SNMP Supplement

Ascend Communications

Part Number 7820-0402-002 January 17, 1996

Ascend Customer Service

When you contact Ascend Customer Service, make sure you have this information:

- The product name and model
- The software and hardware options
- The software version
- The SPIDs (Service Profile Identifiers) associated with your product
- The type of telco switch and mode, such as AT&T 5ESS Custom or Northern Telecom DMS-100 National ISDN 1
- Whether you are routing or bridging with your Ascend product
- The type of computer you are using
- A description of the problem

How to contact Ascend Customer Service

Ways to contact Ascend Customer Service	Telephone number or address
Telephone in the United States	800-ASCEND-4 (800-272-3634)
Telephone outside the United States	510-769-8027
E-mail	support@ascend.com
Facsimile (FAX)	510-814-2300

You can also contact the Ascend main office by dialing 510-769-6001, or you can write to Ascend at the following address:

Ascend Communications
1275 Harbor Bay Parkway
Alameda, CA 94502

Need information on new features and products?

We are committed to constantly improving our products. You can find out about new features and product improvement as follows:

- For the latest information on the Ascend product line, visit our site on the World Wide Web:

  `http://www.ascend.com/`

- For software upgrades, release notes, and addenda to this manual, visit our FTP site:

  `ftp.ascend.com`

Table of Contents

About This Supplement

1 Introduction to SNMP

2 Configuring Ascend Units for SNMP

3 Ascend Enterprise Traps

4 Ascend Enterprise MIB

5 Industry Standard MIB Implementations

About This Supplement

This supplement describes the SNMP (Simple Network Management Protocol) Version 1 implementation for the Ascend products that support SNMP. The implementation is for SNMP over TCP/IP. It includes the Ascend Enterprise MIB (Management Information Base) and Enterprise Traps, which have been registered with the IANA (Internet Assigned Numbers Authority). It also documents related MIB-II objects and groups.

What is in this supplement?

This Supplement contains these chapters:

- Chapter 1, "Introduction to SNMP," gives you a brief overview of SNMP.
- Chapter 2, "Configuring Ascend Units for SNMP," explains how to set SNMP-related parameters in the Ascend unit menus.
- Chapter 3, "Ascend Enterprise Traps," lists and explains the three trap classifications and individual traps within those classifications.
- Chapter 4, "Ascend Enterprise MIB," describes the objects in the Ascend Enterprise MIB.
- Chapter 5, "Industry Standard MIB Implementations," describes related MIB implementations defined in various RFCs.

What you should know

This Supplement is intended for the person who will use SNMP to access management information from the Ascend unit.

You need to understand IP networking and routing to some extent. If you are not familiar with IP networks, you should first read the chapter on Configuring the MAX as an IP router in the *MAX ISP and Telecommuting Configuration Guide* for background information.

Documentation conventions

This section shows the documentation conventions used in this guide.

Convention	Meaning
`Monospace text`	Monospace text represents information that you enter exactly as shown, and it identifies onscreen text, such as, statistical information.
italics	Italics represent variable information. Do not enter the words themselves in the command; enter the information they represent.

Convention	Meaning
Note:	A note signifies important additional information.
Caution:	A caution means that a failure to follow the recommended procedure could result in a loss of data or damage to equipment.
Warning:	A warning means that a failure to take appropriate safety precautions could result in physical injury.

Related publications

This guide does not provide a detailed explanation of products, architectures, or standards developed by other companies or organizations. To find the information you need, refer to these documents:

- RFC 1155, "Structure and Identification of Management Information for TCP/IP-based Internets," May 1990.
- RFC 1158, "Management Information Base for Network Management of TCP/IP-based Internets," March 1991.
- RFC 1212, "Concise MIB Definitions," March 1991.
- RFC 1213, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II," March 1991.
- RFC 1215, "A Convention for Defining Traps for use with the SNMP," March 1991.
- RFC 1315, "Management Information Base for Frame Relay DTEs," April 1992
- RFC 1317, "Definitions of Managed Objects for RS-232-like Hardware Devices," April 1992
- RFC 1369, "Implementation Notes and Experience for The Internet Ethernet MIB," October 1992.
- RFC 1398, "Definitions of Managed Objects for the Ethernet-like Interface Types," January 1993.
- RFC 1406, "Definitions of Managed Objects for the DS1 and E1 Interface Types," January 1993.
- RFC 1600, "Internet Official Protocol Standards," March 1994. [This RFC is updated periodically by the Internet Architecture Board.]
- RFC 1696, "Modem Management Information Base (MIB) using SMIv2," August 1994.
- Ascend Communications, Enterprises 529, "SNMPv1 based on RFC 1696"

Note: Copies of these RFCs can be obtained via anonymous ftp at ds.internic.net. Anonymous ftp uses the name "anonymous" during login on the ftp server.

You can download the Ascend Enterprise MIB by anonymous ftp from ftp.ascend.com. Log in with the user name anonymous and no password is required.

Introduction to SNMP

1

This chapter covers these topics:

What is SNMP?

The Simple Network Management Protocol (SNMP) is an Internet-recommended standard for network management.In Ascend units, SNMP requires Ethernet as the network medium and TCP/IP as the transport protocol.

An SNMP management system consists of one or more SNMP managers and SNMP agents.

An SNMP manager is a utility that runs on a network host. It displays and tracks information gathered from SNMP agents across the network. SNMP managers often display information graphically (using maps or graphs), track statistics, and sound alarms for certain kinds of events.

SNMP agents send "trap" messages when certain kinds of conditions occur, and respond to SNMP manager queries. SNMP agents can be managed remotely via SNMP commands such as GET and SET.

Note: For information about the Ethernet interface, see the *MAX ISP and Telecommuting Configuration Guide.*

What is a MIB?

A MIB (Management Information Base) is a database of objects that represent network or device management information. MIB objects include network protocol information, routes, and other objects related to how a particular kind of device functions. Both the SNMP manager and the SNMP agent must use the same MIB to exchange information.

There are two types of MIB: Internet standard MIBs, which track information common to most networks, and private Enterprise MIBs, which track information specific to a particular kind of device, such as the MAX or Pipeline. This document describes both the Ascend Enterprise MIB and the MIB II implementations described in various RFCs.

What you need before you start

To configure an Ascend unit as an SNMP agent, you need these items:

- An IP host that can run an SNMP manager
- The Ascend Enterprise MIB
- Connection to the Ascend unit
- An optional Ethernet Interface Card, if you have a standard MAX

Where do you go from here?

Now you are ready to configure the Ascend unit for your environment, continue with the next chapter.

Configuring Ascend Units for SNMP

2

This chapter contains:

Configuring for SNMP operation

Ascend units that support SNMP can be configured to communicate traps to and accept commands from an SNMP manager. Using SNMP commands, the SNMP manager can control operations such as dialing and hanging up, and monitor the unit's operational status.

The Ascend Enterprise MIB is accessible via FTP on Ascend's FTP server. You must compile the Ascend Enterprise MIB into a format acceptable to your SNMP manager's program. Remember to turn on the traps options during the compile, if you intend to use the Ascend Enterprise Traps.

If you are using Sun NetManager, bringing the MIB into the manager is called the Enterprise MIB to Schema process. If you have HP OpenView, you must load the Ascend Enterprise MIB into the MIB database. If you have HP OpenView, also remember to load the Event Configuration file.

Note: The Ascend unit must be able to communicate via TCP/IP with the host running the SNMP manager. Make sure that the Ascend unit can locate that host, either by enabling RIP on the Ethernet interface or by configuring a static route.

Enabling SNMP traps in the Ascend unit

Ascend units generate traps for important events. The events are classified as follows:

- Alarm events
- Port state change events
- Security events

For details on each classification, see Chapter 3, "Ascend Enterprise Traps.".

A trap is a mechanism for reporting system change in real time, for example, reporting an incoming call to a serial host port. To report system changes, a traps-PDU (protocol data unit) is sent across the Ethernet interface to the SNMP manager. A brief summary of the system changes which cause a traps-PDU follows. (A complete list specifying these events is available in Chapter 3, "Ascend Enterprise Traps.")

- change of state of a serial host port including changes resulting from a call state change
- cold start (see RFC 1215)
- link up (at the WAN or LAN interface, see RFC 1215)
- link down (at the WAN or LAN interface, see RFC 1215)
- authentication error (see RFC 1215)
- TELNET login attempts exceeded
- DLCI Status change (see RFC 1315)

To configure the destinations of the traps-PDUs sent by MAX to the SNMP manager, define an SNMP Traps Profile. To do so, open the SNMP Traps menu below the Ethernet menu. You will see eight numbers, each of which can be defined as an SNMP Traps Profile.

```
                    Edit
90-700 SNMP Traps
>90-701
 90-702
 90-703
 90-704
 90-705
 90-706
 90-707
 90-708
```

Press Enter to open a profile.

```
                    Edit
90-701
>Name=
 Alarm=Yes
 Port=No
 Security=No
 Comm=
 Dest=0.0.0.0
```

For details on each of the parameters you can set in an SNMP Traps Profile, see the *MAX Reference Guide.*

- Name

  This parameter identifies the SNMP Traps Profile and usually is set to the destination of the trap PDUs. The Name can be up to 31 characters.

