieved from connected SFP devices.
atPluggableDiagTxPowerEntry	{ atPluggableDiagTxPowerTable 1 }	The information about TxPower (mW) parameters of pluggables such as: current status reading, current alarm PowerChannel, StatusReading, CurrentAlarm, AlarmMax, etc.
atPluggableDiagTxPowerIfIndex	{ atPluggableDiagTxPowerEntry 1 }	The index of the interface for the pluggable.
atPluggableDiagTxPowerChannel	{ atPluggableDiagTxPowerEntry 2 }	The transceiver channel of the pluggables. QSFPs have 4 channels, other modules may have only 1 channel.
atPluggableDiagTxPowerStatusReading	{ atPluggableDiagTxPowerEntry 3 }	The TxPower (mW) status reading.
atPluggableDiagTxPowerCurrentAlarm	{ atPluggableDiagTxPowerEntry 4 }	The current TxPower (mW) alarm reading.
atPluggableDiagTxPowerAlarmMax	{ atPluggableDiagTxPowerEntry 5 }	The maximum TxPower (mW) alarm threshold value.
atPluggableDiagTxPowerAlarmMin	{ atPluggableDiagTxPowerEntry 6 }	The minimum TxPower (mW) alarm threshold value.
atPluggableDiagTxPowerCurrentWarning	{ atPluggableDiagTxPowerEntry 7 }	The current TxPower (mW) warnings.
atPluggableDiagTxPowerWarningMax	{ atPluggableDiagTxPowerEntry 8 }	The maximum TxPower (mW) warning threshold value.
atPluggableDiagTxPowerWarningMin	{ atPluggableDiagTxPowerEntry 9 }	The minimum TxPower (mW) warning threshold value.
atPluggableDiagRxPowerTable	{ atPluggableDiagTable 5 }	A table of information containing RxPower (mW) parameters that can be retrieved from connected SFP devices.
atPluggableDiagRxPowerEntry	{ atPluggableDiagRxPowerTable 1 }	The information about RxPower (mW) parameters of pluggables such as StatusReading, CurrentAlarm, AlarmMax, AlarmMin, etc.
atPluggableDiagRxPowerIfIndex	{ atPluggableDiagRxPowerEntry 1 }	The index of the interface for the pluggable.

Table 27: Objects defined in AT-PLUGGABLE-DIAGNOSTICS-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atPluggableDiagRxPowerChannel	{ atPluggableDiagRxPowerEntry 2 }	The transceiver channel of the pluggables. QSFPs have 4 channels, other modules may have only 1 channel.
atPluggableDiagRxPowerStatusReading	{ atPluggableDiagRxPowerEntry 3 }	The RxPower (mW) status reading.
atPluggableDiagRxPowerCurrentAlarm	{ atPluggableDiagRxPowerEntry 4 }	The RxPower (mW) current alarm reading.
atPluggableDiagRxPowerAlarmMax	{ atPluggableDiagRxPowerEntry 5 }	The maximum RxPower (mW) alarm threshold value.
atPluggableDiagRxPowerAlarmMin	{ atPluggableDiagRxPowerEntry 6 }	The minimum RxPower (mW) alarm threshold value.
atPluggableDiagRxPowerCurrentWarning	{ atPluggableDiagRxPowerEntry 7 }	The RxPower (mW) current warnings.
atPluggableDiagRxPowerWarningMax	{ atPluggableDiagRxPowerEntry 8 }	The maximum RxPower (mW) power warning threshold value.
atPluggableDiagRxPowerWarningMin	{ atPluggableDiagRxPowerEntry 9 }	The minimum RxPower (mW) warning threshold value.
atPluggableDiagRxLosTable	{ atPluggableDiagTable 6 }	A table of information containing the RxLos of signal (Los) parameters that can be retrieved from connected modules.
atPluggableDiagLosEntry	{ atPluggableDiagLosTable 1 }	The information about RxLos of Signal (Los) parameters of pluggables such as: Channel, StatusReading, etc.
atPluggableDiagRxLosIfIndex	{ atPluggableDiagLosEntry 1 }	The index of the interface for the pluggable.
atPluggableDiagRxLosChannel	{ atPluggableDiagLosEntry 2	The transceiver channel of the pluggables. QSFPs have 4 channels, other modules may have only 1 channel.
atPluggableDiagRxLosStatusReading	{ atPluggableDiagLosEntry 3 }	The RxLos status reading indicates whether: ■ light is being received (Rx Up) and therefore the link is up, or ■ light is not being received (Rx Down) and therefore the link is down.

AT-PRODUCTS-MIB

AT-PRODUCT-MIB defines object identifiers for Allied Telesis products. Objects in this MIB, see Table 28, have the object identifier **products** { alliedTelesis 1 }, OID 1.3.6.1.4.1.207.1.

Table 28: Object identifiers for Allied Telesis products supported by the AlliedWare Plus™ Operating System

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
products	{ alliedTelesis 1 } 1.3.6.1.4.1.207.1	This file defines the identities of Allied Telesis products.
swhub	{ products 4 }	Subtree beneath which switching hubs are defined.
at_x200_GE52T	{ swhub 181 }	x200-GE52T layer two switch.
at_x200_GE28T	{ swhub 182}	x200-GE28T layer two switch.
at_x210_9GT	{ swhub 196}	x210-9GT, 8xGigbit, 1xSFP/T.
at_x210_16GT	{ swhub 197}	x210-16GT, 14xGigbit, 2xcombo SFP/T.
at_x210_24GT	{ swhub 198}	x210-24GT, 20xGigbit, 4xcombo SFP/T.
at_x310_26FT	{ swhub 216}	AT-x310-26FT layer two switch with 24 10/100 Base-T ports, two 10/100/1000 Base-T ports, and four 10 Gb/s SFP+ ports.
at_x310_50FT	{ swhub 217}	AT-x310-50FT layer two switch with 48 10/100 Base-T ports, two 10/100/1000 Base-T ports, and four 10 Gb/s SFP+ ports.
at_x310_26FP	{ swhub 218}	AT-x310-26FP layer two switch with 24 10/100 Base-T ports, two 10/100/1000 Base-T ports, four 10 Gb/s SFP+ ports, plus PSE function available on pins 1/2 and 3/6 (Mode A) of every copper port.
at_x310_50FP	{ swhub 219}	AT-x310-50FP layer two switch with 48 10/100 Base-T ports, two 10/100/1000 Base-T ports, four 10 Gb/s SFP+ ports, plus PSE function available on pins 1/2 and 3/6 (Mode A) of every copper port.
at_x230_10GT	{ swhub 224 }	AT-x230-10GP L3 switch with 8 x 10/100/1000T PoE ports and 2 x 100/1000X SFP ports.
at_x230_18GT	{ swhub 225 }	AT-x230-18GT L3 switch with 16 x 10/100/1000T ports and 2 x 100/1000X SFP ports.
at_x230_28GT	{ swhub 226 }	AT-x230-28GT L3 switch with 24 x 10/100/1000T ports and 4 x 100/1000X SFP ports.
at_x230_10GP	{ swhub 228 }	AT-x230-10GP, 8 10/100/1000BASE-RJ-45 ports, 2 10/100/1000Base SFP ports.
at_x230_18GP	{ swhub 229 }	AT-x 230-18GP, 16 10/100/1000BASE RJ-45 ports and 2 10/100/1000BASE SFP ports.
at_x230_28GP	{ swhub 230 }	AT-x 230-28GP, 24 10/100/1000BASE RJ-45 ports and 4 10/100/1000BASE SFP ports.
at_GS924MX	{ swhub 253 }	AT-GS924MX, 24 x 10/100/1000T ports, 2 combo ports (10/100/1000T or 100/1000X SFP) and 2 X 10G SFP+ Stacking/User ports.
at_GS924MPX	{ swhub 254 }	AT-GS924MPX, 24 x 10/100/1000T POE+ ports, 2 combo ports (10/100/1000T or 100/1000X SFP) and 2 X 10G SFP+ Stacking/User ports.
at_GS948MX	{ swhub 255 }	AT-GS948MX, 48 x 10/100/1000T ports, 2 combo ports (10/100/1000T or 100/1000X SFP) and 2 X 10G SFP+ Stacking/User ports.

Table 28: Object identifiers for Allied Telesis products supported by the AlliedWare Plus™ Operating System **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
at_GS948MPX	{ swhub 256 }	AT-GS948MPX, 48 x 10/100/1000T POE+ ports, 2 combo ports (10/100/1000T or 100/1000X SFP) and 2 X 10G SFP+ Stacking/User ports.
at_XS916MXT	{ swhub 257 }	AT-S916MXT, 12 x 100/1000/10G BASE-T ports and 4 x SFP/SFP+ slots.
at_XS916MXS	{ swhub 258 }	AT-XS916MXS, 12 x SFP/SFP+ slots and 4 x 100/1000/10G BASE-T ports.
at_SH230_10GP	{ swhub 260 }	AT-x230-10GP, 8 x 10/100/1000BASE RJ-45 ports, 2 x 10/100/1000BASE SFP ports and PoE+.
at_SH230_18GP	{ swhub 261 }	AT-x 230 -18GP, 16 x 10/100/1000BASE RJ-45 ports, 2 x 10/100/1000BASE SFP ports and PoE+.
at_SH230_28GP	{ swhub 262 }	AT-x 230 -28GP, 24 x 10/100/1000BASE RJ-45 ports, 2 x 10/100/1000BASE SFP ports and PoE+.
at_SH210_9GT	{ swhub 263 }	AT-x210-9GT, 8 x 10/100/1000BASE RJ-45 ports, 1 x 10/100/1000BASE SFP port.
at_SH210_16GT	{ swhub 264 }	AT-x210-16GT, 14 x 10/100/1000BASE RJ-45 ports, 2 x 10/100/1000BASE combo ports.
at_SH210_24GT	{ swhub 265 }	AT-x210-24GT, 20 x 10/100/1000BASE RJ-45 ports, 4 x 10/100/1000BASE combo ports.
at_SH310_26FT	{ swhub 266 }	AT-x310-26FT, 24 10/100BASE RJ-45 ports, 2 x 10/100/1000BASE (combo) ports and 2 x stacking ports.
at_SH310_50FT	{ swhub 267 }	AT-x310-50FT, 48 10/100BASE RJ-45 ports, 2 x 10/100/1000BASE (combo) ports and 2 x stacking ports.
at_SH310_26FP	{ swhub 268 }	AT-x310-26FP, 24 10/100BASE RJ-45 ports, 2 x 10/100/1000BASE (combo) ports and 2 x stacking ports.
at_SH310_50FP	{ swhub 269 }	AT-x310-50FP, 48 10/100BASE RJ-45 ports, 2 x 10/100/1000BASE (combo) ports and 2 x stacking ports.
at_SH230_10GT	{ swhub 270 }	AT-x230-10GT, L3 switch with 8 x 10/100/1000T ports and 2 x 100/1000X SFP ports.
at_SH230_18GT	{ swhub 271 }	AT-x230-18GT, L3 switch with 16 x 10/100/1000T ports and 2 x 100/1000X SFP ports.
at_SH230_28GT	{ swhub 272 }	AT-x230-28GT, L3 switch with 24 x 10/100/1000T ports and 4 x 100/1000X SFP ports.
atFS980M9	{ swhub 274 }	AT-FS980M/9, 8 x 10/100TX ports and 1 combo (10/100/1000T or 100/1000X SFP) uplink port.
atFS980M9PS	{ swhub 275 }	AT-FS980M/9PS, 8 x 10/100TX PoE+ ports and 1 combo (10/100/1000T or 100/1000X SFP) uplink port.
atFS980M18	{ swhub 276 }	AT-FS980M/18, 16 x 10/100TX ports and 2 combo (10/100/1000T or 100/1000X SFP) uplink ports.
atFS980M18PS	{ swhub 277 }	AT-FS980M/18PS, 16 x 10/100TX PoE+ ports and 2 combo (10/100/1000T or 100/1000X SFP) uplink ports
atFS980M28	{ swhub 278 }	AT-FS980M/28, 24 x 10/100TX ports and 4 x 100/1000X SFP uplink/stacking ports.
atFS980M28PS	{ swhub 279 }	AT-FS980M/28PS, 24 x 10/100TX PoE+ ports and 4 x 100/1000X SFP uplink/stacking ports.

Table 28: Object identifiers for Allied Telesis products supported by the AlliedWare Plus™ Operating System **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atFS980M52	{ swhub 280 }	AT-FS980M/52, 48 x 10/100TX ports and 4 x 100/1000X SFP uplink/stacking ports.
atFS980M52PS	{ swhub 281 }	AT-FS980M/52PS, 48 x 10/100TX PoE+ ports and 4 x 100/1000X SFP uplink/stacking ports.
atGS970M28PS	{ swhub 312 }	AT-GS970M/28PS, Layer 3 24 x 10/100/1000T Gigabit PoE+ ports and 4 x 100/1000X SFP ports.
atGS970M18PS	{ swhub 313 }	AT-GS970M/18PS, Layer 3 16 x 10/100/1000T Gigabit PoE+ ports and 2 x 100/1000X SFP ports.
atGS970M10PS	{ swhub 314 }	AT-GS970M/10PS, Layer 3 8 x 10/100/1000T Gigabit PoE+ ports and 2 x 100/1000X SFP ports.
atGS970M28	{ swhub 315 }	AT-GS970M/28, Layer 3 24 x 10/100/1000T Gigabit Ethernet ports and 4 x 100/1000X SFP ports.
atGS970M18	{ swhub 316 }	AT-GS970M/18, Layer 3 16 x 10/100/1000T Gigabit Ethernet ports and 2 x 100/1000X SFP ports.
atGS970M10	{ swhub 317 }	AT-GS970M/10, Layer 3 8 x 10/100/1000T Gigabit Ethernet ports and 2 x 100/1000X SFP ports.
routerSwitch	{ products 14 }	Subtree beneath which router and (non industrial) switch product MIB object IDs are assigned.
at_SwitchBladex908	{ routerSwitch 69 }	SwitchBlade x908 8 Slot Layer 3 Switch Chassis.
at_x900_12XTS	{ routerSwitch 70 }	AT-x900-12XT/S Advanced Gigabit Layer 3+ Expandable Switch, 12 x combo ports (10/100/1000BASE-T copper or SFP), 1 x 30Gbps expansion bay.
at_x900_24XT	{ routerSwitch 75 }	x900-24XT Enhanced Gigabit Layer 3+ Expandable Switch, 24 x 10/100/1000BASE-T copper ports (RJ-45 connectors), 2 x 20 Gigabit expansion bays.
at_x900_24XS	{ routerSwitch 76 }	x900-24XS Enhanced Gigabit Layer 3+ Expandable Switch, 24 x 10/100/1000BASE-T copper ports (RJ-45 connectors), 2 x 20 Gigabit expansion bays.
at_x900_24XT_N	{ routerSwitch 77 }	x900-24XT-N Enhanced Gigabit Layer 3+ Expandable Switch, 24 x 10/100/1000BASE-T copper ports (RJ-45 connectors), 2 x 20 Gigabit expansion bays, NEBS compliant.
at_x600_24Ts	{ routerSwitch 80 }	x600-24Ts Stackable Managed L2+/L3 Ethernet Switch, 24 x 10/100/1000BASE-T copper ports, 4 x SFP (combo) ports.
at_x600_24TsXP	{ routerSwitch 81 }	x600-24Ts/XP Stackable Managed L2+/L3 Ethernet Switch, 24 x 10/100/1000BASE-T copper ports, 4 x SFP (combo) ports, 2 x XFP ports.
at_x600_48Ts	{ routerSwitch 82 }	x600-48Ts Stackable Managed L2+/L3 Ethernet Switch, 48 x 10/100/1000BASE-T copper ports, 4 x SFP ports.
at_x600_48TsXP	{ routerSwitch 83 }	x600-48Ts/XP Stackable Managed L2+/L3 Ethernet Switch, 48 x 10/100/1000BASE-T copper ports, 4 x SFP ports, 2 x XFP ports.
at-SBx8112	{ routerSwitch 86 }	AT-SBx8112, SwitchBlade x8112 chassis.
at-SBx81CFC400	{ routerSwitch 87 }	AT-SBx81CFC, Control Fabric Card for SwitchBlade x8112.