- Alarm

  Set Alarm to Yes to enable traps for Alarm events.

- Port

  Set Port to Yes to enable traps for Port change state events.

- Security

  Set Security to Yes to enable traps for Security events.

- Comm

  Comm is equivalent to the SNMP "community name," which becomes a password sent to the SNMP management station when an SNMP trap event occurs. It authenticates the sender who is identified by the source IP address.

  The Comm name can be up to 31 characters.

  To turn off SNMP traps, delete the value for this parameter and set the next parameter (Dest) to 0.0.0.0.

- Dest

  Dest establishes the destination address of the trap-status report. Use IP dotted decimal format. Its default value is 0.0.0.0. To turn off SNMP traps, set Dest=0.0.0.0 and delete the value for Comm.

Password protecting SNMP SET commands

To password-protect SNMP events, so that only an authorized SNMP management utility can obtain the trapped information, open the SNMP Options submenu of the Ethernet Profile.

```
Edit
90-900 Mod Config
 SNMP options...
 >Read Comm=*SECURE*
  R/W Comm=*SECURE*
```

The parameters in this submenu determine whether the SNMP user (utility) may access only GET commands or both GET and SET commands.

Specifying a local location and contact

SNMP management utilities can read the Location and Contact fields in the Sys Config Profile. To provide information about where the Ascend unit is located physically, and who at your site to contact in error conditions, open the Sys Config Profile below the System menu.

```
Edit
00-100 Sys Config
>Name=                         ^
 Location=
 Contact=
 Date=0/0/0
 Time=00:00:00
 Term Rate=9600
 Console=Standard
 Remote Mgmt=Yes
 Parallel Dial=5
 Single Answer=Yes
 Sub-Adr=None
 Serial=0
 LAN=0
 DM=0
 V.110=0
 Use Trnk Grps=No
 Excl Routing=Yes              v
```

In the Location and Contact fields, specify where the MAX is physically located and the name of someone who should be contacted if an error condition occurs. You can enter up to 38 characters in each field.

Ascend Enterprise Traps

3

This chapter covers these topics:

For information about enabling traps in the Ascend unit, see "Configuring Ascend Units for SNMP" on page 2-1.

Alarm events

Alarm events (also called "error events") use trap types defined in RFC 1215 and 1315, as well as an Ascend Enterprise trap type. The following trap types from RFC 1215 are supported:

- coldStart (RFC-1215 trap-type 0)

  A coldStart trap signifies that the Ascend unit sending the trap is reinitializing itself so that the configuration of the SNMP manager or the unit might be altered.

- warmStart (RFC-1215 trap-type 1)

  A warmStart trap signifies that the Ascend unit sending the trap is reinitializing itself so that neither the configuration of SNMP manager or the unit is altered.

- linkDown (RFC-1215 trap-type 2)

  A linkDown trap signifies that the Ascend unit sending the trap recognizes a failure in one of the communication links represented in the SNMP manager's configuration.

- linkUp (RFC-1215 trap-type 3)

  A linkUp trap signifies that the Ascend unit sending the trap recognizes that one of the communication links represented in the SNMP manager's configuration has come up.

- frDLCIStatusChange (RFC-1315 trap-type 1)

  A DLCIStatusChange trap signifies that the Ascend unit sending the trap recognizes that one of the virtual circuits (to which a DLCI number has been assigned) has changed state; that is, the link has either been created, invalidated, or it has toggled between the active and inactive states.

- eventTableOverwrite (ascend trap-type 16)

  A new event has overwritten an unread event. This trap is sent only for systems that support Ascend's accounting MIB. Once sent, additional overwrites will not cause another trap to be sent until at least one table's worth of new events have occurred.

Port state change events

This section lists the trap types reported when a serial host port changes state. MAX and Multiband Plus have serial host ports, while Pipeline has no serial host ports.

Serial host port changes are usually generated by a change in the call state at that port. Serial host port changes are independent of port type, and apply to any serial interface, including V.35/RS-449/X.21 signaling, linkup and linkdown. Serial host port changes do not include changes to LAN connections such as Ethernet.

These traps are effective on a port-by-port basis for each port pointed to by ifIndex. The hostPort objects are used to associate a change with ifIndex objects.

- portInactive (ascend trap-type 0)

  Serial host port associated with the passed index has become inactive.

- portDualDelay (ascend trap-type 1)

  Serial host port associated with the passed index is delaying the dialing of a second to avoid overloading devices that cannot handle two calls in close succession (see the Delay Dual parameter in the *MAX Reference Guide*).

- portWaitSerial (ascend trap-type 2)

  Serial host port associated with the passed index has detected DTR and is waiting for an HDLC controller to come online. CTS is off (V.25 bis dialing only).

- portHaveSerial (ascend trap-type 3)

  Serial host port associated with the passed index is waiting for V.25 bis commands. CTS is on.

- portRinging (ascend trap-type 4)

  Serial host port associated with the passed index has been notified of an incoming call.

- portCollectDigits (ascend trap-type 5)

  Serial host port associated with the passed index is receiving digits from an RS366 interface (RS-366 dialing only).

- portWaiting (ascend trap-type 6)

  Serial host port associated with the passed index is waiting for connect notification from the WAN after dialing or answer notification has been issued.

- portConnected (ascend trap-type 7)

  Serial host port associated with the passed index has changed state. This change of state can be from connected to unconnected or vice versa. If connected to the far end, end-to-end data can flow but has not yet been enabled.

  The following trap report sequence shows a link is up:

  portWaiting (6)

  portConnected (7)

  portCarrier (8)

  The following trap report sequence shows a link is down:

  portConnected (7)

  portInactive (0)

- portCarrier (ascend trap-type 8)

  Serial host port associated with the passed index has end-to-end data flow enabled.

- portLoopback (ascend trap-type 9)

  Serial host port associated with the passed index has been placed in local loopback mode (see the Local LB parameter in the *MAX Reference Guide*).

- portAcrPending (ascend trap-type 10)

  Serial host port associated with the passed index has set ACR on the RS366 interface, and is waiting for the host device (RS-366 dialing only).

- portDTENotReady (ascend trap-type 11)

  Serial host port associated with the passed index is waiting for DTE to signal a ready condition when performing X.21 dialing.

Security events

Security events are used to notify users of security problems and track access to the unit from the console. The mib-II event authenticationError is a security event. The other security events are Ascend-specific.

- authenticationFailure (RFC-1215 trap-type 4)

  An authenticationFailure trap signifies that the Ascend unit sending the trap is the addressee of a protocol message that is not properly authenticated.
- consoleStateChange (ascend trap-type 12)

  The console associated with the passed console index has changed state. To read the console's state get ConsoleEntry from the Enterprise MIB.
- portUseExceeded (ascend trap-type 13)

  The serial host port's use exceeds maximum set by Max DS0 Mins Port Profile parameter associated with the passed index (namely, the interface number).
- systemUseExceeded (ascend trap-type 14)

  The serial host port's use exceeds maximum set by Max DS0 Mins System Profile parameter associated with the passed index (namely, the interface number).
- maxTelnetAttempts (ascend trap-type 15)

  There have been three consecutive failed attempts to login onto this Ascend unit via TELNET.

Ascend Enterprise MIB

4

This chapter covers these topics:

Ascend MIB definitions

The Enterprise MIB is registered with the IANA (Internet Assigned Numbers Authority) as:

```
enterprises 529
```

with this value:

```
1.3.6.1.4.1.529
```

Ascend interface numbering

The Ascend MAX products have up to eight slots, each of which is associated with a functional module. In the MAX products (except for the MAX 1800), six of the eight modules are removable cards and two are fixed in the motherboard. Each physical interface on a module is called a port. The number and type of the ports depends on the type of module. For example, serial host modules have a port for each synchronous serial interface, and WAN modules have a port for each T1, E1, BRI, or switched-56 line.

The maximum slot number for each product family is shown below.

- 3 (Pipeline product family)
- 3 (Multiband product family)
- 8 (MAX Product family except MAX 1800)
- 4 (MAX 1800)

MIB-2 interface numbering and the Enterprise MIB

MIB-2 interfaces are indexed by a single pointer, ifIndex, while the Enterprise MIB indexes these same interfaces by slot and item (lines, serial ports, bridge/routing sessions) or by the separate console indices. One of the purposes of the Enterprise MIB is to map between MIB-2 interface number and the slots and item index. This is done by assuming the following relationships:

- Every slot contains 1 or more items.
- Every item may be referred to by zero or more interface numbers.
- An item is the addressable entity that exists on a slot. The enumerated items on the various slots.
  - Serial Host Slots (Each serial host port is an item.)
  - Serial WAN Slots (Each serial port that interfaces to the WAN is an item.)
  - Network Slots (Each T1, E1, BRI, or switched-56 line that interfaces to the WAN is an item.)
  - BRI Host Slots (Each BRI line that interfaces to a TE/TA is an item.)
  - LAN Slots (The Ethernet interface, the software loopback, and each concurrent bridging/routing session supported by the entity are items.) Note: An online bridge/router session interfaces to a link across the WAN.