Table 28: Object identifiers for Allied Telesis products supported by the AlliedWare Plus™ Operating System **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
at-SBx81CFC960	{ routerSwitch 88 }	AT-SBx81CFC960 Control Fabric Card for SwitchBlade x8100 Series chassis, four 10GbE SFP+ ports.
at_x600-24TsPoE	{ routerSwitch 91}	x600-24Ts-POE Stackable Managed L2+/L3 Ethernet PoE Switch, 24 x 1000BASE-T PoE ports, 4 x SFP (combo) ports.
at_x600_24TPoEPlus	{routerSwitch 92}	x600-24Ts-POE+ Stackable Managed L2+/L3 Ethernet PoE+ Switch, 24 x 1000BASE-T PoE+ ports, 4 x SFP (combo) ports.
x610_48Ts_X_POEPlus	{routerSwitch 93}	x610-48Ts/X-POE+ Stackable Managed L2+/L3 Ethernet PoE+ Switch, 48 x 10/100/1000BASE-T PoE+ ports, 2 x SFP (combo) ports, 2 x SFP+ ports.
x610_48Ts_POEPlus	{routerSwitch 94}	x610-48Ts-POE+ Stackable Managed L2+/L3 Ethernet PoE+ Switch, 48 x 10/100/1000BASE-T PoE+ ports, 4 x SFP (combo) ports.
x610_24Ts_X_POEPlus	{routerSwitch 95}	x610-24Ts/X-POE+ Stackable Managed L2+/L3 Ethernet PoE+ Switch, 24 x 10/100/1000BASE-T PoE+ ports, 4 x SFP (combo) ports, 2 x SFP+ ports.
x610_24Ts_POEPlus	{routerSwitch 96}	x610-24Ts-POE+ Stackable Managed L2+/L3 Ethernet PoE+ Switch, 24 x 10/100/1000BASE-T PoE+ ports, 4 x SFP (combo) ports.
x610_48Ts_X	{routerSwitch 97}	x610-48Ts/X Stackable Managed L2+/L3 Ethernet Switch, 48 x 10/100/1000BASE-T copper ports, 2 x SFP (combo) ports, 2 x SFP+ ports.
x610_48Ts	{routerSwitch 98}	x610-48Ts Stackable Managed L2+/L3 Ethernet Switch, 24 x 10/100/1000BASE-T copper ports, 4 x SFP (combo) ports.
x610_24Ts_X	{routerSwitch 99}	x610-24Ts/X Stackable Managed L2+/L3 Ethernet Switch, 24 x 10/100/1000BASE-T copper ports, 4 x SFP (combo) ports, 2 x SFP+ ports.
x610_24Ts	{routerSwitch 100}	x610-24Ts Stackable Managed L2+/L3 Ethernet Switch, 24 x 10/100/1000BASE-T copper ports, 4 x SFP (combo) ports.
x610_24SP_X	{routerSwitch 101}	x610-24SP/X Stackable Managed L2+/L3 Ethernet Switch, 24 x SFP (combo) ports, 2 x SFP+ ports.
x510_28GTX	{routerSwitch 109}	x510-28GTX Stackable Managed L2+/L3 Ethernet Switch with 24 x 10/100/1000 Base-T ports and 4 x 10 Gb/s SFP+ ports.
x510_28GPX	{routerSwitch 110}	x510-28GPX Stackable Managed L2+/L3 Ethernet Switch with 24 x 10/100/1000 Base-T ports with PoE, 4 x 10 Gb/s SFP+ ports.
x510_28GSX	{routerSwitch 111}	x510-28GSX Stackable Managed L2+/L3 Ethernet Switch with 24 x 100/1000 SFP ports and 4 x 10 Gb/s SFP+ ports.
x510_52GTX	{routerSwitch 112}	x510-52GTX Stackable Managed L2+/L3 Ethernet Switch with 48 x 10/100/1000 Base-T ports and 4 x 10 Gb/s SFP+ ports.
x510_52GPX	{routerSwitch 113}	x510-52GPX Stackable Managed L2+/L3 Ethernet Switch with 48 x 10/100/1000 Base-T ports with PoE, and 4 x 10 Gb/s SFP+ ports.
at-SBx8106	{routerSwitch 114}	AT-SBx8106, SwitchBlade x8106 chassis.
x510DP_52GTX	{routerSwitch 116}	x510DP-52GTX, Stackable Managed L2+/L3 Ethernet Switch with 48 x 10/100/1000 Base-T ports and 4 x 10 Gb/s SFP+ ports.
IX5_28GPX	{routerSwitch 117}	IX5-28GPX, Stackable Managed L2+ Ethernet Switch with 24 x 10/100/1000 Base-T ports with PoE, 4 x 10 Gb/s SFP+ ports.
at_x930_28GTX	{ routerSwitch 118 }	AT-x930-28GTX, 24 10/100/1000BASE-T ports, 4 x 10 Gb/s SFP+ ports.

Table 28: Object identifiers for Allied Telesis products supported by the AlliedWare Plus™ Operating System **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
at_x930_28GPX	{ routerSwitch 119 }	AT-x930-28GPX, 24 10/100/1000BASE-T ports, 4 x 10 Gb/s SFP+ ports.
at_x930_28GSTX	{ routerSwitch 120 }	AT-x930-28GSTX, 24 10/100/1000BASE-T ports (combo), 24 x 1000 SFP ports (combo), 4 x 10 Gb/s SFP+ ports.
at_x930_52GTX	{ routerSwitch 121 }	AT-x930-52GTX, 48 10/100/1000BASE-T ports, 4 x 10 Gb/s SFP+ ports.
at_x930_52GPX	{ routerSwitch 122 }	AT-x930-52GPX, 48 10/100/1000BASE-T ports, 4 x 10 Gb/s SFP+ ports.
at_dc2552xs	{ routerSwitch 123 }	AT-DC2552XS/L3, Stackable 10 Gigabit Layer 3 switch with 48 x SFP+ ports, 4 x QSFP ports.
at_x510_28GSX_DC	{ routerSwitch 124 }	AT-x510-28GSX, 24 x 100/1000X fiber access ports and 4 x 1G/10G SFP+ uplink ports.
at_x510DP_28GTX	{ routerSwitch 126 }	AT-x510DP-28GTX, 24 10/100/1000BASE RJ-45 ports, 4 x SFP+ ports.
at_x510L_28GT	{ routerSwitch 127 }	AT-x510L-28GT, 24 x 10/100/1000BASE RJ-45 ports, 4 x SFP+ ports.
at_x510L_52GT	{ routerSwitch 128 }	T-x510L-52GT Gigabit edge switch features 48 x 10/100/1000T ports and 4 x 1G/10G SFP+ uplink ports.
at_x510L_28GP	{ routerSwitch 129 }	AT-x510L-28GP, 24 x 10/100/1000BASE RJ-45 ports, 4 x SFP+ ports.
at_x510L_52GP	{ routerSwitch 130 }	AT-x510L-52GT, 48 x 10/100/1000BASE RJ-45 ports, 4 x SFP+ ports.
atx51028GTXR	{ routerSwitch 131 }	AT-x510-28GTX, 24 x 10/100/1000T ports and 4 x 10G/1G SFP+ uplink ports.
atx51052GTXR	{ routerSwitch 132 }	AT-x510-52GTX, 48 x 10/100/1000T ports and 4 x 10G/1G SFP+ uplink port.
atSH51028GTX	{ routerSwitch 133 }	AT-x510-28GTX, 24 x 10/100/1000T ports and 4 x 10G/1G SFP+ uplink ports.
atSH51052GTX	{ routerSwitch 134 }	AT-x510-52GTX, 48 x 10/100/1000T ports and 4 x 10G/1G SFP+ uplink ports.
atSH51028GPX	{ routerSwitch 135 }	AT-x510-28GPX, 24 x 10/100/1000T PoE+ ports and 4 x 10G/1G SFP+ uplink ports.
atSH51052GPX	{ routerSwitch 136	AT-x510-52GPX, 48 x 10/100/1000T PoE+ ports and 4 x 10G/1G SFP+ uplink ports.
atsbx908g2	{ routerSwitch 137 }	AT-SBx908GEN2, High capacity Layer 3+ modular switch chassis with 8 x high speed expansion bays.
atx55018XTQ	{ routerSwitch 139 }	AT-x550-18XTQ-xx, 16-port 1G/10G BaseT stackable switch with 2 QSFP ports.
atx55018XSQ	{ routerSwitch 140 }	AT-x550-18XSQ-xx, 16-port 1G/10G SFP+ stackable switch with 2 QSFP ports.
atx55018XSPQm	{ routerSwitch 141 }	AT-x550-18XSPQm-xx, 8-port 1G/2.5G/5G/10G BaseT PoE+ and 8-port 1G/10G SFP+ stackable switch with 2 QSFP ports.
atSBx81XLEM	{ routerSwitch 142 }	AT-SBx81XLEM, Modular 40G line card with 12 x 100/1000X SFP
atx53028GTXm	{ routerSwitch 143 }	AT-x530-28GTXm, layer 3 switch is high-performing and feature-rich, with 24 x 100M/1G copper ports and 4 x 10G uplinks.
atx53028GPXm	{ routerSwitch 144 }	AT-x530-28GPXm, layer 3 switch is high-performing and feature-rich, with 24 x 100M/1G copper PoE+ ports and 4 x 10G uplinks.

Table 28: Object identifiers for Allied Telesis products supported by the AlliedWare Plus™ Operating System **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atx95028XTQm	{ routerSwitch 150 }	AT-x950-28XTQm, 24-port 1/2.5/5/10G copper stackable switch with 4 x 40G/100G QSFP+/QSFP28 ports.
atx95028XSQ	{ routerSwitch 151 }	AT-x950-28XSQ, 24-port 1/10G SFP/SFP+ stackable switch with 4 x 40G/100G QSFP+/QSFP28 ports.
atx530L28GTX	{ routerSwitch 158 }	AT-x530-28GTXm, 24-port 100/1000T stackable switch with 4 SFP+ ports.
atx530L28GPX	{ routerSwitch 159 }	AT-x530-28GPXm, 24-port 100/1000T PoE+ stackable switch with 4 SFP+ ports.
industrialSwitch	{ products 24 }	Subtree beneath which industrial switch product MIB object IDs are assigned.
at_IE200_6GT	{industrialSwitch 1}	IE200-6GT, L2+ managed industrial Switch with 4 x 10/100/1000T LAN ports and 2 x SFP uplinks (100/1000X).
at_IE200_6GP	{industrialSwitch 2}	IE200-6GP, L2+ managed industrial Switch with 4 x 10/100/1000T LAN ports (with 802.3at PoE+) and 2 x SFP uplinks (100/1000X).
at_IE200_6GPW	{industrialSwitch 3}	IE200-6GPW, L2+ managed industrial Switch with 4 x 10/100/1000T LAN ports (with 802.3at PoE+) and 2 x SFP uplinks (100/1000X) and 802.11bgn wireless.
at_IE200_6FT	{ industrialSwitch 6 }	AT-IE200-6FT, 4 × 10/100TX ports and 2 × 100/1000X SFP ports.
at_IE200_6FP	{ industrialSwitch 7 }	AT-IE200-6FP, 4 × 10/100TX ports (PoE+ support) and 2 × 100/1000X SFP ports.
at_IE300_12GT	{ industrialSwitch 8 }	AT-IE300-12GT-80, 8 x 10/100/1000T, 4 x 100/1000X SFP, Industrial Ethernet, Layer 3 Switch.
at_IE300_12GP	{ industrialSwitch 9 }	AT-IE300-12GP-80, 8 x 10/100/1000T, 4 x 100/1000X SFP, Industrial Ethernet, Layer 3 Switch.

AT-PTP-MIB

See Table 29 for the list of objects defined in the AT-PTP-MIB. These objects contain definitions of managed objects for Precision Time Protocol (PTP). Objects in this group have the object identifier **atPtpMIB** { modules 504 } OID 1.3.6.1.4.1.207.8.4.4.4.504.

Table 29: Objects defined in AT-PTP-MIB

<table>
<tr><th>OBJECT</th><th>OBJECT IDENTIFIER</th><th>DESCRIPTION</th></tr>
<tr><td>atPtpMIB</td><td>{ modules 504 }
1.3.6.1.4.1.207.8.4.4.4.504</td><td>This MIB file contains definitions of managed objects for the IEEE 1588v2 Precision Time Protocol (PTP) module.</td></tr>
<tr><td colspan="3">TEXTUAL CONVENTIONS</td></tr>
<tr><td>PtpClockDomainType</td><td colspan="2">The Domain is identified by an integer, the domainNumber, in the range of 0 to 255. An integer value that is used to assign each PTP device to a particular domain. The following values define the valid domains:
Table 30:
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>0</td><td>Default domain</td></tr>
<tr><td>1</td><td>Alternate domain 1</td></tr>
<tr><td>2</td><td>Alternate domain 2</td></tr>
<tr><td>3</td><td>Alternate domain 3</td></tr>
<tr><td>4-127</td><td>User-defined domains</td></tr>
<tr><td>28-255</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>PtpClockIdentity</td><td colspan="2">The clock Identity is an 8-octet array and will be presented in the form of a character array. Network byte order is assumed. The value of the PtpClockIdentity should be taken from the IEEE EUI-64 individual assigned numbers as indicated in Section 7.5.2.2.2 of [IEEE 1588-2008]. It can also be non-EUI-64 address as defined in section 7.5.2.2.3 of [IEEE 1588-2008]. The EUI-64 address is divided into the following fields:
■ OUI bytes (0-2)
■ Extension identifier bytes (3-7)
The clock identifier can be constructed from existing EUI-48 assignments and here is an abbreviated example extracted from section 7.5.2.2.2 [IEEE 1588-2008].
■ Company EUI-48 = 0xACDE4823456716
■ EUI-64 = ACDE48FFFE23456716
It is important to note the IEEE Registration Authority has deprecated the use of MAC-48 in any new design.</td></tr>
</table>

Table 29: Objects defined in AT-PTP-MIB **(continued)**

<table>
<tr><th>OBJECT</th><th>OBJECT IDENTIFIER</th><th>DESCRIPTION</th></tr>
<tr><td>PtpClockInstanceType</td><td colspan="2">The instance of the Clock of a given clock type in a given domain.</td></tr>
<tr><td>PtpClockIntervalBase2</td><td colspan="2">The interval included in message types Announce, Sync, Delay_Req, and Pdelay_Req as indicated in section 7.7.2.1 of [IEEE 1588-2008].
The mean time interval between successive messages shall be represented as the logarithm to the base 2 of this time interval measured in seconds on the local clock of the device sending the message. The values of these logarithmic attributes shall be selected from integers in the range -128 to 127 subject to further limits established in an applicable PTP profile.</td></tr>
<tr><td>PtpClockMechanismType</td><td colspan="2">The clock type based on whether end-to-end or peer-to-peer mechanisms are used. The mechanism used to calculate the Mean Path Delay as indicated in Table 9 of [IEEE 1588-2008]:
Table 31:
<table>
<tr><th>Display mechanism</th><th>Value (hex)</th><th>Specification</th></tr>
<tr><td>E2E</td><td>01</td><td>The port is configured to use the delay request-response mechanism.</td></tr>
<tr><td>P2P</td><td>02</td><td>The port is configured to use the peer delay mechanism.</td></tr>
<tr><td>DISABLED</td><td>FE</td><td>The port does not implement the delay mechanism</td></tr>
</table></td></tr>
<tr><td>PtpClockPortNumber</td><td colspan="2">An index identifying a specific Precision Time Protocol (PTP) port on a PTP node.</td></tr>
</table>

Table 29: Objects defined in AT-PTP-MIB **(continued)**

<table>
<tr><th>OBJECT</th><th>OBJECT IDENTIFIER</th><th>DESCRIPTION</th></tr>
<tr><td>PtpClockPortState</td><td colspan="2">This is the value of the current state of the protocol engine associated with this port:
Table 32:
<table><tr><th>Port state</th><th>Value</th><th>Description</th></tr><tr><td>initializing</td><td>1</td><td>In this state a port initializes its data sets, hardware and communication facilities.</td></tr><tr><td>faulty</td><td>2</td><td>The fault state of the protocol.</td></tr><tr><td>disabled</td><td>3</td><td>The port shall not place any messages on its communication path.</td></tr><tr><td>listening</td><td>4</td><td>The port is waiting for the announceReceiptTimeout to expire or to receive an Announce message from a master.</td></tr><tr><td>preMaster</td><td>5</td><td>The port shall behave in all respects as though it were in the MASTER state except that it shall not place any messages on its communication path except for Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up, Signaling, or management messages.</td></tr><tr><td>master</td><td>6</td><td>The port is behaving as a master port.</td></tr><tr><td>passive</td><td>7</td><td>The port shall not place any messages on its communication path except for Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up, or signaling messages, or management messages that are a required response to another management message.</td></tr><tr><td>uncalibrated</td><td>8</td><td>The local port is preparing to synchronize to the master port.</td></tr><tr><td>slave</td><td>9</td><td>The port is synchronizing to the selected master port.</td></tr></table></td></tr>
<tr><td>PtpClockPortTransportTypeAddress</td><td colspan="2">The Clock port transport protocol address used for this communication between the clock nodes. This is a string corresponding to the address type as specified by the transport type used. The transport types can be defined elsewhere, in addition to the ones defined in this document. This can be an address of type IP version 4, IP version 6, Ethernet, DeviceNET, ControlNET or IEC61158. The OCTET STRING representation of the OID of ptpWellKnownTransportTypes will be used in the values contained in the OCTET STRING.</td></tr>
<tr><td>PtpClockProfileType</td><td colspan="2">Clock Profile used. A profile is the set of allowed Precision Time Protocol (PTP) features applicable to a device.</td></tr>
</table>

Table 29: Objects defined in AT-PTP-MIB **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
PtpClockQualityAccuracyType	The ClockQuality as specified in sections 5.3.7, 7.6.2.5 and Table 6 of [IEEE 1588-2008]. The following values are not represented in the enumerated values. ■ 0x01-0x1F Reserved ■ 0x32-0x7F Reserved It is important to note that section 7.1.1 of [RFC 2578] allows for gaps and enumerate values starting at zero when indicated by the protocol.	

Table 29: Objects defined in AT-PTP-MIB **(continued)**

<table>
<tr><th>OBJECT</th><th>OBJECT IDENTIFIER</th><th>DESCRIPTION</th></tr>
<tr><td>PtpClockQualityClassType</td><td colspan="2">The ClockQuality as specified in section 5.3.7 ClockQuality, 7.6.2.4 clockClass and Table 5 clockClass specifications of [IEEE 1588-2008]:
Table 33:
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Reserved to enable compatibility with future versions.</td></tr><tr><td>1-5</td><td>Reserved.</td></tr><tr><td>6</td><td>Shall designate a clock that is synchronized to a primary reference time source. The timescale distributed shall be PTP. A clockClass 6 clock shall not be a slave to another clock in the domain.</td></tr><tr><td>7</td><td>Shall designate a clock that has previously been designated as clockClass 6 but that has lost the ability to synchronize to a primary reference time source and is in holdover mode and within holdover specifications. The timescale distributed shall be PTP. A clockClass 7 clock shall not be a slave to another clock in the domain.</td></tr><tr><td>8</td><td>Reserved.</td></tr><tr><td>9-10</td><td>Reserved to enable compatibility with future versions.</td></tr><tr><td>11-12</td><td>Reserved.</td></tr><tr><td>13</td><td>Shall designate a clock that is synchronized to an application-specific source of time. The timescale distributed shall be ARB. A clockClass 13 clock shall not be a slave to another clock in the domain.</td></tr><tr><td>14</td><td>Shall designate a clock that has previously been designated as clockClass 13 but that has lost the ability to synchronize to an application-specific source of time and is in holdover mode and within holdover specifications. The timescale distributed shall be ARB. A clockClass 14 clock shall not be a slave to another clock in the domain.</td></tr><tr><td>15-51</td><td>Reserved.</td></tr><tr><td>52</td><td>Degradation alternative A for a clock of clockClass 7 that is not within holdover specification. A clock of clockClass 52 shall not be a slave to another clock in the domain.</td></tr></table></td></tr>
</table>

Table 29: Objects defined in AT-PTP-MIB **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
PtpClockQualityClassType	(continued)	

Table 34:

Value	Description
53-57	Reserved.
58	Degradation alternative A for a clock of clockClass 14 that is not within holdover specification. A clock of clockClass 58 shall not be a slave to another clock in the domain.
59-67	Reserved.
68-122	For use by alternate PTP profiles.
123-127	Reserved.
128-132	Reserved.
133-170	For use by alternate PTP profiles.
171-186	Reserved.
187	Degradation alternative B for a clock of clockClass 7 that is not within holdover specification. A clock of clockClass 187 may be a slave to another clock in the domain.
188-192	Reserved.
193	Degradation alternative B for a clock of clockClass 14 that is not within holdover specification. A clock of clockClass 193 may be a slave to another clock in the domain.
194-215	Reserved.
216-232	For use by alternate PTP profiles.
233-247	Reserved.
248	Default. This clockClass shall be used if none of the other clockClass definitions apply.
249-250	Reserved.
251	Reserved for version 1 compatibility; see Clause 18.
252-254	Reserved.
255	Shall be the clockClass of a slave-only clock; see 9.2.2

Table 29: Objects defined in AT-PTP-MIB **(continued)**

<table>
<tr><th>OBJECT</th><th>OBJECT IDENTIFIER</th><th>DESCRIPTION</th></tr>
<tr><td>PtpClockRoleType</td><td colspan="2">The Clock Role. The protocol generates a Master Slave relationship among the clocks in the system:
Table 35:
<table>
<tr><th>Clock Role</th><th>Value</th><th>Description</th></tr>
<tr><td>Master clock</td><td>1</td><td>A clock that is the source of time to which all other clocks on that path synchronize.</td></tr>
<tr><td>Slave clock</td><td>2</td><td>A clock which synchronizes to another clock (master).</td></tr>
</table></td></tr>
<tr><td>PtpClockStateType</td><td colspan="2">The clock state returned by a PTP engine:
Table 36:
<table>
<tr><th>Clock state</th><th>Value</th><th>Description</th></tr>
<tr><td>Freerun state</td><td>1</td><td>Applies to a slave device that is not locked to a master. This is the initial state a slave starts out with when it is not getting any PTP packets from the master or because of some other input error (erroneous packets, etc).</td></tr>
<tr><td>Holdover state</td><td>2</td><td>In this state the slave device is locked to a master but communication with the master has been lost or the timestamps in the PTP packets are incorrect. Since the slave was locked to the master, it can run in this state, with similar accuracy for some time. If communication with the master is not restored for an extended period (dependent on the clock implementation), the device should move to the Freerun state.</td></tr>
<tr><td>Acquiring state</td><td>3</td><td>The slave device is receiving packets from a master and is trying to acquire a lock.</td></tr>
<tr><td>Freq_locked state</td><td>4</td><td>Slave device is locked to the Master with respect to frequency, but not phase aligned.</td></tr>
<tr><td>Phase_aligned state</td><td>5</td><td>Locked to the master with respect to frequency and phase.</td></tr>
</table></td></tr>
</table>

Table 29: Objects defined in AT-PTP-MIB **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
PtpClockTimeInterval	This textual convention corresponds to the TimeInterval structure indicated in section 5.3.2 of [IEEE 1588-2008]. It will be presented in the form of a character array. Network byte order is assumed. The TimeInterval type represents time intervals. struct TimeInterval {Integer64 scaledNanoseconds} The scaledNanoseconds member is the time interval expressed in units of nanoseconds and multiplied by 2**16. Positive or negative time intervals outside the maximum range of this data type shall be encoded as the largest positive and negative values of the data type, respectively. For example, 2.5 ns is expressed as string '0000 0000 0002 8000' in Base16.	
PtpClockTimeSourceType	The ClockQuality as specified in Sections 5.3.7, 7.6.2.6 and Table 7 of [IEEE 1588-2008]. The following values are not represented in the enumerated values: ■ 0xF0-0xFE - For use by alternate PTP profiles ■ 0xFF - Reserved It is important to note that section 7.1.1 RFC 2578 allows for gaps and enumerate values to start with zero when indicated by the protocol.	
PtpClockTxModeType	Transmission mode: ■ Unicast: - Using unicast communication channel. ■ Multicast: - Using Multicast communication channel. ■ multicast-mix: - Using multicast-unicast communication channel.	
PtpClockType	The clock types as defined in the MIB module description.	
ptpSystemTable	{ ptpMIBSystemInfo 1 }	Table of count information about the PTP system for all domains.
ptpSystemEntry	{ ptpSystemTable 1 }	An entry in the table, containing count information about a single domain. New row entries are added when the PTP clock for this domain is configured, while the unconfiguration of the PTP clock removes it.

Table 29: Objects defined in AT-PTP-MIB **(continued)**

<table>
<tr><th>OBJECT</th><th>OBJECT IDENTIFIER</th><th>DESCRIPTION</th></tr>
<tr><td>ptpDomainIndex</td><td>{ ptpSystemEntry 1 }</td><td>This object specifies the domain number used to create a logical group of PTP devices. The Clock Domain is a logical group of clocks and devices that synchronize with each other using the PTP protocol:
Table 37:<table><tr><th>Value</th><th>Definition</th></tr><tr><td>0</td><td>Default domain</td></tr><tr><td>1</td><td>Alternate domain 1</td></tr><tr><td>2</td><td>Alternate domain 2</td></tr><tr><td>3</td><td>Alternate domain 3</td></tr><tr><td>4-127</td><td>User-defined domains</td></tr><tr><td>28-255</td><td>Reserved</td></tr></table></td></tr>
<tr><td>ptpInstanceIndex</td><td>{ ptpSystemEntry 2 }</td><td>This object specifies the instance of the Clock for this domain.</td></tr>
<tr><td>ptpDomainClockPortsTotal</td><td>{ ptpSystemEntry 3 }</td><td>This object specifies the total number of clock ports configured within a domain in the system.</td></tr>
<tr><td>ptpSystemDomainTable</td><td>{ ptpMIBSystemInfo 2 }</td><td>Table of information about the PTP system for all clock modes -- ordinary, boundary or transparent.</td></tr>
<tr><td>ptpSystemDomainEntry</td><td>{ ptpSystemDomainTable 1 }</td><td>An entry in the table, containing information about a single clock mode for the PTP system. A row entry gets added when PTP clocks are configured on the node.</td></tr>
<tr><td>ptpSystemDomainClockTypeIndex</td><td>{ ptpSystemDomainEntry 1 }</td><td>This object specifies the clock type as defined in the Textual convention description.</td></tr>
<tr><td>ptpSystemDomainTotals</td><td>{ ptpSystemDomainEntry 2 }</td><td>This object specifies the total number of PTP domains for this particular clock type configured in this node.</td></tr>
<tr><td>ptpSystemProfile</td><td>{ ptpMIBSystemInfo 3 }</td><td>This object specifies the PTP Profile implemented on the system.</td></tr>
<tr><td>ptpClockCurrentDSTable</td><td>{ ptpMIBClockInfo 1 }</td><td>Table of information about the PTP clock Current Datasets for all domains.</td></tr>
<tr><td>ptpClockCurrentDSEntry</td><td>{ ptpClockCurrentDSTable 1 }</td><td>An entry in the table, containing information about a single PTP clock Current Datasets for a domain.</td></tr>
</table>

Table 29: Objects defined in AT-PTP-MIB **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
ptpClockCurrentDSDomainIndex	{ ptpClockCurrentDSEntry 1 }	This object specifies the domain number used to create a logical group of PTP devices.
ptpClockCurrentDSClockTypeIndex	{ ptpClockCurrentDSEntry 2 }	This object specifies the clock type as defined in the Textual convention description
ptpClockCurrentDSInstanceIndex	{ ptpClockCurrentDSEntry 3 }	This object specifies the instance of the clock for this clock type in the given domain.
ptpClockCurrentDSStepsRemoved	{ ptpClockCurrentDSEntry 4	The current clock dataset StepsRemoved value. This object specifies the distance measured by the number of Boundary clocks between the local clock and the Foreign master as indicated in the stepsRemoved field of Announce messages.
ptpClockCurrentDSOffsetFromMaster	{ ptpClockCurrentDSEntry 5 }	This object specifies the current clock dataset ClockOffset value. The value of the computation of the offset in time between a slave and a master clock
ptpClockCurrentDSMeanPathDelay	{ ptpClockCurrentDSEntry 6 }	This object specifies the current clock dataset MeanPathDelay value. The mean path delay between a pair of ports as measured by the delay request-response mechanism
ptpClockParentDSTable	{ ptpMIBClockInfo 2 }	Table of information about the PTP clock Parent Datasets for all domains.
ptpClockParentDSEntry	{ ptpClockParentDSTable 1 }	An entry in the table, containing information about a single PTP clock Parent Datasets for a domain.
ptpClockParentDSDomainIndex	{ ptpClockParentDSEntry 1 }	This object specifies the domain number used to create a logical group of PTP devices.
ptpClockParentDSClockTypeIndex	{ ptpClockParentDSEntry 2 }	This object specifies the clock type as defined in the Textual convention description.
ptpClockParentDSInstanceIndex	{ ptpClockParentDSEntry 3 }	This object specifies the instance of the clock for this clock type in the given domain.
ptpClockParentDSParentPortIdentity	{ ptpClockParentDSEntry 4 }	This object specifies the value of portIdentity of the port on the master that issues the Sync messages used in synchronizing this clock.
ptpClockParentDSParentStats	{ ptpClockParentDSEntry 5 }	This object specifies the Parent Dataset ParentStats value. This value indicates whether the values of ParentDSOffset and ParentDSClockPhChRate have been measured and are valid. A TRUE value shall indicate valid data.
ptpClockParentDSOffset	{ ptpClockParentDSEntry 6 }	This object specifies the Parent Dataset ParentOffsetScaledLogVariance value. This value is the variance of the parent clock's phase as measured by the local clock.
ptpClockParentDSClockPhChRate	{ ptpClockParentDSEntry 7 }	This object specifies the clock's parent dataset ParentClockPhaseChangeRate value. This value is an estimate of the parent clock's phase change rate as measured by the slave clock.
ptpClockParentDSGMClockIdentity	{ ptpClockParentDSEntry 8 }	This object specifies the parent dataset Grandmaster clock identity.

Table 29: Objects defined in AT-PTP-MIB **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
ptpClockParentDSGMClockPriority1	{ ptpClockParentDSEntry 9 }	This object specifies the parent dataset Grandmaster clock priority1.
ptpClockParentDSGMClockPriority2	{ ptpClockParentDSEntry 10 }	This object specifies the parent dataset grandmaster clock priority2.
ptpClockParentDSGMClockQualityClass	{ ptpClockParentDSEntry 11 }	This object specifies the parent dataset grandmaster clock quality class.
ptpClockParentDSGMClockQualityAccuracy	{ ptpClockParentDSEntry 12 }	This object specifies the parent dataset grandmaster clock quality accuracy.
ptpClockParentDSGMClockQualityOffset	{ ptpClockParentDSEntry 13 }	This object specifies the parent dataset grandmaster clock quality offset.
ptpClockDefaultDSTable	{ ptpMIBClockInfo 3 }	Table of information about the PTP clock Default Datasets for all domains.
ptpClockDefaultDSEntry	{ ptpClockDefaultDSTable 1 }	An entry in the table, containing information about a single PTP clock Default Datasets for a domain.
ptpClockDefaultDSDomainIndex	{ ptpClockDefaultDSEntry 1 }	This object specifies the domain number used to create a logical group of PTP devices.
ptpClockDefaultDSClockTypeIndex	{ ptpClockDefaultDSEntry 2 }	This object specifies the clock type as defined in the Textual convention description.
ptpClockDefaultDSInstanceIndex	{ ptpClockDefaultDSEntry 3 }	This object specifies the instance of the clock for this clock type in the given domain.
ptpClockDefaultDSTwoStepFlag	{ ptpClockDefaultDSEntry 4 }	This object specifies whether the Two Step process is used.
ptpClockDefaultDSClockIdentity	{ ptpClockDefaultDSEntry 5 }	This object specifies the default Datasets clock identity.
ptpClockDefaultDSPriority1	{ ptpClockDefaultDSEntry 6 }	This object specifies the default Datasets clock Priority1.
ptpClockDefaultDSPriority2	{ ptpClockDefaultDSEntry 7 }	This object specifies the default Datasets clock Priority2.
ptpClockDefaultDSSlaveOnly	{ ptpClockDefaultDSEntry 8 }	Whether the SlaveOnly flag is set.
ptpClockDefaultDSQualityClass	{ ptpClockDefaultDSEntry 9 }	This object specifies the default dataset Quality Class.
ptpClockDefaultDSQualityAccuracy	{ ptpClockDefaultDSEntry 10 }	This object specifies the default dataset Quality Accuracy.
ptpClockDefaultDSQualityOffset	{ ptpClockDefaultDSEntry 11 }	This object specifies the default dataset Quality offset.
ptpClockRunningTable	{ ptpMIBClockInfo 4 }	Table of information about the PTP clock Default Datasets for all domains.
ptpClockRunningEntry	{ ptpClockRunningTable 1 }	An entry in the table, containing information about a single PTP clock running Datasets for a domain.
ptpClockRunningDomainIndex	{ ptpClockRunningEntry 1 }	This object specifies the domain number used to create a Logical group of PTP devices.
ptpClockRunningClockTypeIndex	{ ptpClockRunningEntry 2 }	This object specifies the clock type as defined in the Textual convention description.
ptpClockRunningInstanceIndex	{ ptpClockRunningEntry 3 }	This object specifies the instance of the clock for this clock type in the given domain.

Table 29: Objects defined in AT-PTP-MIB **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
ptpClockRunningState	{ ptpClockRunningEntry 4 }	This object specifies the Clock state returned by a PTP engine which was described earlier. **Freerun state**. Applies to a slave device that is not locked to a master. This is the initial state a slave starts out with when it is not getting any PTP packets from the master, or because of some other input error (erroneous packets, etc). Holdover state. In this state the slave device is locked to a master but communication with the master has been lost or the timestamps in the PTP packets are incorrect. Since the slave was previously locked to the master, it can run in this state, with similar accuracy for some time. **Acquiring state**. The slave device is receiving packets from a master and is trying to acquire a lock. **Freq_locked state**. Slave device is locked to the Master with respect to frequency, but not phase aligned. **Phase_aligned state**. Locked to the master with respect to frequency and phase.
ptpClockRunningPacketsSent	{ ptpClockRunningEntry 5 }	This object specifies the total number of all unicast and multicast packets that have been sent out for this clock in this domain for this type. These counters are discontinuous.
ptpClockRunningPacketsReceived	{ ptpClockRunningEntry 6 }	This object specifies the total number of all unicast and multicast packets that have been received for this clock in this domain for this type. These counters are discontinuous.
ptpClockTimePropertiesDSTable	{ ptpMIBClockInfo 5 }	Table of information about the PTP clock time properties datasets for all domains.
ptpClockTimePropertiesDSEntry	{ ptpClockTimePropertiesDSTable 1 }	An entry in the table, containing information about a single PTP clock timeproperties Datasets for a domain
ptpClockTimePropertiesDSDomainIndex	{ ptpClockTimePropertiesDSEntry 1 }	This object specifies the domain number used to create a logical group of PTP devices.
ptpClockTimePropertiesDSClockTypeIndex	{ ptpClockTimePropertiesDSEntry 2 }	This object specifies the clock type as defined in the Textual convention description.
ptpClockTimePropertiesDSInstanceIndex	{ ptpClockTimePropertiesDSEntry 3 }	This object specifies the instance of the clock for this clock type in the given domain.
ptpClockTimePropertiesDSCurrentUTCOffsetValid	{ ptpClockTimePropertiesDSEntry 4 }	This object specifies the timeproperties dataset value of whether the current UTC offset is valid.

Table 29: Objects defined in AT-PTP-MIB **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
ptpClockTimePropertiesDSCurrentUTCOffset	{ ptpClockTimePropertiesDSEntry 5 }	This object specifies the timeproperties dataset value of the current UTC offset. In PTP systems whose epoch is the PTP epoch, the value of timePropertiesDS.currentUtcOffset is the offset between TAI and UTC; otherwise the value has no meaning. The value shall be in units of seconds. The initialization value shall be selected as follows: ■ If the timePropertiesDS.ptpTimescale (see 8.2.4.8) is TRUE, the value is the value obtained from a primary reference if the value is known at the time of initialization, else, ■ The value shall be the current number of leap seconds (7.2.3) when the node is designed.
ptpClockTimePropertiesDSLeap59	{ ptpClockTimePropertiesDSEntry 6 }	This object specifies the Leap59 value in the clock Current Dataset.
ptpClockTimePropertiesDSLeap61	{ ptpClockTimePropertiesDSEntry 7 }	This object specifies the Leap61 value in the clock Current Dataset.
ptpClockTimePropertiesDSTimeTraceable	{ ptpClockTimePropertiesDSEntry 8 }	This object specifies the Time Traceable value in the clock Current Dataset.
ptpClockTimePropertiesDSFreqTraceable	{ ptpClockTimePropertiesDSEntry 9 }	This object specifies the Frequency Traceable value in the clock Current Dataset.
ptpClockTimePropertiesDSPTPTimescale	{ ptpClockTimePropertiesDSEntry 10 }	This object specifies the PTP Timescale value in the clock Current Dataset.
ptpClockTimePropertiesDSSource	{ ptpClockTimePropertiesDSEntry 11 }	This object specifies the Timesource value in the clock Current Dataset.
ptpClockTransDefaultDSTable	{ ptpMIBClockInfo 6 }	Table of information about the PTP Transparent clock Default Datasets for all domains.
ptpClockTransDefaultDSEntry	{ ptpClockTransDefaultDSTable 1 }	An entry in the table, containing information about a single PTP Transparent clock Default Datasets for a domain.
ptpClockTransDefaultDSDomainIndex	{ ptpClockTransDefaultDSEntry 1 }	This object specifies the domain number used to create a logical group of PTP devices.
ptpClockTransDefaultDSInstanceIndex	{ ptpClockTransDefaultDSEntry 2 }	This object specifies the instance of the clock for this clock type in the given domain.
ptpClockTransDefaultDSClockIdentity	{ ptpClockTransDefaultDSEntry 3 }	This object specifies the value of the clockIdentity attribute of the local clock.
ptpClockTransDefaultDSNumOfPorts	{ ptpClockTransDefaultDSEntry 4 }	This object specifies the number of PTP ports of the device. These counters are discontinuous.
ptpClockTransDefaultDSDelay	{ ptpClockTransDefaultDSEntry 5 }	This object, if the transparent clock is an end-to-end transparent clock, has the value of E2E; if the transparent clock is a peer-to-peer transparent clock, the value shall be P2P.