Groups implemented in the Ascend Enterprise MIB

The Enterprise MIB follows the RFC 1155 specification and consists of six object groups. Each of these groups is given an object identifier. For example, the previous section gives the

Enterprise MIB an object identifier of 529, and since the object identifier of the slots group within the Enterprise MIB is 2, the value of the slots would be 1.3.6.1.4.1.529.2.

Table 4-1. Implemented groups

Value	Object Identifier	Reference
products	ascend 1	products group
slots	ascend 2	slots group
hostTypes	ascend 3	hostTypes group
wanTypes	ascend 4	wanTypes group
lanTypes	ascend 5	lanTypes group
doGroup	ascend 6	doGroup group
hostStatus	ascend 7	hostStatus group
console	ascend 8	console group
systemStatusGroup	ascend 9	system status group
eventGroup	ascend 10	event group
callStatusGroup	ascend 11	call status group
sessionStatusGroup	ascend 12	session status group
radiusGroup	ascend 13	radius group

Ascend products group (ascend 1)

The following are defined object identifiers in the Products Group of the Enterprise MIB. When this document refers to Ascend products, it means the product families listed in this group. For example, the previous sections give the Enterprise MIB an object identifier of 529 and the Products Group identifier is 1, and since the object identifier of Pipeline within the Products Group is 3, the value of the Pipeline product family would be 1.3.6.1.4.1.529.1.3.

- Multiband (products 1)
- MAX (products 2)
- Pipeline (products 3)

Note: The Pipeline 25 does not support SNMP.

Ascend slots group (ascend 2)

The slot group (ascend 2) contains the MIB variables used to determine the number of slots supported by the device and the type of slot card currently installed.

The slot table

Ascend products (defined in the Products Group) contain a card in each slot. In the Multiband and Pipeline product families, cards are fixed: that is, cards are not physically separable from the motherboard.

The MAX has both fixed and removable cards. It determines the functionality of a removable card at start-up time. The MAXslotIndex 1 and 2 hold fixed cards, while 3 through 8 hold removable cards or are empty. The MAX slotIndex 1, 2, 9, A, and B hold fixed cards, while 3 through 8 hold removable cards or are empty.

The slot table contains slotNumber entries describing each slot. The slot table and its entries are defined as follows:

slotNumber (slots 1) read-only

An integer that represents the number of slots, fixed or removable, supported by this device.

slotTable (slots 2) not-accessible

A list of slot entries. The number of entries is the value of slotNumber.

slotEntry (slotTable 1) not-accessible

An entry containing objects to describe the slot.

slotIndex (slotEntry 1) read-only

An integer between 1 and slotNumber.

slotName (slotEntry 2) read-only

The slot name is the ASCII representation of the name of the card in the indexed slot. The name is the same name displayed on the Main Edit Menu less the menu number. Current returned values are as follows:

Table 4-2. Slot names

Value	Description
Empty	No card present.
Not Avail	Card failed POST (pre-operational self test).
Unknown	Card type not recognized or assigned.
Net/T1	DS1 (T1) interface.
Net/E1	E1 interface.
Net/DPNSS	DPNSS (E1) interface. Only returned on E1 units running DPNSS software.
Net/BRI	basic rate interface. (TE; that is, terminates lines from the WAN)
Net/S56-4	Four-wire switched 56 interface.
Net/S56-2	Two-wire switched 56 interface.
Host/Dual	Two-port serial host (V.35/RS-449) inverse multiplexing interface.

Table 4-2. Slot names (continued)

Value	Description
Host/AIM6	Six-port serial host (V.35/RS-449) inverse multiplexing interface.
Host/Quad	Four-port serial host (V.35/RS-449). Only the first two ports are inverse multiplexing capable.
Host/BRI	basic rate interface (NT; that is, provides BRI lines to TE devices).
Ethernet	Ethernet support for packet switched data connections.
Ether Data	Extra support for packet switched data connections. Ether-Data card also called HDLC (High-Level Data Link Control) card.
Lan Modem	Digital modem interface.
Serial WAN	Serial WAN interface.

Note: Net/T1, Net/E1, Net/BRI, Net/S56-2, Net/S56-4, and Serial WAN are all DTE-type interfaces, and as such they terminate digital subscriber lines from the WAN. The number of lines terminated at a slot depends on equipment options

slotType (slotEntry 3) read-only

This field returns a string that indicates the type of card in the indexed slot. The information returned contains slightly more meaning than the slotName field and is more amenable to processing by a network management station. (See "slotName (slotEntry 2) read-only" on page 4-4.) The values returned are as follows:

Table 4-3. Slot types

Value	Description
other (1)	none of the following
empty (2)	no card present (empty slot)
sysT1 (3)	Fixed T1 card (see Net/T1, slotName)
slotT1 (4)	Removable T1 card (see Net/T1, slotName)
sysE1 (5)	Fixed E1 card (see Net/E1, slotName)
slotE1 (6)	Removable E1 card (see Net/E1, slotName)
bri (7)	Fixed BRI card in TE (talks to network) configuration (see Net/BRI, slotName)
s56-2 (8)	Fixed two-wire switched 56 card (see Net/S56-2, slotName)
s56-4 (9)	Fixed four-wire switched 56 card (see Net/S56-4, slotName)
dualHost (10)	Fixed two-port serial host card (see Host/Dual, slotName)
quadHost (11)	Fixed four-port serial host card (see Host/Quad, slotName)
aim2 (12)	Removable two-port serial host card (see Host/Dual, slotName)

Table 4-3. Slot types (continued)

Value	Description
aim6 (13)	Removable six-port serial host card (see Host/6, slotName)
ethernet (14)	Ethernet card (see Ethernet, slotName)
ethernetData (15)	Ether -Data card, also called HDLC Module (see Ether Data, slotName)
slotBriTE (16)	Removable BRI card in TE (talks to network) configuration (see Net/BRI, slotName)
slotBriNT (17)	Removable BRI card in NT (talks to user stations) configuration (see Host/BRI, slotName)
lanModem (18)	Removable LAN Modem card, also called Digital Modem (see Lan modem, slotName)
serialWan (19)	Fixed Serial WAN card

Note: See slotItems for the number of interfaces in each slot. For example: a sysT1 slot card has two T1/PRI ports and therefore has 2 items; a quadHost slot card has four V.35/Rs-449/X.21 ports and therefore has 4 items; an ethernetData slot card supports 32 bridge/routing sessions and therefore has 32 items; however the Ethernet slot card on the standard MAX has 35 items (32 active bridge/routing sessions, 1 idle bridge/routing session, 1 software loopback, and 1 Ethernet port).

slotFixed (slotEntry 4) read-only

This value indicates if the card in the indexed slot is fixed(1) or removable(2).

slotItems (slotEntry 5) read-only

An “item” is a port for host slots or a line for network slots. Each slot contains 1 or more items that map to an interface number. The slot item table defines indicates the number of items on the indexed slot card. Each of following is counted as a slotItem:

- every port to a WAN line (Net/BRI, T1/PRI, or Switched-56)
- every port to a local line (Host/BRI)
- every serial host port (V.35, RS-449, X.21)
- every serial WAN port (Serial WAN)
- the Ethernet port
- every concurrent bridge/routing session supported

Console ports are not counted as items on indexed slot cards. Console ports are counted as items of the whole system, not of any one card. See “Ascend console group (ascend 8)” on page 4-17 for details.

slotSpecific (slotEntry 6) read-only

The slotSpecific variable references MIB definitions specific to the hardware in this slot. It contains the object identifier {0 0} if there is no information for the current slot type.

The slot item table

The slot item table is list of slot item entries, each of which contains the starting interface and number of interfaces used by the indexed slot/item.

The slot item table and its entries are defined as follows:

slotItemTable (slots 3) not-accessible

A lit of slot item entries.

slotItemEntry (slotItemTable 1) not-accessible

An entry containing the starting interface and number of interfaces used by the indexed slot/item.

slotItemSlotIndex (slotItemEntry 1) read-only

A unique value for each slot.

slotItemIndex (slotItemEntry 2) read-only

A unique value for each item in each slot.

slotItemFirstIf (slotItemEntry 3) read-only

The interface number associated with the indexed item on the indexed slot. May be set to 0 if the slot/item is not associated with any interface.