Table 29: Objects defined in AT-PTP-MIB **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
ptpClockTransDefaultDSPrimaryDomain	{ ptpClockTransDefaultDSEntry 6 }	This object specifies the value of the primary syntonization domain. The initialization value shall be 0.
ptpClockPortTable	{ ptpMIBClockInfo 7 }	Table of information about the clock ports for a particular domain.
ptpClockPortEntry	{ ptpClockPortTable 1 }	An entry in the table, containing information about a single clock port.
ptpClockPortDomainIndex	{ ptpClockPortEntry 1 }	This object specifies the domain number used to create a logical group of PTP devices.
ptpClockPortClockTypeIndex	{ ptpClockPortEntry 2 }	This object specifies the clock type as defined in the Textual convention description
ptpClockPortClockInstanceIndex	{ ptpClockPortEntry 3 }	This object specifies the instance of the clock for this clock type in the given domain.
ptpClockPortTablePortNumberIndex	{ ptpClockPortEntry 4 }	This object specifies the PTP Portnumber for this port.
ptpClockPortName	{ ptpClockPortEntry 5 }	This object specifies the PTP clock port name configured on the node.
ptpClockPortRole	{ ptpClockPortEntry 6 }	This object describes the current role (slave/master) of the port.
ptpClockPortSyncTwoStep	{ ptpClockPortEntry 7 }	This object specifies that two-step clock operation between the PTP master and slave device is enabled
ptpClockPortCurrentPeerAddressType	{ ptpClockPortEntry 8 }	This object specifies the current peer's network address type used for PTP communication.
ptpClockPortCurrentPeerAddress	{ ptpClockPortEntry 9 }	This object specifies the current peer's network address used for PTP communication.
ptpClockPortNumOfAssociatedPorts	{ ptpClockPortEntry 10 }	For a master port - the number of PTP slave sessions (peers) associated with this PTP port. For a slave port - the number of masters available to this slave port (might or might not be peered).
ptpClockPortDSTable	{ ptpMIBClockInfo 8 }	Table of information about the clock ports dataset for a particular domain.
ptpClockPortDSEntry	{ ptpClockPortDSTable 1 }	An entry in the table, containing port dataset information for a single clock port.
ptpClockPortDSDomainIndex	{ ptpClockPortDSEntry 1 }	This object specifies the domain number used to create a logical group of PTP devices.
ptpClockPortDSClockTypeIndex	{ ptpClockPortDSEntry 2 }	This object specifies the clock type as defined in the Textual convention description.
ptpClockPortDSClockInstanceIndex	{ ptpClockPortDSEntry 3 }	This object specifies the instance of the clock for this clock type in the given domain.
ptpClockPortDSPortNumberIndex	{ ptpClockPortDSEntry 4 }	This object specifies the PTP portnumber associated with this PTP port.
ptpClockPortDSName	{ ptpClockPortDSEntry 5 }	This object specifies the PTP clock port dataset name.

Table 29: Objects defined in AT-PTP-MIB **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
ptpClockPortDSPortIdentity	{ ptpClockPortDSEntry 6 }	This object specifies the PTP clock port Identity.
ptpClockPortDSlogAnnouncementInterval	{ ptpClockPortDSEntry 7 }	This object specifies the Announce message transmission interval associated with this clock port.
ptpClockPortDSAnnounceRctTimeout	{ ptpClockPortDSEntry 8 }	This object specifies the Announce receipt timeout associated with this clock port
ptpClockPortDSlogSyncInterval	{ ptpClockPortDSEntry 9 }	This object specifies the Sync message transmission interval.
ptpClockPortDSMinDelayReqInterval	{ ptpClockPortDSEntry 10 }	This object specifies the Delay_Req message transmission interval.
ptpClockPortDSPeerDelayReqInterval	{ ptpClockPortDSEntry 11 }	This object specifies the Pdelay_Req message transmission interval.
ptpClockPortDSDelayMech	{ ptpClockPortDSEntry 12 }	This object specifies the delay mechanism used. If the clock is an end-to-end clock, the value of the is e2e, else if the clock is a peer to-peer clock, the value shall be p2p.
ptpClockPortDSPeerMeanPathDelay	{ ptpClockPortDSEntry 13 }	This object specifies the peer meanPathDelay.
ptpClockPortDSGrantDuration	{ ptpClockPortDSEntry 14 }	This object specifies the grant duration allocated by the master.
ptpClockPortDSPTPVersion	{ ptpClockPortDSEntry 15 }	This object specifies the PTP version being used.
ptpClockPortRunningTable	{ ptpMIBClockInfo 9 }	Table of information about the clock ports running datasets for a particular domain
ptpClockPortRunningEntry	{ ptpClockPortRunningTable 1 }	An entry in the table, containing running dataset information about a single clock port.
ptpClockPortRunningDomainIndex	{ ptpClockPortRunningEntry 1 }	This object specifies the domain number used to create a logical group of PTP devices.
ptpClockPortRunningClockTypeIndex	{ ptpClockPortRunningEntry 2 }	This object specifies the clock type as defined in the Textual convention description.
ptpClockPortRunningClockInstanceIndex	{ ptpClockPortRunningEntry 3 }	This object specifies the instance of the clock for this clock type in the given domain.
ptpClockPortRunningPortNumberIndex	{ ptpClockPortRunningEntry 4 }	This object specifies the PTP portnumber associated with this clock port
ptpClockPortRunningName	{ ptpClockPortRunningEntry 5 }	This object specifies the PTP clock port name.

Table 29: Objects defined in AT-PTP-MIB **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
ptpClockPortRunningState	{ ptpClockPortRunningEntry 6 }	This object specifies the port state returned by PTP engine: Table 38:
ptpClockPortRunningRole	{ ptpClockPortRunningEntry 7 }	This object specifies the Clock Role.
ptpClockPortRunningInterfaceIndex	{ ptpClockPortRunningEntry 8 }	This object specifies the interface on the node being used by the PTP Clock for PTP communication.
ptpClockPortRunningTransport	{ ptpClockPortRunningEntry 9 }	This object specifies the transport protocol being used for PTP communication (the mapping used).
ptpClockPortRunningEncapsulationType	{ ptpClockPortRunningEntry 10 }	This object specifies the type of encapsulation if the interface is adding extra layers (e.g., VLAN, Pseudowire encapsulation...) for the PTP messages.

Initializing	In this state a port initializes its data sets, hardware, and communication facilities.
faulty	The fault state of the protocol.
disabled	The port shall not place any messages on its communication path.
listening	The port is waiting for the announceReceiptTimeout to expire or to receive an Announce message from a master.
preMaster	The port shall behave in all respects as though it were in the MASTER state except that it shall not place any messages on its communication path except for Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up, signaling, or management messages.
master	The port is behaving as a master port.
passive	The port shall not place any messages on its communication path except for Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up, or signaling messages, or management messages that are a required response to another management message.
uncalibrated	The local port is preparing to synchronize to the master port.
slave	The port is synchronizing to the selected master port.

Table 29: Objects defined in AT-PTP-MIB **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
ptpClockPortRunningTxMode	{ ptpClockPortRunningEntry 11 }	This object specifies the clock transmission mode as: ■ unicast: Using unicast communication channel. ■ multicast: Using Multicast communication channel. ■ multicast-mix: Using multicast-unicast communication channel.
ptpClockPortRunningRxMode	{ ptpClockPortRunningEntry 12 }	This object specifies the clock receive mode as: ■ unicast: Using unicast communication channel. ■ multicast: Using Multicast communication channel. ■ multicast-mix: Using multicast-unicast communication channel.
ptpClockPortRunningPacketsReceived	{ ptpClockPortRunningEntry 13 }	This object specifies the packets received on the clock port (cumulative). These counters are discontinuous.
ptpClockPortRunningPacketsSent	{ ptpClockPortRunningEntry 14 }	This object specifies the packets sent on the clock port (cumulative). These counters are discontinuous.
ptpClockPortTransDSTable	{ ptpMIBClockInfo 10 }	Table of information about the Transparent clock ports running dataset for a particular domain.
ptpClockPortTransDSEntry	{ ptpClockPortTransDSTable 1 }	An entry in the table, containing clock port Transparent dataset information about a single clock port.
ptpClockPortTransDSDomainIndex	{ ptpClockPortTransDSEntry 1 }	This object specifies the domain number used to create a Logical group of PTP devices.
ptpClockPortTransDSInstanceIndex	{ ptpClockPortTransDSEntry 2 }	This object specifies the instance of the clock for this clock type in the given domain.
ptpClockPortTransDSPortNumberIndex	{ ptpClockPortTransDSEntry 3 }	This object specifies the PTP port number associated with this port.
ptpClockPortTransDSPortIdentity	{ ptpClockPortTransDSEntry 4 }	This object specifies the value of the PortIdentity attribute of the local port.
ptpClockPortTransDSlogMinPdelayReqInt	{ ptpClockPortTransDSEntry 5 }	This object specifies the value of the logarithm to the base 2 of the minPdelayReqInterval.
ptpClockPortTransDSFaultyFlag	{ ptpClockPortTransDSEntry 6 }	This object specifies the value TRUE if the port is faulty and FALSE if the port is operating normally.
ptpClockPortTransDSPeerMeanPathDelay	{ ptpClockPortTransDSEntry 7 }	This object specifies, if the delayMechanism used is P2P, the value of the estimate of the current one-way propagation delay, i.e., <meanPathDelay> on the link attached to this port, computed using the peer delay mechanism. If the value of the delayMechanism used is E2E, then the value will be zero.

Table 29: Objects defined in AT-PTP-MIB **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
ptpClockPortAssociateTable	{ ptpMIBClockInfo 11 }	Table of information about a given port's associated ports. ■ For a master port: multiple slave ports that have established sessions with the current master port. ■ For a slave port: the list of masters available for a given slave port. Session information (packets, errors) to be displayed based on availability and scenario.
ptpWellKnownTransportTypes	{ ptpMIBClockInfo 12 }	Well known transport types for PTP communication.
ptpTransportTypeIPversion4	{ ptpWellKnownTransportTypes 1 }	IP version 4.
ptpTransportTypeIPversion6	{ ptpWellKnownTransportTypes 2 }	IP version 6.
ptpTransportTypeEthernet	{ ptpWellKnownTransportTypes 3 }	Ethernet.
ptpTransportTypeDeviceNET	{ ptpWellKnownTransportTypes 4 }	Device NET.
ptpTransportTypeControlNET	{ ptpWellKnownTransportTypes 5 }	Control NET.
ptpTransportTypeIEC61158	{ ptpWellKnownTransportTypes 6 }	IEC61158.
ptpWellKnownEncapsulationTypes	{ ptpMIBClockInfo 13 }	Well Known encapsulation types for PTP communication.
ptpEncapsulationTypeEthernet	{ ptpWellKnownEncapsulationTypes 1 }	Ethernet Encapsulation type.
ptpEncapsulationTypeVLAN	{ ptpWellKnownEncapsulationTypes 2 }	VLAN Encapsulation type.
ptpEncapsulationTypeUDPIPLSP	{ ptpWellKnownEncapsulationTypes 3 }	UDP/IP over MPLS Encapsulation type.
ptpEncapsulationTypePWUDPIPLSP	{ ptpWellKnownEncapsulationTypes 4 }	UDP/IP Pseudowire over MPLS Encapsulation type.
ptpEncapsulationTypePWEthernetLSP	{ ptpWellKnownEncapsulationTypes 5 }	Ethernet Pseudowire over MPLS Encapsulation type.
ptpClockPortAssociateEntry	{ ptpClockPortAssociateTable 1 }	An entry in the table, containing information about a single associated port for the given clockport.
ptpClockPortCurrentDomainIndex	{ ptpClockPortAssociateEntry 1 }	This object specifies the given port's domain number.
tpClockPortCurrentClockTypeIndex	{ ptpClockPortAssociateEntry 2 }	This object specifies the given port's clock type.
ptpClockPortCurrentClockInstanceIndex	{ ptpClockPortAssociateEntry 3 }	This object specifies the instance of the clock for this clock type in the given domain.
ptpClockPortCurrentPortNumberIndex	{ ptpClockPortAssociateEntry 4 }	This object specifies the PTP Port Number for the given port.
ptpClockPortAssociatePortIndex	{ ptpClockPortAssociateEntry 5 }	This object specifies the associated port's serial number in the current port's context

Table 29: Objects defined in AT-PTP-MIB **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
ptpClockPortAssociateAddressType	{ ptpClockPortAssociateEntry 6 }	This object specifies the peer port's network address type used for PTP communication. The OCTET STRING representation of the OID of ptpWellKnownTransportTypes will be used in the values contained in the OCTET STRING.
ptpClockPortAssociateAddress	{ ptpClockPortAssociateEntry 7 }	This object specifies the peer port's network address used for PTP communication.
ptpClockPortAssociatePacketsSent	{ ptpClockPortAssociateEntry 8 }	The number of packets sent to this peer port from the current port. These counters are discontinuous.
ptpClockPortAssociatePacketsReceived	{ ptpClockPortAssociateEntry 9 }	The number of packets received from this peer port by the current port. These counters are discontinuous.
ptpClockPortAssociateInErrors	{ ptpClockPortAssociateEntry 10 }	This object specifies the input errors associated with the peer port. These counters are discontinuous
ptpClockPortAssociateOutErrors	{ ptpClockPortAssociateEntry 11 }	This object specifies the output errors associated with the peer port. These counters are discontinuous.
ptpMIBCompliances	{ ptpMIBConformance 1 }	Conformance Information Definition.
ptpMIBCompliancesSystemInfo	{ ptpMIBCompliances 1 }	Compliance statement for agents that provide read-only support for PTPBASE-MIB to provide system level information of clock devices. Such devices can only be monitored using this MIB module. The Module is implemented with support for read-only. In other words, only monitoring is available by implementing this MODULE-COMPLIANCE.
ptpMIBCompliancesClockInfo	{ ptpMIBCompliances 2 }	Compliance statement for agents that provide read-only support for PTPBASE-MIB to provide clock related information. Such devices can only be monitored using this MIB module. The Module is implemented with support for read-only. In other words, only monitoring is available by implementing this MODULE-COMPLIANCE.

Table 29: Objects defined in AT-PTP-MIB **(continued)**

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
ptpMIBCompliancesClockPortInfo	{ ptpMIBCompliances 3 }	Compliance statement for agents that provide read-only support for PTPBASE-MIB to provide clock port related information. Such devices can only be monitored using this MIB module. The Module is implemented with support for read-only. In other words, only monitoring is available by implementing this MODULE-COMPLIANCE.
ptpMIBCompliancesTransparentClockInfo	{ ptpMIBCompliances 4 }	Compliance statement for agents that provide read-only support for PTPBASE-MIB to provide Transparent clock related information. Such devices can only be monitored using this MIB module. The Module is implemented with support for read-only. In other words, only monitoring is available by implementing this MODULE-COMPLIANCE.
ptpMIBGroups	{ ptpMIBConformance 2 }	Conformance Information Groups.
ptpMIBSystemInfoGroup	{ ptpMIBGroups 1 }	Group which aggregates objects describing system-wide information.
ptpMIBClockCurrentDSGroup	{ ptpMIBGroups 2 }	Group which aggregates objects describing PTP Current Dataset information.
ptpMIBClockParentDSGroup	{ ptpMIBGroups 3 }	Group which aggregates objects describing PTP Parent Dataset information.
ptpMIBClockDefaultDSGroup	{ ptpMIBGroups 4 }	Group which aggregates objects describing PTP Default Dataset information.
ptpMIBClockRunningGroup	{ ptpMIBGroups 5 }	Group which aggregates objects describing PTP running state information.
ptpMIBClockTimepropertiesGroup	{ ptpMIBGroups 6 }	Group which aggregates objects describing PTP Time Properties information.
ptpMIBClockTranparentDSGroup	{ ptpMIBGroups 7 }	Group which aggregates objects describing PTP Transparent Dataset information.
ptpMIBClockPortGroup	{ ptpMIBGroups 8 }	Group which aggregates objects describing information for a given PTP Port.
ptpMIBClockPortDSGroup	{ ptpMIBGroups 9 }	Group which aggregates objects describing PTP Port Dataset information.
ptpMIBClockPortRunningGroup	{ ptpMIBGroups 10 }	Group which aggregates objects describing PTP running interface information.
ptpMIBClockPortTransDSGroup	{ ptpMIBGroups 11 }	Group which aggregates objects describing PTP TransparentDS information.
ptpMIBClockPortAssociateGroup	{ ptpMIBGroups 12 }	Group which aggregates objects describing information on peer PTP ports for a given PTP clock-port.

AT-QOSv2-MIB

See Table 39 for the list of objects defined in the AT-QOSv2-MIB. These objects manage notifications and variables for the QoS module. Objects in this group have the object identifier **atQosv2** { modules 503 } OID 1.3.6.1.4.1.207.8.4.4.4.503.

Table 39: Objects defined in AT-QOSv2-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atQosv2	{ modules 503 } 1.3.6.1.4.1.207.8.4.4.4.503	Contains objects for Quality of Service.
atQosv2Notification	{ atQosv2 0 }	QoS notifications.
atQosv2StormDetectionTrap	{ atQosv2Notification 1 }	Generated when QoS Storm Protection feature detects a storm.
atQosv2NotificationVariables	{ atQosv2 1 }	QoS variables.
atQosv2IfIndex	{ atQosv2NotificationVariables 1 }	The index of the interface where the storm is detected on.
atQosv2VlanId	{ atQosv2NotificationVariables 2 }	The VLAN ID of the interface where the storm is detected on.

AT-RESOURCE-MIB

The AT-RESOURCE-MIB contains objects for displaying system hardware resource and host information, see Table 40. Objects in this group have the object identifier **rsc** { sysinfo 21 }, OID 1.3.6.1.4.1.207.8.4.4.3.21. See "AT-SYSINFO-MIB" on page 121.

Table 40: Objects defined in AT-RESOURCE-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
resource	{ sysinfo 21 } 1.3.6.1.4.1.207.8.4.4.3.21	Contains objects for displaying system hardware resource and host information.
rscBoardTable	{ resource 1}	Table containing information about boards installed in a device. Indexed by: ■ rscStkId ■ rscResourceId.
rscBoardEntry	{ rscBoardTable 1 }	Information about a single board installed in the device.
rscStkId	{ rscBoardEntry 1 }	The ID of the stack member. It is a number from 1 to 8, assigned to a stackable unit by the operating system when it is stacked. A default of 1 is given to a stand-alone unit.
rscResourceId	{ rscBoardEntry 2 }	The resource ID number of the board. It is a number assigned to a hardware resource when the operating system detects its existence. Can be a value in range 1 to 4294967294.
rscBoardType	{ rscBoardEntry 3 }	The type of board. Can be one of the following: ■ Base ■ Expansion ■ Fan module ■ PSU.
rscBoardName	{ rscBoardEntry 4 }	The name of the board. Can be one of the following: ■ SwitchBlade x908 ■ XEM-12S ■ AT-PWR05-AC.
rscBoardId	{ rscBoardEntry 5 }	The ID number of the board. Its value is an Allied Telesis assigned number, such as 274 for the XEM-12S, or 255 for the AT-9924Ts.
rscBoardBay	{ rscBoardEntry 6 }	The board installation location. Its value can be Bay1, Bay2, PSU1. For a base board, it has a value of a single character space.
rscBoardRevision	{ rscBoardEntry 7 }	The revision number of the board.
rscBoardSerialNumber	{ rscBoardEntry 8 }	The serial number of the board.
hostInfoTable	{ resource 2 }	Table containing general system information. Indexed by rscStkId.
hostInfoEntry	{ hostInfoTable 1 }	Information about a single system parameter.
hostInfoDRAM	{ hostInfoTable 2 }	The host DRAM information.
hostInfoFlash	{ hostInfoTable 3 }	The host Flash information.
hostInfoUptime	{ hostInfoTable 4 }	The host up-time.
hostInfoBootloaderVersion	{ hostInfoTable 5 }	The host boot loader version.

AT-SETUP-MIB

AT-SETUP-MIB defines objects for managing software installation and configuration files, see Figure 6 and Table 41. Objects in this group have the object identifier **setup** { modules 500 }, OID 1.3.6.1.4.1.207.8.4.4.4.500. To see how to use these MIB objects to upgrade to a new software version and boot configuration file, see the SNMP Feature Overview and Configuration Guide. For objects used for file copying, see "AT-FILEv2-MIB" on page 46.

Figure 6: The AT-SETUP-MIB sub-tree

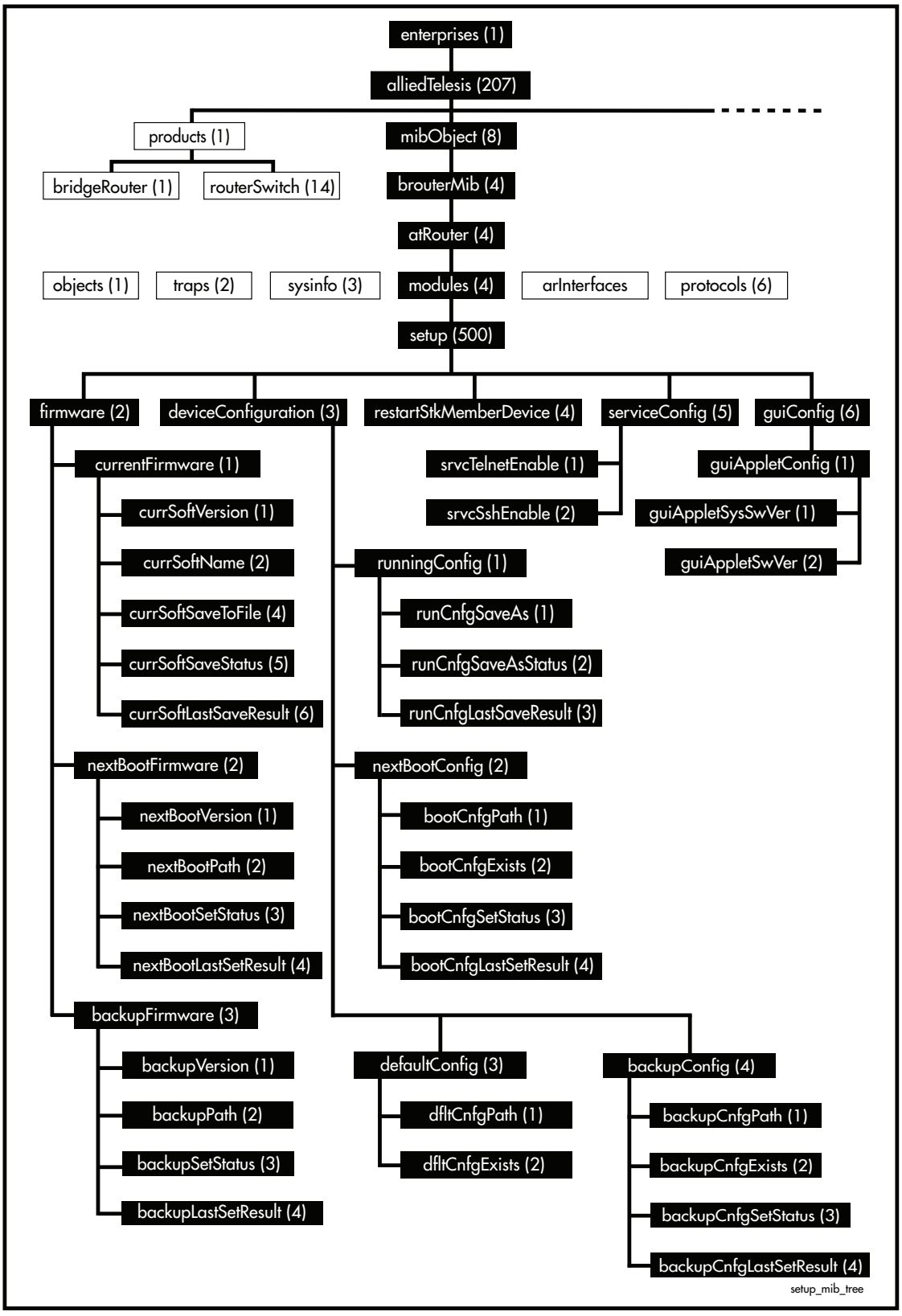


Table 41: Objects defined in AT-SETUP-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
restartDevice	{ setup 1 } 1.3.6.1.4.1.207.8.4.4.4.500.1	Object for restarting the device. When set to '1', the device will restart immediately. **Note:** This object has been deprecated. Use instead the restartStkMemberDevice object.
firmware	{ setup 2 } 1.3.6.1.4.1.207.8.4.4.4.500.2	Objects for managing the software version files that the device will install and run.
currentFirmware	{ firmware 1 }	Information about the current software version installed on the device.
currSoftVersion	{ currentFirmware 1 }	Current software version.
currSoftName	{ currentFirmware 2 }	Current software name.
currSoftSaveAs	{ currentFirmware 3 }	The file name to save the currently running software to the root of the Flash. Only one save operation can be executed at a time across all SNMP users. **Note:** This object has been deprecated. Use instead the currSoftSaveToFile, currSoftSaveStatus and currSoftLastSaveResult objects.
currSoftSaveToFile	{ currentFirmware 4 }	Set with a URL to save the currently running software to the root of Flash or USB Flash drive (e.g. 'flash:/filename.rel' or 'USB:/filename.rel'). The URL must not contain whitespace characters. Only one save operation can be executed at a time across all SNMP users and an operation may not be started unless the current value of currSoftSaveStatus is 'idle'. Immediately upon executing the set action, the actual firmware save operation is started and will continue on the device until it has completed or a failure occurs. When read, this object will return the URL of the last firmware save operation that was attempted.
currSoftSaveStatus	{ currentFirmware 5 }	This object will return the status of any current operation to store the running software to a release file. The following values may be returned: ■ (idle) - there is no release file save operation in progress. ■ (success) - the last release file save operation completed successfully. ■ (failure) - the last release file save operation failed. ■ (saving) - a release file save operation is currently in progress. When a read of this object returns a value of 'success' or 'failure', it will immediately be reset to 'idle' and a new operation may be initiated if desired. A detailed description of the last completed operation may be determined by reading currSoftLastSaveResult.
currSoftLastSaveResult	{ currentFirmware 6 }	Gives an indication of the result of the last completed SNMP operation to save the running firmware to a release file.

Table 41: Objects defined in AT-SETUP-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
nextBootFirmware	{ firmware 2 }	Information about the software version to be installed on the device when booting.
nextBootVersion	{ nextBootFirmware 1 }	Provides information on the software version (major.minor.interim, for example version 5.4.1) that the device will boot from. A zero will be returned if the version cannot be determined.
nextBootPath	{ nextBootFirmware 2 }	The full path to the release file that will be used the next time the device is rebooted. The URL must not contain whitespace characters. Only one set operation can be executed at a time across all SNMP users and an operation may not be started unless the current value of nextBootSetStatus is 'idle'. Immediately upon executing the set action, the system will attempt to set the new configuration path, and the process will continue on the device until it has completed or a failure occurs. This object can be set with an empty string in order to clear the current boot firmware. Otherwise, the path should be of the form ‘flash:/filename.cfg’ or 'card:/filename.cfg'. In order to set this object, the file must meet the following conditions: ■ it must exist. ■ it must be located in the root of Flash (on the active master in a stacked environment) or USB flash drive. ■ it must not be the same as the backup release file. ■ it must have a .rel suffix. ■ it must pass several internal checks to ensure that it is a genuine release file. ■ in a stacked environment, there must be enough disk space available to store the release file on each stack member.
nextBootSetStatus	{ nextBootFirmware 3 }	Returns the status of any current operation to set the next boot release file. The following values may be returned: ■ 1 (idle) - there is no boot release setting operation in progress. ■ 2 (success) - the last boot release setting operation completed successfully. ■ 3 (failure) - the last boot release setting operation failed. ■ 5 (syncing) - a boot release setting operation is currently in progress and the file is being synchronized across the stack. When a read of this object returns a value of 'success' or 'failure', it will immediately be reset to 'idle' and a new operation may be initiated if desired. A detailed description of the last completed operation may be determined by reading nextBootLastSetResult.
nextBootLastSetResult	{ nextBootFirmware 4 }	Gives an indication of the result of the last completed SNMP operation to set the boot release **filename**.
backupFirmware	{ firmware 3 }	Information about the backup software version and path.
backupVersion	{ backupFirmware 1 }	Provides information on the backup software version (major.minor.interim, for example version 5.4.1) that the device will boot from. A zero will be returned if the version cannot be determined.

Table 41: Objects defined in AT-SETUP-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
backupPath	{ backupFirmware 2 }	The full path to the backup release file that will be used the next time the device is rebooted. The URL must not contain whitespace characters. Only one set operation can be executed at a time across all SNMP users and an operation may not be started unless the current value of backupSetStatus is 'idle'. Immediately upon executing the set action, the system will attempt to set the new configuration path, and the process will continue on the device until it has completed or a failure occurs. This object can be set with an empty string in order to clear the current backup firmware. Otherwise, the path should be of the form 'flash:/filename.cfg' or 'card:/filename.cfg'. In order to set this object, the file must meet the following conditions: ■ it must exist. ■ it must be located in the root of Flash (on the active master in a stacked environment) or USB flash drive. ■ it must not be the same as the configured main release file. ■ it must have a .rel suffix. ■ it must pass several internal checks to ensure that it is a genuine release file. ■ in a stacked environment, there must be enough disk space available to store the release file on each stack member.
backupSetStatus	{ backupFirmware 3 }	Returns the status of any current operation to set the backup boot release file. The following values may be returned: ■ 1 (idle) - there is no backup boot release setting operation in progress. ■ 2 (success) - the last backup boot release setting operation completed successfully. ■ 3 (failure) - the last backup boot release setting operation failed. ■ 5 (syncing) - a backup boot release setting operation is currently in progress and the file is being synchronized across the stack. When a read of this object returns a value of 'success' or 'failure', it will immediately be reset to 'idle' and a new operation may be initiated if desired. A detailed description of the last completed operation may be determined by reading backupLastSetResult.
backupLastSetResult	{ backupFirmware 4 }	Gives an indication of the result of the last completed SNMP operation to set the backup boot release filename.
deviceConfiguration	{ setup 3 1.3.6.1.4.1.207.8.4.4.4.500.3	Objects for managing device configuration.
runningConfig	{ deviceConfiguration 1 }	Set with a URL to save the running configuration to the Flash or Card filesystem using that filename (e.g.'flash:/myconfig.cfg' or 'card:/myconfig.cfg').

Table 41: Objects defined in AT-SETUP-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
runCnfgSaveAs	{ runningConfig 1 }	Set with a URL to save the currently running software to the root of Flash or USB flash drive (e.g. 'flash:/filename.rel' or 'usb:/filename.rel'). The URL must not contain whitespace characters. Only one set operation can be executed at a time across all SNMP users and an operation may not be started unless the current value of runCnfgSaveAsStatus is 'idle'. Immediately upon executing the set action, the system will attempt to save the running configuration and the process will continue on the device until it has completed or a failure occurs. When read, this object will return the URL of the last firmware save operation that was attempted.
runCnfgSaveAsStatus	{ runningConfig 2 }	Returns the status of any current operation to save the running configuration. The following values may be returned: ■ (idle) - there is no config file save operation in progress. ■ (success) - the last config file save operation completed successfully. ■ (failure) - the last config file save operation failed. ■ (saving) - a config file save operation is currently in progress. When a read of this object returns a value of 'success' or 'failure', it will immediately be reset to 'idle' and a new operation may be initiated if desired. A detailed description of the last completed operation may be determined by reading runCnfgLastSaveResult.
runCnfgLastSaveResult	{ runningConfig 3 }	Gives an indication of the result of the last completed SNMP operation to save the running configuration.
nextBootConfig	{ deviceConfiguration 2 }	The full path of the configuration file to be used the next time the device restarts. There is no guarantee that the file referenced exists (i.e. it may have been deleted since the configuration path was last set).
bootCnfgPath	{ nextBootConfig 1 }	The full path to the configuration file that will be used the next time the device is rebooted. The URL must not contain whitespace characters. Only one set operation can be executed at a time across all SNMP users and an operation may not be started unless the current value of bootCnfgSetStatus is 'idle'. Immediately upon executing the set action, the system will attempt to set the new configuration path, and the process will continue on the device until it has completed or a failure occurs. This object can be set with an empty string in order to clear the current boot configuration. Otherwise, the path should be of the form ‘flash:/myconfig.cfg’ or 'card:/filename.cfg'. In order to set this object, the file must meet the following conditions: ■ it must exist. ■ it must be located in the root of Flash (on the active master in a stacked environment) or USB flash drive. ■ it must have a .cfg suffix. ■ in a stacked environment, there must be enough disk space available to store the configuration file on each stack member.

Table 41: Objects defined in AT-SETUP-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
bootCnfgExists	{ nextBootConfig 2 }	This object will return the value TRUE if the currently defined boot configuration file exists, or FALSE if it does not.
bootCnfgSetStatus	{ nextBootConfig 3 }	Returns the status of any current operation to set the next boot configuration file. The following values may be returned: ■ 1 (idle) - there is no boot configuration setting operation in progress. ■ 2 (success) - the last boot configuration setting operation completed successfully. ■ 3 (failure) - the last boot configuration setting operation failed. ■ 5 (syncing) - a boot configuration setting operation is currently in progress and the file is being synchronized across the stack. When a read of this object returns a value of 'success' or 'failure', it will immediately be reset to 'idle' and a new operation may be initiated if desired. A detailed description of the last completed operation may be determined by reading bootCnfgLastSetResult.
bootCnfgLastSetResult	{ nextBootConfig 4 }	Gives an indication of the result of the last completed SNMP operation to set the boot configuration **filename**.
defaultConfig	{ deviceConfiguration 3 }	The full path of the configuration file to be used upon device restart, if no user-defined boot or backup configuration files can be found. There is no guarantee that the file referenced exists (i.e. it may have been deleted).
dfltCnfgPath	{ defaultConfig 1 }	The full path of the configuration file to use as backup when the device is rebooted. This object is not settable. The default configuration file is always ‘flash:/default.cfg’.
dfltCnfgExists	{ defaultConfig 2 }	This object will return the value TRUE if the currently defined default configuration file exists, or FALSE if it does not.
backupConfig	{ deviceConfiguration 4 }	The full path of the backup configuration file to be used the next time the device restarts, if the boot configuration file cannot be accessed. There is no guarantee that the file referenced exists (i.e. it may have been deleted since the backup configuration path was last set).

Table 41: Objects defined in AT-SETUP-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
backupCnfgPath	{ backupConfig 1 }	The full path to the backup configuration file that will be used the next time the device is rebooted. The URL must not contain whitespace characters. Only one set operation can be executed at a time across all SNMP users and an operation may not be started unless the current value of backupCnfgSetStatus is 'idle'. Immediately upon executing the set action, the system will attempt to set the new backup configuration path, and the process will continue on the device until it has completed or a failure occurs. This object can be set with an empty string in order to clear the current boot configuration. Otherwise, the path should be of the form ‘flash:/myconfig.cfg’ or 'card:/filename.cfg'. In order to set this object, the file must meet the following conditions: ■ it must exist. ■ it must be located in the root of Flash (on the active master in a stacked environment) or USB flash drive. ■ it must have a .cfg suffix. ■ in a stacked environment, there must be enough disk space available to store the configuration file on each stack member.
backupCnfgExists	{ backupConfig 2 }	This object will return the value TRUE if the currently defined backup configuration file exists, or FALSE if it does not.
backupCnfgSetStatus	{ backupConfig 3 }	Returns the status of any current operation to set the next backup boot configuration file. The following values may be returned: ■ 1 (idle) - there is no backup boot configuration setting operation in progress. ■ 2 (success) - the last backup boot configuration setting operation completed successfully. ■ 3 (failure) - the last backup boot configuration setting operation failed. ■ 5 (syncing) - a backup boot configuration setting operation is currently in progress and the file is being synchronized across the stack. When a read of this object returns a value of 'success' or 'failure', it will immediately be reset to 'idle' and a new operation may be initiated if desired. A detailed description of the last completed operation may be determined by reading backupCnfgLastSetResult.
backupCnfgLastSetResult	{ backupConfig 4 }	Gives an indication of the result of the last completed SNMP operation to set the backup boot configuration **filename**.