Note that slotItemFirstIf gives the first MIB-2 interface number (ifIndex) assigned to the interface pointed to by the slot/item indices. Currently only one interface corresponds to each slot/item entry. Therefore, this table can be used to translate a slot and item to an interface number.

slotItemIfCount (slotItemEntry 4) read-only

The number of interfaces associated with the indexed item on the indexed slot. Typically set to 1 for host slots and the number of channels on network (WAN) slots. May be set to 0 when the item is not associated with an interface.

slotItemSpecific (slotItemEntry 5) read-only

A reference to MIB definitions specific to the indexed item on the indexed slot. Contains the object identifier {0 0} if there is no item specific information for the indexed item.

For example, suppose a user wants to interrogate the speed of the second serial host port on the card in slot 3. Port speed is part of the RS232 MIB which is indexed by interface number. The sequence of events to find the desired data is:

To get the interface number corresponding to slot 3, port 2, get the following:

GET ascend.slots.slotItemTable.slotItemEntry.slotItemFirstIf.3.2

Note: Some SNMP managers cannot get individual items but get whole tables. In such a case, you would dump the slotItemTable and look for the slotItemFirstIf object corresponding to slot 3 (slotItemSlotIndex) and port 2 (slotItemIndex).

Use the interface number to query the port speed from the RS232 MIB using the following query:

GET rs232.rs232PortTable.rs232PortEntry.rs232PortInSpeed.*if*

where if is the interface number retrieved from the Ascend slotItemTable.

The following table is an example of an SNMP dump. In this dump, 1.1 and 1.2 are T1 lines #1 and #2 on a MAX motherboard. Slot.items 2.1 and 2.2 are the V.35/RS-449/X.21 ports on the motherboard. Slot 8 has a Net/BRI card. Slot 9 is the Ethernet slot.

Table 4-4. SNMP dump showing slot items

KEY	slotItem Slot Index	slotItem Index	slotItem FirstIf	slotItemIf Count	slotItem Specific
1.1	1	1	77	1	0.0
1.2	1	2	78	1	0.0
2.1	2	1	79	1	0.0
2.2	2	2	80	1	0.0
8.1	8	1	84	1	0.0
8.2	8	2	85	1	0.0
8.3	8	3	86	1	0.0

Table 4-4. SNMP dump showing slot items

KEY	slotItem Slot Index	slotItem Index	slotItem FirstIf	slotItemIf Count	slotItem Specific
8.4	8	4	87	1	0.0
8.5	8	5	88	1	0.0
8.6	8	6	89	1	0.0
8.7	8	7	90	1	0.0
8.8	8	8	91	1	0.0
9.1	9	1	1	1	0.0
9.2	9	2	2	1	0.0
9.3	9	3	3	1	0.0
9.4-9.74	etc.	etc.	etc.	etc.	etc.
9.75	9	75	75	1	0.0

The slot interface table

The slot interface table maps interfaces to the associated slot and item. This table contains one entry per interface and the contents of the table are the slot and item associated with the index. Items not related to a specific slot, for example, console ports, are also mapped here.

Note: Multiple interfaces can point to the same slot/item.

The slot interface table is defined as follows:

slotIfTable (slots 4) not-accessible

A lit of slot interface entries.

slotIfEntry (slotIfTable 1) not-accessible

An entry for a slot interface.

slotIfIndex (slotIfEntry 1) read-only

The interface index, ranging from 1 to the number of interfaces specified in the MIB-II variable ifNumber. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

For every interface in slotIfTable, the value of slotIfIndex and IfIndex (in MIB-2) is identical. Therefore, using the slotIfTable, you can look up the slot and item corresponding to ifIndex values. The following objects in slotIfTable return slot and item values corresponding to an interface value:

slotIfSlotIndex (slotIfEntry 2) read-only

The slot index, ranging from 0 to the number of slots specified in slotNumber. When zero the interface references an item in the console group of this MIB. Otherwise the slot identified by a particular value of this index is the same slot as identified by the same value of slotIndex.

slotIfItemIndex (slotIfEntry 3) read-only

The item index, ranging from 1 to the number of items supported on the slot indicated by slotIfSlotIndex. When slotIfSlotIndex is non-zero the number of items supported is specified in slotItems for the slot entry index by slotIfSlotIndex. The item identified by a particular value of this index is the same item as identified by the same value of slotItemIndex.

For example, assume a MAX has been configured to send traps to a network management station when network alarms occur. The trap sent is the MIB-2 linkDown trap which contains the interface number 6. The question is: what line went down?

The network management station gets the values of the slot and item as follows. Slot 3, item 2 are returned by these GET commands

GET ascend.slots. slotIfTable. slotIfEntry. slotIfSlotIndex.6 3

GET ascend.slots. slotIfTable. slotIfEntry. slotIfItemIndex.6 2

A final step is to check the slotType object in the slotTable indexed by the slot number returned for slotIfSlotIndex. This tells the management station what kind of line went down. In this example the value 4 indicates a T1 module.

GET ascend.slots. slotTable. slotEntry. slotType i.3 4

Ascend host types group (ascend 3)

Each object identifier in the hostTypes group corresponds to a serial host port. For example, the previous sections give the Enterprise MIB an object identifier of 529 and the hostTypes group identifier is 3, and since the object identifier of hostTypeAim2 within the hostTypes group is 4, the value of the hostTypeAim2 would be 1.3.6.1.4.1.529.3.4. The values reference external MIBs to specific types serial host port services, if present. Currently no MIBs to specific types of serial host port services have been defined, and noSuchName is returned.

Table 4-5. Host types

Value	Object Identifier	Reference
hostTypeAny	hostTypes.1	none of the following
hostTypeDual	hostTypes.2	Host/Dual (see slotType, dualHost)
hostTypeQuad	hostTypes.3	Host/Quad (see slotType, quadHost)
hostTypeAim2	hostTypes.4	Host/Dual (see slotType, aim2)
hostTypeAim6	hostTypes.5	Host/6 (see slotType, aim6)

Ascend WAN types group (ascend 4)

Each object identifier in the wanTypes group corresponds to a type of WAN interface. For example, the previous sections give the Enterprise MIB an object identifier of 529 and the wanTypes group identifier is 4, and since the object identifier of wanTypeE1 within the wanTypes group is 3, the value of the wanTypeE1 would be 1.3.6.1.4.1.529.4.3. The values reference external MIBs to specific types of WAN services, if present. Currently no MIBs to specific types of WAN services been defined, and noSuchName is returned. (See “slotName (slotEntry 2) read-only” on page 4-4.)

Table 4-6. WAN types

Value	Object Identifier	Reference
wanTypeAny	wanTypes.1	none of the following
wanTypeT1	wanTypes.2	Net/T1 (see slotName, Net/T1)
wanTypeE1	wanTypes.3	Net/E1 (see slotName, Net/E1)
wanTypeDpnss	wanTypes.4	Net/DPNSS (see slotName, Net/DPNSS)
wanTypeBRI	wanTypes.5	Net/BRI (see slotName, Net/BRI)
wanTypeS562	wanTypes.4	Net/S56-2 (see slotName, Net/S56-2)
wanTypeS564	wanTypes.5	Net/S56-4 (see slotName, Net/S56-4)

Ascend LAN types group (ascend 5)

Each object identifier in the lanTypes group corresponds to a type of LAN interface. For example, the previous sections give the Enterprise MIB an object identifier of 529 and the lanTypes group identifier is 5, and since the object identifier of lanTypeEthernet within the lanTypes group is 2, the value of the lanTypeEthernet would be 1.3.6.1.4.1.529.5.2. The values reference external MIBs to specific types of LAN services, if present. Currently no MIBs to specific types LAN services have been defined, and noSuchName is returned.(See "slotName (slotEntry 2) read-only" on page 4-4.)

Table 4-7. LAN types

Value	Object Identifier	Reference
lanTypeAny	lanTypes.1	none of the following
lanTypeEthernet	lanTypes.2	Ethernet (see slotName, Ethernet)
lanEtherData	lanTypes.3	Ether Data (see slotName, Ether Data)

Ascend DO group (ascend 6)

This group returns values for objects that indicate the ability of the referenced slot/item to execute or not execute a particular doGroup action. The doGroup objects allow control of a product (in the Products Group) by an SNMP manager. The doGroup operations are equivalent to the DO commands in the menu-driven interface of the product.)

A doGroup action is requested by writing a value to that object. You must have the community value set for read/write to execute a doGroup command; that is, set the Ethernet Profile parameter R/W Comm=write. If the referenced item can execute the requested action, writing a value to this object executes the action. If the referenced slot/item cannot execute the request, the request is ignored.

All indexed entries in the doTable are read-write. In a GET operation, these objects return valid (1) or notValid (2).

In a SET operation, if the object is valid (1), the operation is carried out; otherwise, the SET is ignored.