Table 41: Objects defined in AT-SETUP-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
restartStkMemberDevice	{ setup 4 } 1.3.6.1.4.1.207.8.4.4.4.500.4	This object causes a specified device to restart immediately. The restart is initiated by setting its value to the device's stack member ID. Setting its value to zero will cause all devices in the stack, or a standalone device, to restart. Reading the object will always return zero.
serviceConfig	{ setup 5 } 1.3.6.1.4.1.207.8.4.4.4.500.5	This object represents the state of the telnet server of a device.
srvcTelnetEnable	{ serviceConfig 1 }	This object is used to either read or set the state of the telnet server on a device. Telnet can be enabled by setting the value of this object to ‘enable(1)’ or can be disabled by setting the value ‘disable(2)’.
srvcSshEnable	{ serviceConfig 2 }	This object is used to either read or set the state of the SSH server on a device. SSH can be enabled by setting the value of this object to ‘enable(1)’ or can be disabled by setting the value ‘disable(2)’.
guiConfig	{ setup 6 } 1.3.6.1.4.1.207.8.4.4.4.500.6	This object represents the system software release that the currently selected GUI applet was designed to run on.
guiAppletConfig	{ guiConfig 1 }	This object represents the system software release that the currently selected GUI applet was designed to run on.
guiAppletSysSwVer	{ guiAppletConfig 1 }	This object represents the system software release that the currently selected GUI applet was designed to run on. The system automatically searches for GUI applet files that reside in the root directory of the Flash memory, and selects the latest available file that is applicable to the currently running system software. This is the applet that will be uploaded to a user's web browser when they initiate the GUI.
guiAppletSwVer	{ guiAppletConfig 2 }	This object represents the software version of the currently selected GUI applet. The system automatically searches for GUI applet files residing in the root directory of the Flash memory, and selects the latest available one that is applicable to the currently running system software. This is the applet that will be uploaded to a user's web browser when they initiate the GUI.

AT-SYSINFO-MIB

AT-SYSINFO-MIB defines objects that describe generic system information and environmental monitoring. Objects in this group have the object identifier **sysinfo** { atRouter 3 }, OID 1.3.6.1.4.1.207.8.4.4.3. Table 42 lists the objects supported by the AlliedWare Plus sysinfo MIB.

Table 42: Objects defined in AT-SYSINFO-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
sysinfo	{ atRouter 3 } (1.3.6.1.4.1.207.8.4.4.3)	Subtree containing generic system information.
fanAndPs	{sysinfo 1 } (1.3.6.1.4.1.207.8.4.4.3.1)	A collection of objects for monitoring fans and power supplies. For devices running the AlliedWare Plus™ Operating System, these objects are superceded by objects in the AT-ENVMON-MIB (see "AT-ENVMONv2-MIB" on page 33).
restartGroup	{sysinfo 2 }	A collection of objects and traps for activating and monitoring restarts. This group is not supported by devices running the AlliedWare Plus™ Operating System.
cpu	{sysinfo 3 }	A collection of objects containing information about the CPU utilization over different periods of time. All values are expressed as a percentage - integer in range 0 to 100.
cpuUtilisationMax	{ cpu 1 }	Maximum CPU utilization since the device was last restarted.
cpuUtilisationAvg	{ cpu 2 }	Average CPU utilization since the device was last restarted.
cpuUtilisationAvgLastMinute	{ cpu 3 }	Average CPU utilization over the past minute.
cpuUtilisationAvgLast10Seconds	{ cpu 4 }	Average CPU utilization over the past ten seconds.
cpuUtilisationAvgLastSecond	{ cpu 5 }	Average CPU utilization over the past second.
cpuUtilisationAvgMaxLast5Minutes	{ cpu 6 }	Maximum CPU utilization over the last 5 minutes.
cpuUtilisationAvgLast5Minutes	{ cpu 7 }	Average CPU utilization over the past 5 minutes.
cpuUtilisationStackTable	{ cpu 8 }	A list of stack members.
cpuUtilisationStackEntry	{ cpuUtilisationStackTable 1 }	A set of parameters that describe the CPU utilization of a stack member.

Table 42: Objects defined in AT-SYSINFO-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
cpuUtilisationStackId	{ cpuUtilisationStackEntry 1 }	Stack member ID.
cpuUtilisationStackMax	{ cpuUtilisationStackEntry 2 }	Maximum CPU utilization since the router was last restarted. Expressed as a percentage.
cpuUtilisationStackAvg	{ cpuUtilisationStackEntry 3 }	Average CPU utilization since the router was last restarted. Expressed as a percentage.
cpuUtilisationStackAvgLastMinute	{ cpuUtilisationStackEntry 4 }	Average CPU utilisation over the past minute. Expressed as a percentage.
cpuUtilisationStackAvgLast10Seconds	{ cpuUtilisationStackEntry 5 }	Average CPU utilisation over the past ten seconds. Expressed as a percentage.
cpuUtilisationStackAvgLastSecond	{ cpuUtilisationStackEntry 6 }	Average CPU utilisation over the past second. Expressed as a percentage.
cpuUtilisationStackMaxLast5Minutes	{ cpuUtilisationStackEntry 7 }	Maximum CPU utilisation over the last 5 minutes. Expressed as a percentage.
cpuUtilisationStackAvgLast5Minutes	{ cpuUtilisationStackEntry 8 }	Average CPU utilisation over the past 5 minutes. Expressed as a percentage.
sysTemperature	{ sysinfo 4 }	A collection of objects and traps for monitoring and managing the temperature status. For devices running the AlliedWare Plus™ Operating System.
atContactDetails	{sysinfo 5 }	Contact details for Allied Telesis.
memory	{ sysinfo 7 }	A collection of objects and traps for monitoring memory usage and status.
xem	{ sysinfo 11 }	A collection of objects for monitoring XEMs installed in the device. Objects under this portion of the OID are shown in the “AT-XEM-MIB” on page 138.
atEnvMonv2	{ sysinfo 12 }	AT Environment Monitoring v2 MIB for managing and reporting data relating to voltage rails, fan speeds, temperature sensors and power supply units. Objects under this portion of the OID are shown in the “AT-ENVMONv2-MIB” on page 33.
vcstack	{ sysinfo 13 }	A collection of objects for managing Virtual Chassis Stacking in AlliedWare Plus™. See “AT-VCSTACK-MIB” on page 131.
atPortInfo	{ sysinfo 14 }	Objects containing information about the transceiver of an interface.
atPortInfoTransceiverTable	{ atPortInfo 1 }	A table of information about the transceiver of a interface.
atPortInfoTransceiverEntry	{ atPortInfoTransceiverTable 1 }	The description, the transceiver type of a interface.
atPortInfoTransceiverifIndex	{ atPortInfoTransceiverEntry 1 }	The ifIndex for the interface represented by this entry of the interfaces table.

Table 42: Objects defined in AT-SYSINFO-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atPortInfoTransceiverType	{ atPortInfoTransceiverEntry 2 }	This object indicates the type of transceiver on an interface.
atPortRenumberEvents	{ atPortInfo 2 }	The number of times that port number values (represented by the dot1dBasePort object in BRIDGE-MIB), have been re-assigned due to stack member leave/join events or XEM hot-swap events, since the system was initialised.
atVlanInfo	{ sysinfo 16 }	A collection of objects for counting bytes or incoming frames within a selected VLAN. Note that these objects are only appropriate for the IX5, x510 and x610 series products. Objects under this portion of the OID are shown in the “AT-VLAN-MIB” on page 136.
user	{ sysinfo 20 }	Contains objects for displaying information of users currently logged into a device, or configured in its local database. Objects under this portion of the OID are shown in the “AT-USER-MIB” on page 129.
resource	{ sysinfo 21 }	Contains objects for displaying hardware resource information. Objects under this portion of the OID are shown in the “AT-RESOURCE-MIB” on page 112.
license	{ sysinfo 22 }	This MIB, is used for listing applied software licenses, adding new licenses, and deleting existing licenses. Objects under this portion of the OID are shown in the “AT-LICENSE-MIB” on page 60.
chassis	{ sysinfo 23 }	This MIB is used for accessing trap notifications on chassis based products. Note that these objects are only appropriate for the x8100 series products. Objects under this portion of the OID are shown in the “AT-HHM-MIB” on page 57.
hhm	(sysinfo 24 }	This MIB is used for Hardware Health Monitoring notification log messages. Objects under this portion of the OID are shown in the “AT-HHM-MIB” on page 57.
LinkTrap	{ sysinfo 25 }	This MIB is used for generating traps when an interface is linked up or down. Objects under this portion of the OID are shown in the “AT-LINKTRAP-MIB” on page 66.
Fiber Monitoring	{ sysinfo 27 }	This MIB contains managed objects definition and notification definitions for AT Fiber Monitoring. Objects under this portion of the OID are shown in the “AT-FIBER-MONITORING-MIB” on page 44.

Table 42: Objects defined in AT-SYSINFO-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
Digital Diagnostics	{ sysinfo 28 }	This MIB is used for the Digital Diagnostic Monitoring for optical pluggables. Objects under this portion of the OID are shown in the "AT-PLUGGABLE-DIAGNOSTICS-MIB" on page 81.

AT-TRIGGER-MIB

AT-TRIGGER-MIB defines objects for managing triggers Table 43. Objects in this group have the object identifier **trigger** { modules 53 }, OID 1.3.6.1.4.1.207.8.4.4.4.53. All objects in this group have read only access.

Table 43: Objects defined in AT-TRIGGER-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
triggerTraps	{ trigger 0 } 1.3.6.1.4.1.207.8.4.4.4.53.0	Sub-tree for all trigger traps.
triggerTrap	{ triggerTraps 1 }	Notification generated when a trigger is activated. It returns the value of triggerLastTriggerActivated.
triggerLastTriggerActivated	{ trigger 1 }	Trigger number of the most recent trigger activated on the switch.
triggerConfigInfoTable	{ trigger 9 }	Table of information about each trigger that has been configured, indexed by triggerNumber.
triggerConfigInfoEntry	{ triggerConfigInfoTable 1 }	Information about the configuration of a single trigger.
triggerNumber	{ triggerConfigInfoEntry 1 }	ID number of the trigger. Values are in range 1-250.
triggerName	{ triggerConfigInfoEntry 2 }	Name and description of the trigger.
triggerTypeDetail	{ triggerConfigInfoEntry 3 }	Trigger type and its activation conditions.
triggerActiveDaysOrDate	{ triggerConfigInfoEntry 4 }	The days of a week or the date on which the trigger can be activated.
triggerActivateAfter	{ triggerConfigInfoEntry 5 }	Time after which the trigger can be activated.
triggerActivateBefore	{ triggerConfigInfoEntry 6 }	Time before which the trigger can be activated.
triggerActiveStatus	{ triggerConfigInfoEntry 7 }	Whether or not the trigger can be activated.
triggerTestMode	{ triggerConfigInfoEntry 8 }	Whether or not the trigger is operating in diagnostic (test) mode.
triggerSnmpTrap	{ triggerConfigInfoEntry 9 }	Whether or a not an SNMP trap will be generated when the trigger is activated.
triggerRepeatTimes	{ triggerConfigInfoEntry 10 }	Whether the trigger can repeat an unlimited number of times (continuous) or a specified number of times. If the trigger can repeat only a specified number of times, then the number of times the trigger has already been activated is displayed in brackets.
triggerLasttimeModified	{ triggerConfigInfoEntry 11 }	Date and time that the trigger configuration was last modified.

Table 43: Objects defined in AT-TRIGGER-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
triggerNumberOfActivation	{ triggerConfigInfoEntry 12 }	Number of times the trigger has been activated since the last restart of the device.
triggerLasttimeActivation	{ triggerConfigInfoEntry 13 }	Date and time that the trigger was last activated.
triggerNumberOfScripts	{ triggerConfigInfoEntry 14 }	Number of scripts that this trigger will execute. Values are in range 0-5.
triggerScript1	{ triggerConfigInfoEntry 15 }	Name of the first script that this trigger will execute if the trigger is activated.
triggerScript2	{ triggerConfigInfoEntry 16 }	Name of the second script that this trigger will execute if the trigger is activated.
triggerScript3	{ triggerConfigInfoEntry 17 }	Name of the third script that this trigger will execute if the trigger is activated.
triggerScript4	{ triggerConfigInfoEntry 18 }	Name of the fourth script that this trigger will execute if the trigger is activated.
triggerScript5	{ triggerConfigInfoEntry 19 }	Name of the fifth script that this trigger will execute if the trigger is activated.
triggerCounters	{ trigger 10 }	Collection of counters for trigger activations.
triggerNumOfActivation	{ triggerCounters 1 }	Number of times a trigger has been activated.
triggerNumOfActivationToday	{ triggerCounters 2 }	Number of times a trigger has been activated today.
triggerNumOfPerodicActivationToday	{ triggerCounters 3 }	Number of times a periodic trigger has been activated today.
triggerNumOfInterfaceActivationToday	{ triggerCounters 4 }	Number of times an interface trigger has been activated today.
triggerNumOfResourceActivationToday	{ triggerCounters 5 }	Number of times a CPU or memory trigger has been activated today.
triggerNumOfRebootActivationToday	{ triggerCounters 6 }	Number of times a reboot trigger has been activated today.
triggerNumOfPingPollActivationToday	{ triggerCounters 7 }	Number of times a ping-poll trigger has been activated today.
triggerNumOfStackMasterFailActivationToday	{ triggerCounters 8 }	Number of times a stack master fail trigger has been activated today.
triggerNumOfStackMemberActivationToday	{ triggerCounters 9 }	Number of times a stack member trigger has been activated today.
triggerNumOfStackXemStkActivationToday	{ triggerCounters 10 }	Number of times a stack XEM trigger has been activated today.

AT-UDLD-MIB

see Table 44 for the list of objects defined in the AT-UDLD-MIB. These objects contain definitions of managed objects for the UniDirectional Link Detection (UDLD) protocol. Objects in this group have the object identifier **atQosv2** { modules 550 } OID 1.3.6.1.4.1.207.8.4.4.4.550.

Table 44: Objects defined in AT-UDLD-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atUdld	{ modules 550 } 1.3.6.1.4.1.207.8.4.4.4.550	Contains objects for UDLD.
atUdldTrap	{ atUdld 0 }	UDLP traps.
atUdldPortDisabledTrap	{ atUdldTrap 1 }	Generated when UDLD feature disabled an interface when detecting uni-directional link.
atUdldPortRecoveredTrap	{ atUdldTrap 2 }	Generated when an interface recovers from error disable status when detecting uni-directional link.
atUdldIfIndex	{ atUdld 1 }	The index value of an interface of which the link status is changed.

AT-UFO-MIB

AT-UFO-MIB defines objects for managing Upstream Forwarding Only (UFO) for private VLANs, see Table 45. Objects in this group have the object identifier **atUfo** in { modules 605 }, OID 1.3.6.1.4.1.8.4.4.4.605

Table 45: Objects defined by the AT-UFO-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atUfo	{ modules 605 } (1.3.6.1.4.1.207.8.4.4.4.605)	This MIB file contains definitions of managed objects for the UFO module.
atUfoTraps	{ atUfo 0 } (605.0)	Traps generated when a UFO private VLAN state changes. One of the following: ■ atUfoVlanBlackHoleTrap ■ atUfoBlackHoleAlarmTrap.
atUfoVlanBlackHoleTrap	{ atUfoTraps 1 } (605.0.1)	Generated when a UFO private VLAN changes the blackhole state to one of the following: ■ atUfoVlanId ■ atUfoPreviousState ■ atUfoCurrentState.
atUfoBlackHoleAlarmTrap	{ atUfoTraps 2 } (605.0.2)	Generated when the first UFO private VLAN transitions to the blackhole state indicating an alarm has been triggered, and, when the last UFO private VLAN transitions to the non-blackhole state it indicates the alarm has been cleared: ■ atUfoAlarmState.
atUfoTrapVariables	{ atUfo 1 } (605.1)	UFO Trap variables.
atUfoVlanId	{ atUfoTrapVariables 1 } (605.1.1)	VLAN identifier of the UFO private VLAN.
atUfoPreviousState	{ atUfoTrapVariables 2 } (605.1.2)	The previous blackhole state of the UFO private VLAN. This has the value of 1 (true) if the UFO private VLAN was not blackholed. Otherwise the value is 2 (false) if the UFO VLAN was being blackholed.
atUfoCurrentState	{ atUfoTrapVariables 3 } (605.1.3)	The current blackhole state of the UFO private VLAN. This has the value of 1 (true) if the UFO private VLAN is not blackholed. Otherwise the value is 2 (false) if the UFO VLAN is being blackholed.
atUfoAlarmState	{ atUfoTrapVariables 4 } (605.1.4)	This has the value of 1 (true) if one or more UFO private VLAN/s is in the blackhole state, indicating an alarm has been triggered. Otherwise the value is 2 (false) indicating the alarm has been cleared.

AT-USER-MIB

The AT-USER-MIB contains objects for displaying information about users currently logged into a device, or configured in the Local User Database of the device Table 46. Objects in this group have the object identifier **user** { sysinfo 20 }, OID 1.3.6.1.4.1.207.8.4.4.3.20.