The DO table is defined as follows:

doTable (doGroup 1) not-accessible

The list of DO entries.

doEntry (doTable 1) not-accessible

An entry in the DO table.

doSlotIndex (doEntry 1) read-only

The slot index pointing to the slot that holds an item for which an action is desired. The valid range is 1 through the value slotNumber.

doItemIndex (doEntry 2) read-only

The item index on the indexed slot pointing to the item for which an action is desired. The valid range is 1 through the value slotItems in the slotTable when indexed by a value equal to doSlotIndex.

doDial (doEntry 3) read-write

When read, returns valid(1) if the indexed slot/item can dial a call. Writing any value causes a call to be placed when valid, otherwise the write is ignored. doDial does not cause the dialed phone number to stored in any Call or Connection Profile.

doHangUp (doEntry 4) read-write

When read, returns valid(1) if the indexed slot/item can terminate a call. Writing any value causes the call to be terminated when valid, otherwise the write is ignored.

doAnswer (doEntry 5) read-write

When read returns valid(1) if the indexed slot/item can answer a call. Writing any value causes the call to be answered when valid, otherwise the write is ignored.

doExtendBW (doEntry 6) read-write

When read returns valid(1) if the indexed slot/item can extend bandwidth. Writing any value causes bandwidth to be extended when valid, otherwise the write is ignored.

doContractBW (doEntry 7) read-write

When read returns valid(1) if the indexed slot/item can contract bandwidth. Writing any value causes bandwidth to be contracted when valid, otherwise the write is ignored.

doBegEndRemoteLB (doEntry 8) read-write

When read returns valid(1) if the indexed slot/item can enter or exit remote loopback. If valid and a remote loopback is up, writing any value causes the loopback to be taken down. If valid and a remote loopback is down, writing any value causes the loopback to be put up. If notvalid(2), writing any value is ignored. Returned values are specified in the MIB.

doBegEndBERT (doEntry 9) read-write

When read returns valid(1) if the indexed slot/item can enter or exit BERT test mode. Writing any value causes the current BERT test state to be toggled when valid, otherwise the write is ignored.

doResynchronize (doEntry 10) read-write

When read returns valid(1) if the indexed slot/item can issue a resynchronize command. Writing any value causes a resynchronize command to be issued when valid, otherwise the write is ignored.

Ascend host status group (ascend 7)

The status group displays the status of a session when indexed by slot and item. No such item is returned if the slot/item does not reference something that identifies a session.

The hostStatus group consists of a single table indexed by slot and item number. The ASN.1 notation for the table and the entry within the table is as follows:

hostStatusTable (hostStatus 1) not-accessible

The hostStatusTable only exists for serial host ports. When indexed by slot and item (where item is a port number) important information about the current Call Profile and the state of the port can be displayed. When the slot/item index points to items other than serial host ports, noSuchName is returned.

hostStatusEntry (hostStatusTable 1) not-accessible

The hostStatusEntry is an entry in the hostStatusTable.

hostStatusSlotIndex (hostStatusEntry 1) read-only

The slot index pointing to the slot that holds an item for which a status report is desired. The valid range is 1 through the value slotNumber.

hostStatusItemIndex (hostStatusEntry 2) read-only

The item index on the indexed slot pointing to the serial host port for which a status report is desired. The valid range is 1 through the value slotItems in the slotTable when indexed by a value equal to hostStatusSlotIndex.

hostStatusLocalName (hostStatusEntry 3) read-only

The name of the serial host port pointed to by slot/item. This name is given by the Port Name parameter in the menu-driven interface.

hostStatusDialNum (hostStatusEntry 4) read-only

The phone number of the current Call Profile of the serial host port pointed to by slot/item. This phone number is given by the Dial # parameter in the Directory Menu of the menu-driven interface.

hostStatusCallType (hostStatusEntry 5) read-only

The call type of the current Call Profile pointed to by slot/item. This call type is given by the Call Type parameter in the Directory Menu of the menu-driven interface. For details about these call types, see the Call Type parameter in the *MAX Reference Guide*. This object can have any of the following values:

Table 4-8. Call types

Value	Reference
aim (1)	AIM call type, Call Type parameter
bonding (2)	BONDING call type, Call Type parameter
one-channel (3)	1 Chnl call type, Call Type parameter
two-channel (4)	2 Chnl call type, Call Type parameter
ft1 (5)	BONDING call type, Call Type parameter
ft1Aim (6)	FT1-AIM call type, Call Type parameter
ft1BandO (7)	FT1-B&O call type, Call Type parameter

hostStatusCallMgm (hostStatusEntry 6) read-only

The call management of the current Call Profile pointed to by slot/item. This call management is given by the Call Mgm parameter in the Directory Menu of the menu-driven interface. This object is only meaningful for AIM or BONDING call types. For details, see the Call Mgm parameter in the *MAX Reference Guide*. This object can have any of the following values:

Table 4-9. Call management

Value	Reference
none (1)	AIM Static call management, Call Mgm parameter
manual (2)	AIM Manual call management, Call Mgm parameter
static (3)	AIM Static call management, Call Mgm parameter
dynamic (4)	AIM Dynamic call management, Call Mgm parameter
delta (5)	AIM Delta call management, Call Mgm parameter
one-of-8 (6)	Special call management mode used in Australia
one-of-40 (7)	Special call management mode used in Australia
mode1 (8)	BONDING Mode 1 call management, Call Mgm parameter
mode2 (9)	BONDING Mode 2 call management, Call Mgm parameter

Table 4-9. Call management (continued)

Value	Reference
mode3 (10)	BONDING Mode 3 call management, Call Mgm parameter
mode0 (11)	BONDING Mode 0 call management, Call Mgm parameter.

hostStatusDataSvc (hostStatusEntry 7) read-only

The data service of the current Call Profile pointed to by slot/item. This data service is given by the Data Svc parameter in the Directory Menu of the menu-driven interface. hostStatus-DataSvc can have a value of serviceVoice or variety of digital services, such as service56KR, which is equivalent to digital service of 56 kbit/s, restricted. For a detailed description of the values of this object, see the Data Svc parameter in the *MAX Reference Guide*.

Table 4-10. Data service types

Value	Reference
serviceVoice (1)	Voice, Data Svc parameter (ISDN lines only)
service56KR (2)	56KR, Data Svc parameter (ISDN lines only)
service56K (3)	56K, Data Svc parameter
service64K (4)	64K, Data Svc parameter (ISDN lines only)
service384KR (5)	384KR, Data Svc parameter (PRI lines only)
service384K-H0 (6)	384K/H0, Data Svc parameter (T1-PRI lines only)
service1536K (7)	1536K, Data Svc parameter (T1-PRI lines only)
service1536KR (8)	1536KR, Data Svc parameter (PRI lines only)
service128K (9)	128K, Data Svc parameter (PRI lines only)
service192K (10)	192K, Data Svc parameter (PRI lines only)
service256K (11)	256K, Data Svc parameter (PRI lines only)
service320K (12)	320K, Data Svc parameter (PRI lines only)
service384K (13)	384K, Data Svc parameter (PRI lines only)
service448K (14)	448K, Data Svc parameter (PRI lines only)
service512K (15)	512K, Data Svc parameter (PRI lines only)
service576K (16)	576K, Data Svc parameter (PRI lines only)
service640K (17)	640K, Data Svc parameter (PRI lines only)
service704K (18)	704K, Data Svc parameter (PRI lines only)
service768K (19)	768K, Data Svc parameter (PRI lines only)
service832K (20)	832K, Data Svc parameter (PRI lines only)
service896K (21)	896K, Data Svc parameter (PRI lines only)

Table 4-10. Data service types (continued)

Value	Reference
service960K (22)	960K, Data Svc parameter (PRI lines only)
service1024K (23)	1024K, Data Svc parameter (PRI lines only)
service1088K (24)	1088K, Data Svc parameter (PRI lines only)
service1152K (25)	1152K, Data Svc parameter (PRI lines only)
service1216K (26)	1216K, Data Svc parameter (PRI lines only)
service1280K (27)	1280K, Data Svc parameter (PRI lines only)
service1344K (28)	1344K, Data Svc parameter (PRI lines only)
service1408K (29)	1408K, Data Svc parameter (PRI lines only)
service1472K (30)	1472K, Data Svc parameter (PRI lines only)
service1600K (31)	1600K, Data Svc parameter (PRI lines only)
service1664K (32)	1664K, Data Svc parameter (PRI lines only)
service1728K (33)	1728K, Data Svc parameter (PRI lines only)
service1792K (34)	1792K, Data Svc parameter (PRI lines only)
service1856K (35)	1856K, Data Svc parameter (PRI lines only)
service1920K (36)	1920K, Data Svc parameter (PRI lines only)
servicemodem (37)	Modem, Data Svc parameter

hostStatusCallState (hostStatusEntry 8) read-only

The state of the current call at the serial host port pointed to by slot/item. Except for other(1) and remoteMg(13), this call status is given by the first line of the Call Status menu in the menu-driven interface.