Table 46: Objects defined in AT-USER-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
userInfoTable	{ user 1 } 1.3.6.1.4.1.207.8.4.4.3.20.1	Table containing information about users. Each entry in the table represents a user currently logged into the device. Indexed by: rscBoardType and rscBoardIndex.
userInfoEntry	{ userInfoTable 1 }	Information about a single user logged into the device.
userInfoType	{ userInfoEntry 1 }	The type of connection through which the user logged into the device. Can be: ■ console (1) ■ aux (2) ■ telnet (3) ■ script (4) ■ stack (5).
userInfoIndex	{ userInfoEntry 2 }	Index of the line upon which the user logged into the device. Can be a value in range 1 to 16.
userInfoName	{ userInfoEntry 3 }	User name of the user logged into the device.
userInfoPrivilegeLevel	{ userInfoEntry 4 }	The user's privilege level. Can be a value in range 1 to 15.
userInfoIdleTime	{ userInfoEntry 5 }	The amount of time since the user was last active, in the form hh:mm:ss.
userInfoLocation	{ userInfoEntry 6 }	The user location or login method. It can be an IP Address used by the user to telnet into the device, or an asyn port.
userInfoPasswordLifetime	{ userInfoEntry 7 }	The number of days remaining until the user's password expires. Depending on the current user setting it will display one of the following: ■ No Expiry - the password will never expire (default setting). ■ x days - where x is the remaining lifetime of the current password (maximum lifetime value is 1000 days). ■ -x days (expired) - indicating that the current password expired x days ago.
userInfoPasswordLastChange	{ userInfoEntry 8 }	The number of days since the password was last altered.

Table 46: Objects defined in AT-USER-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
userConfigTable	{ user 2 } 1.3.6.1.4.1.207.8.4.4.3.20.2	Table containing user configuration information. Each entry in the table relates to a user configured in the Local User Database of the device. Indexed by userConfigIndex.
userConfigEntry	{ userConfigTable 1 }	Information about a single user configured in the Local User Database of the device.
userConfigIndex	{ userConfigEntry 1 }	Unique number used to identify entries in the userConfigTable.
userConfigName	{ userConfigEntry 2 }	The user's name.
userConfigPrivilegeLevel	{ userConfigEntry 3 }	The privilege level granted to the user. Can be a value in range 1 to 15.
userSecurityPasswordRules	{ user 3 } 1.3.6.1.4.1.207.8.4.4.3.20.3	Information about user password security rules.
userSecurityPasswordHistory	{ userSecurityPasswordRules 1 }	The number of previous passwords that are retained for comparison when a user password is created. A new password must be unique when compared against the previous history. A value of 0 represents no restriction. The maximum number of retained passwords is 15.
userSecurityPasswordLifetime	{ userSecurityPasswordRules 2 }	The maximum number of days that the password may persist before a change is required. 0 means no expiry. The maximum value is 1000.
userSecurityPasswordWarning	{ userSecurityPasswordRules 3 }	The number of days before the password expires that a warning message is displayed when the user logs in. A value of 0 indicates no warning. The maximum value is 1000 but must always be less than the password lifetime.
userSecurityPasswordMinLength	{ userSecurityPasswordRules 4 }	The minimum allowable password length.
userSecurityPasswordMinCategory	{ userSecurityPasswordRules 5 }	The minimum number of different categories that the password must satisfy to be considered valid. Categories are split into four groups: ■ upper-case letters ■ lower-case letters ■ digits ■ special symbols. ASCII characters not included in the previous three categories.
userSecurityPasswordForced	{ userSecurityPasswordRules 6 }	Whether or not a user with an expired password is forced to change their password at the next login. At login a user with an expired password is prompted to change their password. If the new password meets the current security password rules the user is allowed to log in, otherwise they are rejected.
userSecurityPasswordReject	{ userSecurityPasswordRules 7 }	Whether or not a user login attempt with an expired password is rejected. If the user is not rejected then they can log in.

AT-VCSTACK-MIB

AT-VCSTACK-MIB defines objects for managing Virtual Chassis Stacking, see Table 47. Objects in this group have the object identifier **vcstack** { sysinfo 13 }, OID 1.3.6.1.4.1.207.8.4.4.3.13.

Figure 7 on page 131 shows the tree structure of the AT-VCSTACK objects.

Figure 7: The AT-VCSTACK MIB sub-tree

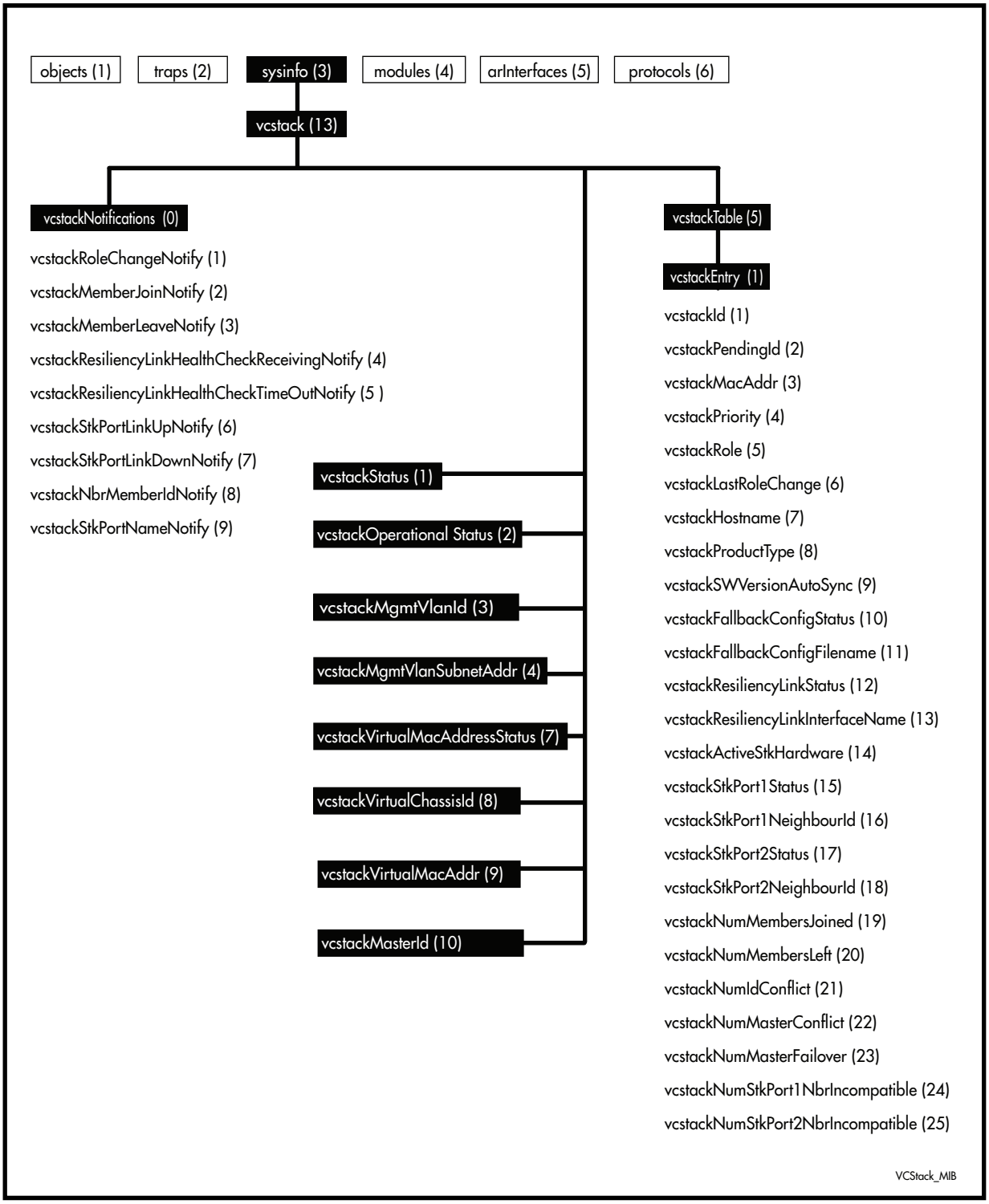


Table 47: Objects defined in AT-VCSTACK-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
vcstack	{ sysinfo (13) } 1.3.6.1.4.1.207.8.4.4.3.13.	Overall stack status.
vcstackNotifications	{ vcstack 0 }	List of traps (notifications) generated for the stack.
vcstackRoleChangeNotify	{ vcstackNotifications 1 }	The stack status can take one of the following states: ■ normalOperation (1) ■ operatingInFailoverState (2) ■ standaloneUnit (3) ■ ringTopologyBroken (4).
vcstackMemberJoinNotify	{ vcstackNotifications 2 }	Notification generated when a member joins the stack. Displays the objects: ■ vcstackId ■ vcstackNbrMemberId.
vcstackMemberLeaveNotify	{ vcstackNotifications 3 }	Notification generated when a member leaves the stack. Displays the objects: ■ vcstackId ■ vcstackNbrMemberId.
vcstackResiliencyLinkHealthCheckReceivingNotify	{ vcstackNotifications 4 }	Notification generated when the resiliency link is activated. Displays the objects: ■ vcstackId ■ vcstackResiliencyLinkInterfaceName.
vcstackResiliencyLinkHealthCheckTimeOutNotify	{ vcstackNotifications 5 }	Notification generated when the backup member's receive timer has timed-out, indicating that the Backup has lost contact with the Master via the resiliency link. Displays the objects: ■ vcstackId ■ vcstackResiliencyLinkInterfaceName.
vcstackStkPortLinkUpNotify	{ vcstackNotifications 6 }	Notification generated when the stack port link is up. Displays the objects: ■ vctasckId ■ vcstackStkPortName.
vcstackStkPortLinkDownNotify	{ vcstackNotifications 7 }	Notification generated when the stack port link is down. Displays the objects: ■ vcstackId ■ vcstackStkPortName.
vvcstackNbrMemberIdNotify	{ vcstackNotifications 8 }	The stack member id related to this trap.
vcstackStkPortNameNotify	{ vcstackNotifications 9 }	The stack port name related to this trap.

Table 47: Objects defined in AT-VCSTACK-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
vcstackOperationalStatus	{ vcstack 2 }	The operational status of the stack can be either: ■ enabled (1) ■ disabled (2).
vcstackMgmtVlanId	{ vcstack 3 }	The current stacking management VLAN ID.
vcstackMgmtVlanSubnetAddr	{ vcstack 4 }	The current stacking management VLAN subnet address.
vcstackTable	{ vcstack 5 }	Table of information about stack members, indexed by vcstackId.
vcstackEntry	{ vcstackTable 1 }	Information about a single stack member, indexed by vcstackId.
vcstackId	{ vcstackEntry 1 }	Stack member ID.
vcstackPendingId	{ vcstackEntry 2 }	Pending stack member ID.
vcstackMacAddr	{ vcstackEntry 3 }	Stack member's hardware MAC address.
vcstackPriority	{ vcstackEntry 4 }	Priority for election of the stack master. The lowest number has the highest priority.
vcstackRole	{ vcstackEntry 5 }	Stack member's role in the stack. Can be one of the following: ■ leaving (1) ■ discovering (2) ■ synchronizing (3) ■ backupMember (4) ■ pendingMaster (5) ■ disabledMaster (6) ■ fallbackMaster (7) ■ activeMaster (8).
vcstackLastRoleChange	{ vcstackEntry 6 }	Time and date when the stack member last changed its role in the stack.
vcstackHostname	{ vcstackEntry 7 }	Stack member's hostname.
vcstackProductType	{ vcstackEntry 8 }	Stack members product type.
vcstackSWVersionAutoSync	{ vcstackEntry 9 }	Whether or not the stack member's software is automatically upgraded.
vcstackFallbackConfigStatus	{ vcstackEntry 10 }	Status of the fallback configuration file. Can be one of: ■ fileExists (1) ■ fileNotFound (2) ■ notConfigured (3).

Table 47: Objects defined in AT-VCSTACK-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
vcstackFallbackConfigFilename	{ vcstackEntry 11 }	Filename of the fallback configuration file.
vcstackResiliencyLinkStatus	{ vcstackEntry 12 }	Status of the stack members resiliency link. Can be one of: ■ configured (1) ■ successful (2) ■ failed (3) ■ notConfigured (4).
vcstackResiliencyLinkInterfaceName	{ vcstackEntry 13 }	Name of the interface the resiliency link is configured on.
vcstackActiveStkHardware	{ vcstackEntry 14 }	Stack ports hardware type. Can be one of: ■ value (0) is now obsolete ■ xemStk (1) ■ builtinStackingPorts (2) ■ none (3) is now obsolete ■ stackXG (4).
vcstackStkPort1Status	{ vcstackEntry 15 }	Status of stack-port 1. Can be one of the following: ■ down (1) ■ neighbourIncompatible (2) ■ discoveringNeighbour (3) ■ learnedNeighbour (4).
vcstackStkPort1NeighbourId	{ vcstackEntry 16 }	ID of the neighbor on stack-port 1. Zero indicates no learned neighbor.
vcstackStkPort2Status	{ vcstackEntry 17 }	Status of stack-port 2. Can be one of: down (1) ■ neighbourIncompatible (2) ■ discoveringNeighbour (3) ■ learnedNeighbour (4).
vcstackStkPort2NeighbourId	{ vcstackEntry 18 }	ID of the neighbor on stack-port 2. Zero indicates no learned neighbor.
vcstackNumMembersJoined	{ vcstackEntry 19 }	Number of times the stack has acquired a member.
vcstackNumMembersLeft	{ vcstackEntry 20 }	Number of times the stack has lost a member.
vcstackNumIdConflict	{ vcstackEntry 21 }	Number of times that a stack member ID conflict has occurred.
vcstackNumMasterConflict	{ vcstackEntry 22 }	Number of times that a stack master conflict has occurred.
vcstackNumMasterFailover	{ vcstackEntry 23 }	Number of times that the stack master has failed.
vcstackNumStkPort1NbrIncompatible	{ vcstackEntry 24 }	Number of times that the neighbor on stack port 1 was incompatible.

Table 47: Objects defined in AT-VCSTACK-MIB (continued)

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
vcstackNumStkPort2NbrIncompatible	{ vcstackEntry 25 }	Number of times that the neighbor on stack port 2 was incompatible.
vcstackVirtualMacAddressStatus	{vcstack 7}	Indicates whether the virtual MAC address is enabled or disabled. Read-only object.
vcstackVirtualChassisId	{vcstack 8}	Displays the current virtual chassis ID. Read-only object.
vcstackVirtualMacAddr	{vcstack 9}	Displays the virtual MAC address used by the stack. Read-only object.
vcstackMasterId	{vcstack 10}	Displays the stack ID of the master unit, or the stack ID of the standalone unit. Read-only object

AT-VLAN-MIB

The atVlanStatistics-MIB Figure 8, and Table 48 defines objects for managing VLANs. The MIB contains a sub tree for managing VLAN statistics. Objects in the VLAN Statistics sub-tree have the object identifier **atvlaninfo** { sysinfo 16 } OID, 1.3.6.1.4.1.207.8.4.4.3.16. see "AT-SYSINFO-MIB" on page 121.

Figure 8: The atVlanStatistics MIB tree

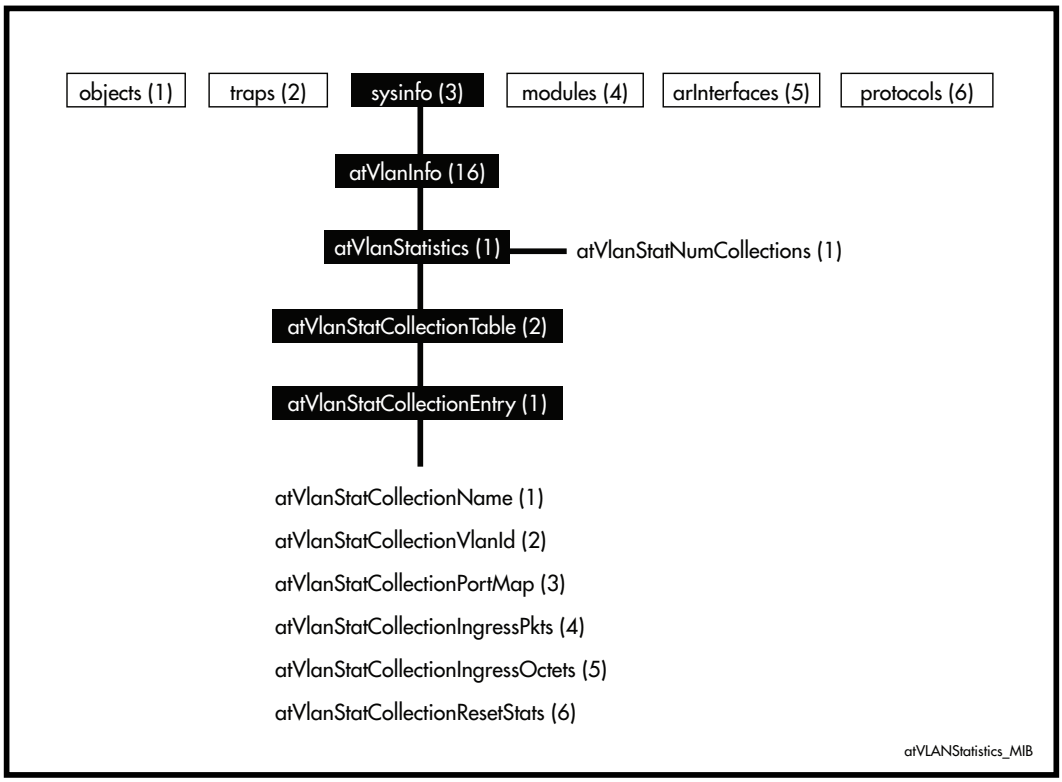


Table 48: Obejects defined in AT-VLANINFO-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
atvlaninfo	{ sysinfo 16 } 1.3.6.1.4.1.207.8.4.4.3.16.	Root of the Allied Telesis Enterprise MIB under the private(4) node defined in RFC1155-SMI.
atVlanStatistics	{ vlaninfo 1 }	The number of unique VLAN statistic gathering instances defined on the device.
atVlanStatNumCollections	{ atVlanStatistics 1 }	The number of unique VLAN statistic gathering instances defined on the device.
atVlanStatCollectionTable	{ atVlanStatistics 2 }	A table of VLAN statistic instances.
atVlanStatCollectionEntry	{ atVlanStatCollectionTable 1 }	Each entry represents a unique VLAN statistic gathering instance defined on the device.
atVlanStatCollectionName	{ atVlanStatCollectionEntry 1 }	The name of a VLAN statistics collection instance.
atVlanStatCollectionVlanId	{ atVlanStatCollectionEntry 2 }	The VLAN ID of ingress packets being monitored by this VLAN statistics collection instance.
atVlanStatCollectionPortMap	{ atVlanStatCollectionEntry 3 }	A bitwise port map indicating the switch ports being monitored by this VLAN statistics collection instance. The bit position within the string, maps to the port with the same index in dot1dBasePortTable in BRIDGE-MIB. A binary '1' indicates that the port is being monitored by this VLAN statistics collection instance, with a '0' indicating that it is not.
atVlanStatCollectionIngressPkts	{ atVlanStatCollectionEntry 4 }	The number of ingress packets received and counted by this VLAN statistics collection instance.
atVlanStatCollectionIngressOctets	{ atVlanStatCollectionEntry 5 }	The number of octets of data received from ingress packets counted by this VLAN statistics collection instance.
atVlanStatCollectionResetStats	{ atVlanStatCollectionEntry 6 }	When read, this object will always return 2 (false). Setting its value to 1 (true) will cause the matching VLAN statistics collection instance's ingress packets and ingress octet values to be reset to zero.

AT-XEM-MIB

AT-XEM-MIB defines objects for managing XEMs (Expansion Modules) installed in the device Table 49. Objects in this group have the object identifier **xem** { sysinfo 11 } OID, 1.3.6.1.4.1.207.8.4.4.3.11 and reside in the sysinfo Group of the Allied Telesis Enterprise MIB.

Table 49: Objects defined in AT-XEM-MIB

OBJECT	OBJECT IDENTIFIER	DESCRIPTION
xemTraps	{ xem 0 } 1.3.6.1.4.1.207.8.4.4.3.11.0	Collection of traps generated when a XEM is inserted or removed.
xemInserted	{ xemTraps 1 }	Trap generated when a XEM card is inserted to the device. It returns the objects xemInfoStackId and xemInfoBayId.
xemRemoved	{ xemTraps 2 }	Trap generated when a XEM card is removed from the device. It returns the objects xemInfoStackId and xemInfoBayId.
xemInsertedFail	{ xemTraps 3 }	Trap generated when the insertion of a XEM card into the device fails. It returns the objects xemInfoStackId and xemInfoBayId.
xemNumOfXem	{ xem 1 }	Total number of XEMs installed in the device.
xemInfoTable	{ xem 2 }	Table of information about the XEMs installed in the device, indexed by xemInfoMemberId and xemInfoBayId.
xemInfoEntry	{ xemInfoTable 1 }	Information about a single XEM.
xemInfoMemberId	{ xemInfoEntry 1 }	Stack Member ID in which the XEM is installed.
xemInfoBayId	{ xemInfoEntry 2 }	Number of the XEM bay in which the XEM is installed.
xemInfoXemId	{ xemInfoEntry 3 }	Numeric board identity of the XEM. This information is displayed in the ID column of the SHOW SYSTEM command.
xemInfoBoardType	{ xemInfoEntry 4 }	Board type. For XEMs this is always “Expansion”. This information is displayed in the Board column of the SHOW SYSTEM command.
xemInfoBoardName	{ xemInfoEntry 5 }	Name of the XEM. This information is displayed in the Board Name column of the SHOW SYSTEM command.
xemInfoRevision	{ xemInfoEntry 6 }	Board revision of the XEM. This information is displayed in the Rev column of the SHOW SYSTEM command.
xemInfoSerialNumber	{ xemInfoEntry 7 }	Serial number of the XEM. This information is displayed in the Serial Number column of the SHOW SYSTEM command.

Other Enterprise MIBs

In general, all objects are supported except where the relevant protocol or feature is either not supported or not applicable to the device. The following MIBs, although under the Enterprise Branch (OID 1.3.6.1.4.1) and utilized by AlliedWare Plus products, are not within the AlliedTelesis branch of the MIB object tree.

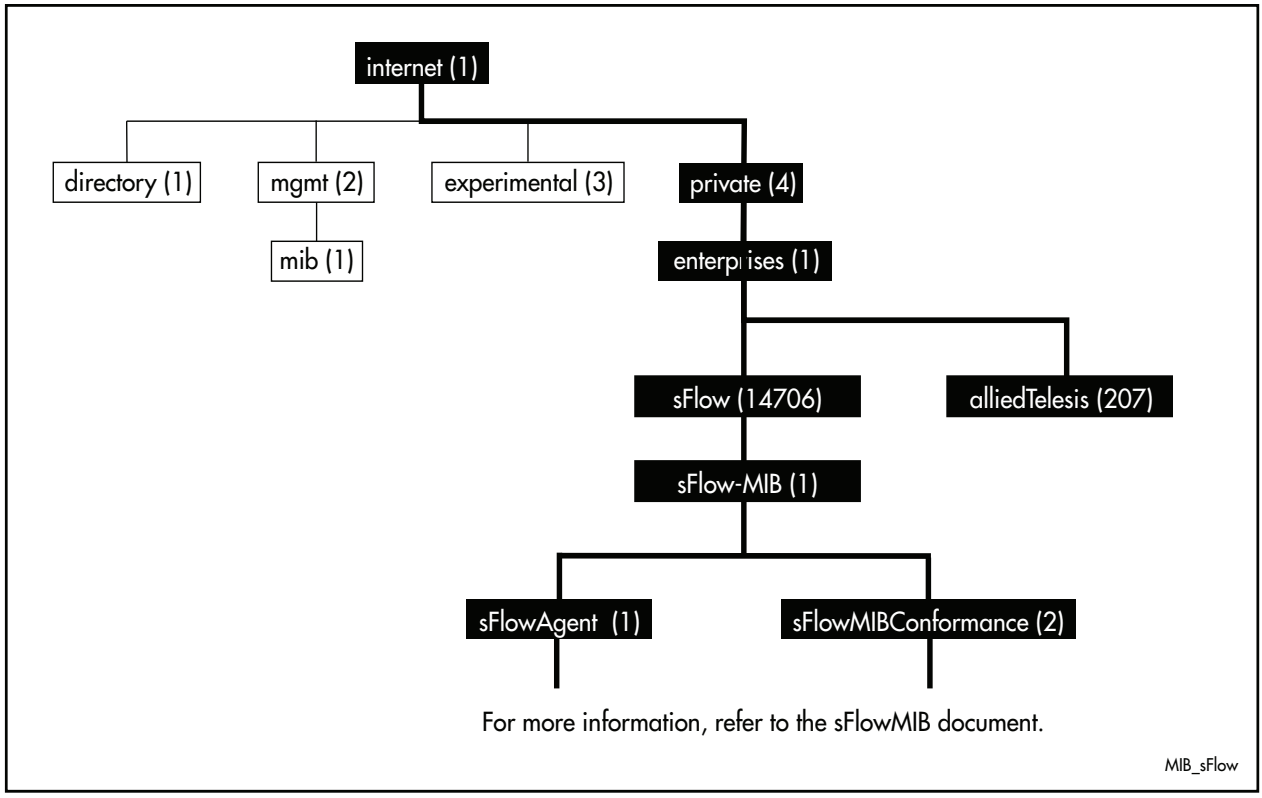
sFlow-MIB

The sFlow-MIB Figure 9, and Table 50 show references to objects for managing the generation and transportation of sFlow data records.

Table 50: Support for the sFlow-MIB

MIB NAME	REFERENCE / IMPLEMENTATION
sFlow-MIB	All MIB objects are fully supported. For more information, see www.sflow.org/SFLOW-MIB5.txt.

Figure 9: The sFlow Statistics MIB tree



Public MIBs

The following table lists the public MIBs supported by the AlliedWare Plus™ Operating System. In general, all objects are supported except where the relevant protocol or feature is either not supported or not applicable to the device. Any variations from the standard are listed.

Table 51: Public MIBs Supported by AlliedWare Plus™

MIB NAME	REFERENCE / IMPLEMENTATION
IANAifType-MIB	www.iana.org/assignments/ianaiftype-mib, IANAifType textual convention.
RFC1155-SMI	RFC 1155, Structure and Identification of Management Information for TCP/IP-based Internets.
-	RFC 1212, Concise MIB Definitions.
RFC1213-MIB	See IP-MIB.
-	RFC 1215, A Convention for Defining Traps for use with the SNMP.
-	RFC 1239, Reassignment of Experimental MIBs to Standard MIBs.
BGP4-MIB	RFC 1657, Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2.
IP-MIB	The IP MIB tree encompasses IP-MIB, RFC1213-MIB and IP-FORWARD-MIB definitions. The following documents define the components: ■ RFC 1213, ***Management Information Base for Network Management of TCP/IP-based internets: MIB-II*** ■ RFC 4292, ***IP Forwarding Table MIB*** ■ RFC 4293, ***Management Information Base for the Internet Protocol (IP).*** The following objects **are** supported: ■ ipForwarding ■ ipDefaultTTL ■ All ipAddrTable objects except ipAdEntReasmMaxSize ■ All ipNetToPhysicalTable objects except ipNetToPhysicalRowStatus (all read-only) ■ ipCidrRouteNumber ■ All ipCidrRouteTable objects except ipCidrRouteTos. All other objects in these MIBs are not supported. Note that an Enterprise version of ipAddressTable objects is provided by atIpAddressTable in AT-IP-MIB. This provides equivalent functionality along with support for primary and secondary IP addresses.
TCP-MIB	RFC 2012, SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2.
UDP-MIB	RFC 2013, SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2.
IP-FORWARD-MIB	See IP-MIB.
-	RFC 2257, Agent Extensibility (AgentX) Protocol Version 1.
PIM-MIB	RFC 2394, Protocol Independent Multicast MIB for IPv4. The following objects are deprecated: ■ pimNeighborMode in pimNeighborTable ■ pimRPTable ■ pimV1MIBCompliance ■ pimV1MIBGroup ■ The following objects are not supported: ■ pimCandidateRPTable.

Table 51: Public MIBs Supported by AlliedWare Plus™ (continued)

MIB NAME	REFERENCE / IMPLEMENTATION
SNMP-MPD-MIB	RFC 2572, Message Processing and Dispatching for the Simple Network Management Protocol (SNMP).
SNMP-COMMUNITY-MIB	RFC 2576, Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework.
SNMPv2-SMI	RFC 2578, Structure of Management Information Version 2 (SMIv2).
SNMPv2-TC	RFC 2579, Textual Conventions for SMIv2.
SNMPv2-CONF	RFC 2580, Conformance Statements for SMIv2.
P-BRIDGE-MIB	RFC 2674, Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions. The following objects are not supported: ■ dot1dTpPortOverflowTable ■ dot1dTrafficClassesEnabled ■ dot1dGmrpStatus ■ dot1dPortCapabilitiesTable ■ dot1dUserPriority ■ dot1dTrafficClassPriority ■ dot1dPortOutboundAccessPriorityTable ■ all objects in the dot1dGarp group ■ all objects in the dot1dGmrp group. The following read-write object is implemented as read-only: ■ dot1dPortNumTrafficClasses.
Q-BRIDGE-MIB	RFC 2674, Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions. The following objects are not supported: ■ dot1qGvrpStatus ■ dot1qFdbId ■ dot1qTpFdbAddress ■ dot1qTpGroupTable ■ dot1qForwardAllTable ■ dot1qForwardUnregisteredTable ■ all objects in the dot1qStatic group ■ dot1qVlanTimeMark ■ dot1qVlanIndex ■ dot1qVlanCurrentEgressPorts ■ dot1qVlanCurrentUntaggedPorts ■ dot1qVlanForbiddenEgressPorts ■ dot1qPortGvrpStatus ■ dot1qPortGvrpFailedRegistrations ■ dot1qPortGvrpLastPduOrigin ■ dot1qPortRestrictedVlanRegistration ■ dot1qPortVlanStatisticsTable ■ dot1qPortVlanHCStatisticsTable ■ dot1qLearningConstraintsTable. The following read-write objects are implemented as read-only: ■ dot1qPvid ■ dot1qPortAcceptableFrameTypes.

Table 51: Public MIBs Supported by AlliedWare Plus™ (continued)

MIB NAME	REFERENCE / IMPLEMENTATION
VRRPv3-MIB	RFC 6527, Definitions of Managed Objects for the Virtual Router Redundancy Protocol Version 3 (VRRPv3). All objects with read-write and read-create access are implemented as read-only. RFC 6527 (VRRPv3-MIB) obsoletes RFC 2787 (VRRP-MIB).
HOST-RESOURCES-MIB	RFC 2790, Host Resources MIB. The following objects are not supported: ■ hrStorageAllocationFailures ■ All objects in hrDevice ■ All objects in hrSWRun ■ All objects in hrSWRunPerf ■ All objects in hrSWInstalled ■ All objects in hrMIBAdminInfo.
SNMPv2-PDU	RFC 3416, Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP).
SNMPv2-TM	RFC 3417, Transport Mappings for the Simple Network Management Protocol (SNMP).
SNMPv2-MIB	RFC 3418, Management Information Base (MIB) for the Simple Network Management Protocol (SNMP).
POE-MIB	RFC 3621, Power Ethernet MIB. In each of the following objects, if one entry is set then all other entries for the same object in the table are set to the same value: ■ pethMainPseUsageThreshold ■ pethNotificationControlEnable. The following objects indicate PSE threshold usage notification: ■ pethMainPowerUsageOnNotification ■ pethMainPowerUsageOffNotification. The following read-write object is implemented as read-only: ■ pethPsePortPowerPairs.
EtherLike-MIB	RFC 3635, Definitions of Managed Objects for the Ethernet-like Interface Types. The following objects are deprecated: ■ dot3StatsEtherChipSet ■ all objects in the dot3Tests group ■ all objects in the dot3Errors group. The following read-write object is implemented as read-only: ■ dot3PauseAdminMode.

Table 51: Public MIBs Supported by AlliedWare Plus™ (continued)

MIB NAME	REFERENCE / IMPLEMENTATION
MAU-MIB	RFC 3636, Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs). The following objects are not supported: ■ all objects in the dot3RpMauBasicGroup group ■ ifMauTypeListBits ■ ifMauHCFalseCarriers ■ all object identifiers in the dot3MauType group ■ ifMauAutoNegCapabilityBits ■ ifMauAutoNegCapAdvertisedBits ■ ifMauAutoNegCapReceivedBits ■ ifMauAutoNegRemoteFaultAdvertised ■ ifMauAutoNegRemoteFaultReceived ■ all objects in the mauMod group. The following objects are deprecated: ■ ifMauTypeList ■ all objects in the dot3BroadMauBasicGroup group ■ ifMauAutoNegCapability ■ ifMauAutoNegCapAdvertised ■ ifMauAutoNegCapReceived. The following read-write object is implemented as read-only: ■ ifMauStatus.
INET-ADDRESS-MIB	RFC 4001, Textual Conventions for Internet Network Addresses.
BRIDGE-MIB	RFC 4188, Definitions of Managed Objects for Bridges. The following read-write objects are implemented as read-only: ■ dot1dStpPortEnable ■ dot1dStpPortPathCost. The following objects are not supported: ■ dot1dStaticTable ■ dot1dBaseDelayExceededDiscards ■ dot1dBasePortMtuExceededDiscards.
RSTP-MIB	RFC 4318, Definitions of Managed Objects for Bridges with Rapid Spanning Tree Protocol. The following read-write objects are implemented as read-only: ■ dot1dStpPortProtocolMigration ■ dot1dStpPortAdminEdgePort ■ dot1dStpPortAdminPointToPoint ■ dot1dStpPortAdminPathCost. The following object is deprecated: ■ dot1dStpPathCostDefault.
DISMAN-PING-MIB	RFC 4560, Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations. The following (lldpLocManAddrTable and lldpConfigManAddrTable) read-write object is implemented as read-only: ■ pingMaxConcurrentRequests. You can specify multiple ping operations, but the device only performs one ping at a time (pingMaxConcurrentRequests). The device uses ICMP echo for ping operations (pingImplementationTypeDomains).

Table 51: Public MIBs Supported by AlliedWare Plus™ (continued)

MIB NAME	REFERENCE / IMPLEMENTATION
LLDP-MIB	IEEE Standard 802.1AB-2005, Section 12, LLDP MIB Definitions. The following local management address table supports only a single management address per port: ■ lldpConfigManAddrTable.
LLDP-EXT-DOT1-MIB	IEEE Standard 802.1AB-2005, Annex F, IEEE 802.1 Organizationally Specific TLVs. Section F.7.1, IEEE 802.1LLDP extension MIB module. In each of the following tables, if one entry is set, all other entries in the table are set to the same value: ■ lldpXdot1ConfigVlanNameTxEnable ■ lldpXdot1ConfigProtoVlanTxEnable ■ lldpXdot1ConfigProtocolTxEnable.
LLDP-EXT-DOT3-MIB	IEEE Standard 802.1AB-2005, Annex G, IEEE 802.3 Organizationally Specific TLVs, Section G.7.1, IEEE 802.3 LLDP extension MIB module.
LLDP-EXT-MED-MIB	ANSI/TIA-1057- 2006, Section 13.3, LLDP-MED MIB Definition.
RIPv2-MIB	RFC1724 - RIP Version 2 MIB Extension.
ENTITY-MIB	RFC 6933 Entity MIB Version 4. Provides inventory information about the system as well as the physical ports that are present. The support of this MIB includes implementation of the Entity Physical Table. This table includes entries for boards, fixed ports, pluggable ports and sensors. Each entry includes the objects Name and Description. Pluggable sensors are not included in this table. SET operation is not supported. The Entity MIB Trap definition (entConfigChange) notification is supported. The following read-write objects in entPhysicalTable are implemented as read-only: ■ entPhysicalAlias ■ entPhysicalAssetID ■ entPhysicalSerialNum ■ entPhysicalUris Objects in the entLogicalTable are not supported.
ENTITY-SENSOR-MIB	RFC 3433 Entity Sensor MIB. Provides information about the sensors that are present in the system and includes entries for all board and bay sensors in the system. Every sensor entry includes information Type, Scale, Units and Values. Pluggable sensors are not included in this table.
ENTITY-STATE-MIB	RFC 4268 Entity State MIB. Provides information on the possible state attributes that could be tracked for a given entity. The implementation includes support for the Entity State Table. The table includes state information for all boards, ports, pluggables and sensor entities listed in the Entity Physical Table. The following read-write objects in the entStateTable are implemented as read-only: ■ entStateAdmin