Table 4-11. Call states

Value	Reference
other (1)	none of the following
answering (2)	ANSWERING, Call Status menu
calling (3)	CALLING, Call Status menu
clearing (4)	CLEARING, Call Status menu
localLoop (5)	LOCAL LOOP, Call Status menu
handshake (6)	HANDSHAKE, Call Status menu
idle (7)	IDLE, Call Status menu
online (8)	ONLINE, Call Status menu

Table 4-11. Call states (continued)

Value	Reference
loopMast (9)	LOOP MAST, Call Status menu
loopSlav (10)	LOOP SLAV, Call Status menu
bertMast (11)	BERT MAST, Call Status menu
bertSlav (12)	BERT SLAV, Call Status menu
remoteMg (13)	Remote Mgm. DO menu
ringing (14)	RINGING, Call Status menu
setupAdd (15)	SETUP ADD, Call Status menu (preparing to add channels with I-mux call online)
setupHnd (16)	SETUP HND, Call Status menu (preparing to resynchronize with I-mux call online)
setupRem (17)	SETUP REM, Call Status menu (preparing to remove channels with I-mux call online)

hostStatusRemName (hostStatusEntry 9) read-only

The name of the far-end of the current AIM call at the serial host port pointed to by slot/item. If this is not an AIM call or if no call is online, this object returns an empty string. This far-end name is given by the fourth line of the Call Status menu in the menu-driven interface.

hostStatusChannels (hostStatusEntry 10) read-only

The number of channels in use by the current AIM call at the serial host port pointed to by slot/item. If no call is online, this object returns an empty string. The number of channels of a call is given by the third line of the Call Status menu in the menu-driven interface. For further information, see "Ascend call status group (ascend 11)" on page 4-24.

hostStatusDuration (hostStatusEntry 11) read-only

The duration of the current call at the serial host port pointed to by slot/item; measured in seconds. If no call is online, this object returns an empty string.

Ascend console group (ascend 8)

The console group manages console interfaces. These interfaces include every physical console interface, including VT100 and Palmtop console ports, and interfaces to potential TELNET sessions whether active or not. The consoleTable maps console numbers to interface numbers (ifIndex in the Interfaces Group of MIB-2) and includes the type of console.

Consoles of Ascend products are numbered by the following rules:

- The DE-9 port on the back of the unit is always console port 1 and is called the primary port.
- The first console port on the lowest numbered slot (fixed or removable) is console 2. On the Multiband Plus, this is the Palmtop port on the front of the unit. On the standard MAX, this is the Palmtop port for the fixed 2-port serial host card in slot 2.

- If there are multiple console ports on a slot (fixed or removable), they are assigned in ascending order. On a Multiband Plus with 4 serial host ports, for example, the rear left Palmtop port is console 3 and the rear right Palmtop port is console port 4.
- Virtual console ports are numbered following the physical console ports. There is one virtual console port for every possible TELNET or remote management session. The number of sessions varies depending on the Ascend model and its hardware options. Some representative configurations follow:
  - Multiband Plus: 2 virtual sessions
  - Pipeline 50: 5 virtual sessions
  - Pipeline 400/BRI: 11 virtual sessions
  - MAX/T1 with Ethernet card: 35 virtual sessions
  - MAX/T1 with Ethernet and Ether-Data card: 51 virtual sessions
  - MAX/T1 with Ethernet, Ether-Data card, and second Net/T1 card: 67 virtual sessions

The console number states the number of supported consoles.

consoleNumber (console 1) read-only

The number of consoles supported by the Ascend unit, including the virtual console ports.

The console table and its entries are defined as follows:

consoleTable (console 2) not-accessible

The table of consoles supported by the Ascend unit.

consoleEntry (consoleTable 1) not-accessible

An entry in the consoleTable.

consoleIndex (consoleEntry 1) read-only

The index to the consoleTable. This index ranges from 1 to the value of consoleNumber.

consoleIf (consoleEntry 2) read-only

The interface (console port) number (ifIndex in the Interfaces Group of MIB-2) associated with the entry in this table pointed to by consoleIndex.

consoleType (consoleEntry 3) read-only

The type of console pointed to by consoleIndex. This object returns the following values:

Table 4-12. Console types

Value	Description
other(1)	none of the following
primary(2)	VT100 (or emulator) plugged into primary port
secondary(3)	VT100 (or emulator) plugged into Palmtop port
palmtop(4)	Palmtop plugged into Palmtop port
inactive(5)	no console plugged into physical port or inactive virtual port
remote(6)	active remote management or TELNET session

consoleSecurity (consoleEntry 4) read-only

The Security Profile, in the range of 1 through the number of Security Profiles supported by the unit, of the indexed console.

consoleSpecific (consoleEntry 5) read-only

A reference to MIB definitions specific to the indexed console. There is no item specific information at this time and noSuchName is returned.

Ascend system status group (ascend 9)

This group contains general information regarding the system as a whole. This information provided by this group can be viewed as an extension of the system group under MIB-II.

sysAbsoluteStartupTime (systemStatusGroup 1) read-only

The time the system started up in seconds since January 1, 1990. This value in concert with sysUpTime in MIB-II can be used to compute the current time in seconds since 1/1/90.

sysSecsSinceStartup (systemStatusGroup 2) read-only

The time in seconds since the system stated up.

sysMibVersionNum (systemStatusGroup 3) read-only

Version number of this private MIB. Versions numbers start at 1 and increment by 1 for every major revision. Major revisions are additions/deletions to this MIB.

sysMibMinorRevNum (systemStatusGroup 4) read-only

The minor revision number of this private MIB. Minor revision numbers start at 0 and increment by 1 for each revision. Minor revisions are textual or comment changes.

Ascend events group (ascend 10)

This group contains a table of events. This events table is organized in a “ring” buffer fashion. That is, when the events table is full, a new event will overwrite the oldest event in the table. (Multiband Plus, VSX, and Pipeline 50 do not support this group.)

Each event is assigned a unique ID number (eventIdNumber) which is also used as the “index” into the table. The eventIdNumber increments successively by 1 for each new event. The type of events recorded are specified by eventType, for example each incoming call is an event.

SNMP manager should periodically poll the events table and retrieve new events. The poll scenario follows:

1. Manager determines the ID of the most recent event by getting the current value of eventLatestEventIdNumber and compares it against the value from the previous poll. No difference implies no new events have occurred and manager takes no further action. If a difference exists, it indicates the number of new events that have occurred.
2. Manager retrieves all the new events. For example, if eventLatestEventIdNumber=677 and 10 new events have occurred, The new events have indices (eventIdNumber)= 667, 668, 669, to 677.
3. After retrieving the last event, manager saves the value of eventLatestEventIdNumber so it can be used for comparison on the next poll.

Note: If the events table is not read frequently enough, the number of unread events can exceed the table's capacity (given in eventMaximumNumberOfEvents). Since the table is organized as a “ring” buffer, retrieving the oldest unread entries can result in a noSuchName error. Also note that event IDs can wrap around. If eventOldestEventIdNumber is greater than eventLatestEventIdNumber wrap-around has occurred.

Information held in the Ascend Events Group is erased and its values are initialized when the Ascend unit is reset by software or by toggling the power off and on. It is highly recommended that sysAbsoluteStartupTime be retrieved at the start of every poll. Comparing sysAbsoluteStartupTime against the previous poll’s value indicates whether the unit has reset.

These objects determine the number of event entries.

eventMaximumNumberOfEvents (eventGroup 1) read-only

The maximum number of event entries that can exist in the event table.

eventOldestEventIdNumber (eventGroup 2) read-only

The ID number of the oldest event in the table (given by its eventIdNumber). A value of 0 indicates no events exist in the table.

eventLatestEventIdNumber (eventGroup 3) read-only

The ID number of the most recent event in the table (given by its eventIdNumber). The SNMP manager should poll this variable periodically and compare it with the previous poll’s value to detect the presence of new events. A value of 0 indicates no events exist in the table.

The event table and its entries are defined as follows:

eventTable (eventGroup 4) not-accessible

A list of event entries.

eventEntry (eventTable 1) not-accessible

An entry containing object variables which describes an event.

eventIdNumber (eventEntry 1) read-only

A unique number assigned to every event entry. Numbers are assigned incrementally starting with 1. Wrap-around occurs after the upper limit (2147483648) is reached.

eventTimeStamp (eventEntry 2) read-only

The time of occurrence for this event in seconds since startup. Use sysAbsoluteStartupTime to convert to absolute time.

eventType (eventEntry 3) read-only

Identifies the type of event associated with this entry. (serviceChanged and nameChanged not apply to calls to/from serial host ports.)

Table 4-13. Event types

Event Type	Description
callOriginated(1)	unit dialed out
callAnswered(2)	unit answered call

Table 4-13. Event types (continued)

Event Type	Description
callCleared(3)	call terminated
serviceChanged(4)	user service changed
nameChanged(5)	name, IP address, mask changed

eventCallReferenceNum (eventEntry 4) read-only

A unique number assigned to identify a particular session. The start of a session is marked by either a callOriginated(1) or a callAnswered(2) event. The end of a session is marked by a callCleared(3) event. Numbers are assigned incrementally starting with 1. Wrap-around occurs after the upper limit (2147483648) is reached.

eventDataRate (eventEntry 5) read-only

The data rate for non-modem calls or the baud rate for modem calls. Rate is given as bits-per-second. Applicable only if eventType is callOriginated(1) or callAnswered(2) otherwise 0 is returned.

eventSlotNumber (eventEntry 6) read-only

Identifies the slot containing the line on which a call was received, dialed, or cleared. eventSlotNumber value ranges between 1 and the value slotNumber in Ascend’s Slot group. This variable is equivalent to slotIndex in Ascend’s slot group. Applicable only if eventType is callOriginated(1), callAnswered(2), or callCleared(3) otherwise 0 is returned.

eventSlotLineNumber (eventEntry 7) read-only

Identifies the number of the line on which a call was received, dialed, or cleared. The line is in the slot identified by eventSlotNumber. This variable is equivalent to slotItemIndex in Ascend’s Slot group. Applicable only if eventType is callOriginated(1), callAnswered(2), or callCleared(3) otherwise 0 is returned.

eventSlotChannelNumber (eventEntry 8) read-only

Identifies the channel on which a call was received, dialed, or cleared. The channel is on the line identified by eventSlotLineNumber. Applicable only if eventType is callOriginated(1), callAnswered(2), or callCleared(3) otherwise 0 is returned.

eventModemSlotNumber (eventEntry 9) read-only

Identifies the slot containing the modem on which a call was received or dialed. Its value ranges between 0 and the value slotNumber in Ascend’s Slot group. A value of 0 indicates modems are not associated with this call. Applicable only if eventType is callOriginated(1) or callAnswered(2) otherwise 0 is returned.

eventModemOnSlot (eventEntry 10) read-only

Identifies the particular modem on which a call was received or dialed. The modem is in the slot identified by eventModemSlotNumber. A value of 0 indicates modems are not associated with this call. Applicable only if eventType is callOriginated(1) or callAnswered(2) otherwise 0 is returned.

eventCurrentService (eventEntry 11) read-only

The service provided to the remote user after service or name was changed. Applicable only if eventType is serviceChanged(4) or nameChanged(5) otherwise none(1) is returned.

Table 4-14. Current service types

Current Service	Description
none(1)	N/A (Calls to/from serial host ports have no “packet service”, because they are treated as simple byte streams; for example, videoconference.)
other(2)	none of the following
ppp(3)	Point-To-Point Protocol
slip(4)	Serial Line IP
mpp(5)	Multichannel PPP
x25(6)	X.25
combinet(7)	Combinet
frameRelay(8)	Frame Relay
euraw(9)	EURAW
euui(10)	EUUI
telnet(11)	telnet
telnetBinary(12)	binary telnet
rawTcp(13)	raw TCP
terminalServer(14)	terminal server
mp(15)	Multilink PPP

eventUserName (eventEntry 12) read-only

The name of the remote user. Applicable only if eventType is serviceChanged(4) or nameChanged(5). The null string is returned if the name is unknown or if not applicable. (Does not apply to calls to/from serial host ports.)

eventUserIPAddress (eventEntry 13) read-only

The IP address of the remote user. Applicable only if eventType is serviceChanged(4) or nameChanged(5). The value 0.0.0.0 is returned if address is unknown or if not applicable. (Does not apply to calls to/from serial host ports.)

eventUserSubnetMask (eventEntry 14) read-only

The subnet mask of the remote user. Applicable only if eventType is serviceChanged(4) or nameChanged(5). The value 0.0.0.0 is returned if mask is unknown or if not applicable. (Does not apply to calls to/from serial host ports.)

eventDisconnectReason (eventEntry 15) read-only

The reason a call was disconnected. Applicable only if eventType is callCleared(3), otherwise notApplicable(1) is returned. Unknown(2) indicates that the reason for the disconnect is unknown.

The following reasons apply to modem connections:

- noModemNoCarrier(10): no Carrier detected ever
- noModemLossCarrier(11): loss of Carrier
- noModemResultCodes(12): fail to detect modem result codes

The following reasons apply to terminal server sessions:

- tsUserExit(20): user exited terminal server
- tsIdleTimeout(21): timeout waiting for user input
- tsExitTelnet(22): disconnect due to exiting telnet
- tsNoIPAddr(23): could not switch to SLIP/PPP, Remote has no IP address or could not assign one
- tsExitTcp(24): disconnect due to exiting raw TCP
- tsPassWordFail(25): bad passwords
- tsRawTCPDisable(26): raw TCP disabled
- tsControlC(27): Control-C char detected
- tsDestroyed(28): terminal server destroyed

The followings reasons apply to PPP sessions:

- pppLcpTimeout(40): PPP LCP negotiation timed out
- pppLcpNegotiateFail(41): PPP LCP negotiation failed
- pppPAPAuthFail(42): PPP PAP authentication failed
- pppCHAPAuthFail(43): PPP CHAP authentication failed
- pppRemoteAuthFail(44): PPP remote authentication failed
- pppRcvTerminate(45): PPP receive Terminate Request from far end
- pppCloseEvent(46): upper layer requested a Close

The following are miscellaneous reasons:

- sessTimeOut(100): session timed out
- sessFailSecurity(101): session failed for security reasons (such as an incoming call for an outgoing-only profile).
- sessCallback(102): session terminated due to callback
- invalidProtocol(120): call refused, detected protocol is disabled

eventConnectProgress (eventEntry 16) read-only

State of the connection before disconnecting. Applicable only if eventType is callCleared(3) otherwise prNotApplicable(1) is returned.

- prNotApplicable(1)
- prUnknown(2): progress unknown
- prCallUp(10): call up
- prModemUp(30): modem up
- prModemWaitDCD(31): waiting for DCD (Carrier)

- prModemWaitCodes(32): waiting for result codes
- prTermSrvStarted(40): terminal server session started up
- prLanSessionUp(60): LAN session up
- prOpeningLCP(61): LCP negotiations allowed
- prOpeningCCP(62): CCP negotiations allowed
- prOpeningIPNCP(63): IP NCP negotiations allowed
- prOpeningBNCP(64): Bridging NCP negotiations allowed
- prLCPOpened(65): LCP in Open state
- prCCPOpened(66)
- prIPNCPOpened(67): IP NCP in Open state
- prBNCPOpened(68): Bridging NCP in Open state
- prLCPStateInitial(69): LCP in Initial state
- prLCPStateStarting(70): LCP in Starting state
- prLCPStateClosed(71): LCP in Closed state
- prLCPStateStopped(72): LCP in Stopped state
- prLCPStateClosing(73): LCP in Closing state
- prLCPStateStopping(74): LCP in Stopping state
- prLCPStateReqSent(75): LCP in Request Sent state
- prLCPStateAckRecd(76): LCP in ACK Received state
- prLCPStateAckSent(77): LCP in ACK Sent state

Ascend call status group (ascend 11)

This group contains a table of entries for the status of each possible call in the system. (Multiband Plus, VSX, and Pipeline 50 do not support this group.)

The call status table and its entries are defined as follows:

callStatusMaximumEntries (callStatusGroup 1) read-only

The maximum number of entries that can exist in the Call Status table.

callStatusTable (callStatusGroup 2) not-accessible

A list of call status entries.

callStatusEntry (callStatusTable 1) not-accessible

An entry containing object variables to describe a call's status.

callStatusIndex (callStatusEntry 1) read-only

The index number for this call status entry. Its value ranges from 1 to callStatusMaximumEntries.

callStatusValidFlag (callStatusEntry 2) read-only

The value can be invalid(1) or valid(2). The value indicates whether this entry indexed by callStatusIndex contains valid information or not.

callStatusStartingTimeStamp (callStatusEntry 3) read-only

The starting time for this call in seconds since startup. A value of 0 is returned if entry is invalid (that is, callStatusValidFlag is invalid(1)).

callStatusCallReferenceNum (callStatusEntry 4) read-only

The unique number identifying the session for which this call is associated. See "eventCallReferenceNum (eventEntry 4) read-only" on page 4-21 for further details. A value of 0 is returned if entry is invalid.

callStatusDataRate (callStatusEntry 5) read-only

The data rate for non-modem calls or the baud rate for modem calls. A value of 0 is returned if entry is invalid.

callStatusSlotNumber (callStatusEntry 6) read-only

Identifies the slot of the line this call uses. Its value ranges between 1 and the value slotNumber in Ascend's slots group. This variable is equivalent to slotIndex in the slot group. A value of 0 is returned if entry is invalid.

callStatusSlotLineNumber (callStatusEntry 7) read-only

Identifies the number of the line this call uses. This line is in the slot given by callStatusSlotNumber. This variable is equivalent to slotItemIndex in Ascend's slot group. A value of 0 is returned if entry is invalid.

callStatusSlotChannelNumber (callStatusEntry 8) read-only

Identifies the channel associated with this call. This channel is in the line given by callStatusSlotLineNumber. A value of 0 is returned if entry is invalid.

callStatusModemSlotNumber (callStatusEntry 9) read-only

Identifies the slot containing the modem associated with this call. Its value ranges between 1 and the value slotNumber in Ascend's slot group. A value of 0 is returned if entry is invalid or if modems are not involved in the connection.

callStatusModemOnSlot (callStatusEntry 10) read-only

Identifies the particular modem associated with this call. This modem is in the slot given by callStatusModemSlotNumber. A value of 0 indicates modems are not involved in this connection or if entry is invalid.

callStatusIfIndex (callStatusEntry 11) read-only

The interface index of this call. callStatusIfIndex ranges from 1 to the number of interfaces specified in the MIB-II variable ifNumber. The interface identified by a particular value of this index is the same interface as identified by the same value if ifIndex. A value of 0 is returned if entry is invalid.

callSessionIndex (callStatusEntry 12) read-only

The index of the associated sessionStatusEntry. Value ranges from 1 to ssnStatusMaximumSessions. A value of 0 is returned if entry is invalid.

Ascend session status group (ascend 12)

This group contains a table of the status for each possible session in the system. (Multiband Plus, VSX, and Pipeline 50 do not support this group.)

The session status table and its entries are defined as follows:

ssnStatusMaximumSessions (sessionStatusGroup 1) read-only

The maximum number of sessions that can exist in the system.

sessionStatusTable (sessionStatusGroup 2) not-accessible

A list of session status entries.

sessionStatusEntry (sessionStatusTable 1) not-accessible

An entry containing object variables which describe an active session.

ssnStatusIndex (sessionStatusEntry 1) read-only

The index number for this session status entry. Its value ranges from 1 to ssnStatusMaximum-Sessions.

ssnStatusValidFlag (sessionStatusEntry 2) read-only

The value indicates whether this entry contains valid information or not. It can be invalid(1) or valid(2).

ssnStatusUserName (sessionStatusEntry 3) read-only

The name of the remote user. The null string is returned if entry is invalid or if the name is unknown. (Does not apply to calls to/from serial host ports.)

ssnStatusUserIPAddress (sessionStatusEntry 4) read-only

The IP address of the remote user, including the case where the Ascend unit assigns the remote user an IP address assigned from an address pool. The value 0.0.0.0 is returned if entry is invalid or if the IP address is unknown. (Does not apply to calls to/from serial host ports.)

ssnStatusUserSubnetMask (sessionStatusEntry 5) read-only

The subnet mask of the remote user. The value 0.0.0.0 is returned if entry is invalid or if the subnet mask is unknown. (Does not apply to calls to/from serial host ports.)

ssnStatusCurrentService (sessionStatusEntry 6) read-only

The current service provided to the remote user. The value none(1) is returned if entry is invalid. The value none(1) is also returned if the user dials into the terminal server and is in midst of a login sequence. This variable has the same range of values as eventCurrentService (eventEntry 11) in the Events Group.

ssnStatusCallReferenceNum (sessionStatusEntry 7) read-only

A unique number identifying this session. See "eventCallReferenceNum (eventEntry 4) read-only" on page 4-21 for further details. The value 0 is returned if entry is invalid.

Ascend RADIUS group (ascend 13)

The RADIUS group contains statistics for RADIUS requests. The first two objects define the number of RADIUS servers supported.

radiusNumAuthServers (radiusGroup 1) read-only

The maximum number of RADIUS authentication servers supported by the system.

radiusNumAcctServers (radiusGroup 2) read-only

The maximum number of RADIUS accounting servers supported by the system.

RADIUS authentication statistics

The RADIUS authentication statistics table is defined as follows:

radiusAuthStatsTable (radiusGroup 3) not-accessible

A list of entries for RADIUS authentication statistics.

radiusAuthStatsEntry (radiusAuthStatsTable 1) not-accessible

An entry containing object variables to describe a session.

radAuthServerIndex (radiusAuthStatsEntry 1) read-only

The index number for this session status entry. Its value ranges from 1 to radiusNumAuthServers and identifies which server the entry is associated with.

radAuthLoginRqstSent (radiusAuthStatsEntry 2) read-only

Total number of authentication requests sent. These requests are strictly for authentication purposes. The sum of this value and radAuthOtherRqstSent represents the total number of requests sent (excluding RADIUS accounting requests).

radAuthOtherRqstSent(radiusAuthStatsEntry 3) read-only

Total number of requests sent that are not related to authentication or accounting. These include requests for routes, filters, banners, etc.

radAuthRqstTimedOut (radiusAuthStatsEntry 4) read-only

Total number of authentication requests which timed out. Value should be less than or equal to radAuthLoginRqstSent.

radAuthOtherRqstTimedOut (radiusAuthStatsEntry 5) read-only

Total number of requests excluding authentication and accounting which timed out. Value should be less than or equal to radAuthOtherRqstSent.

radAuthRspRcvd (radiusAuthStatsEntry 6) read-only

Total number of responses to authentication requests received. Response could be a NAK or an ACK. Value should match radAuthLoginRqstSent.

The sum of this value and radAuthOtherRspRcvd, radAuthUnexpRspRcvd, and radAuthBadRspRcvd represents the total number of responses received (excluding those related to RADIUS accounting).

radAuthOtherRspRcvd (radiusAuthStatsEntry 7) read-only

Total number of responses to requests that were not related to authentication or accounting. Response could be a NAK or an ACK. Value should be less than or equal to radAuthOther-RqstSent.

radAuthUnexpRspRcvd (radiusAuthStatsEntry 8) read-only

Total number of responses received that did not match an outstanding request. A positive value may indicate configured timeout is insufficient. Total excludes accounting responses.

radAuthBadRspRcvd (radiusAuthStatsEntry 9) read-only

Total number of authentication responses received that were rejected due to:

(1) authentication NAKed.

(2) invalid authenticator field.

(3) invalid packet format.

(4) invalid code field.

(5) unrecognized attribute.

A positive value may indicate mismatch in the shared secrets.

radAuthAckRspRcvd (radiusAuthStatsEntry 10) read-only

Total number of authentication requests which were ACKed or authenticated successfully. Value is less than or equal to radAuthLoginRqstSent.

RADIUS accounting statistics

The RADIUS accounting statistics table is defined as follows:

radiusAcctStatsTable (radiusGroup 4) not-accessible

A list of entries for RADIUS accounting statistics.

radiusAcctStatsEntry (radiusAcctStatsTable 1) not-accessible

An entry containing object variables to describe a session.

radAcctServerIndex (radiusAcctStatsEntry 1) read-only

The index number for this session status entry. Its value ranges from 1 to radiusNumAcctServers and identifies which server the entry is associated with.

radAcctRqstSent (radiusAcctStatsEntry 2) read-only

Total number of accounting requests sent.

radAcctRqstTimedOut (radiusAcctStatsEntry 3) read-only

Total number of accounting requests which timed out.

radAcctRspRcvd (radiusAcctStatsEntry 4) read-only

Total number of responses to accounting requests received.

radAcctUnexpRspRcvd (radiusAcctStatsEntry 5) read-only

Total number of accounting responses received that did not match an outstanding request. A positive value may indicate configured timeout is insufficient.

Industry Standard MIB Implementations

5

All Internet standard MIBs are defined in an RFC (Request for Comment). This chapter describes objects related to Ascend functionality in the following Internet standard MIBs:

MIB 2 implementation (RFC 1213)

The MIB 2 objects are organized into 10 groups. Querying a MIB object that is not supported returns a noSuchName.

In general, read and write access are fully supported per the ACCESS clause in the MIB. However, any data updated with a write is lost at the next system reset.

System group (MIB-2 1)

The system group contains objects that identify the managed node, namely the Ascend product. The objects of the System group and their supported values follow.

sysDescr (system 1) read-only

The Ascend products return the description of the unit in the following format:

Platform Network-interface Serial-number Software-version

where the following definitions apply:

- *Platform* can be Ascend Multiband Plus, Ascend MAX, or Ascend Pipeline
- *Network-interface* is T1/PRI (T1/Primary Rate Interface), E1/PRI (E1/Primary Rate Interface), DPNSS (Digital Private Network Signalling System), BRI (Basic Rate Interface), S56 (Switched-56 Interface: 2-wire or 4-wire), or T1/PRI & BRI (T1/Primary Rate and Basic Rate Interfaces).
- *Serial-number* contains “S/N:” followed by the serial number of the node.
- *Software-version* is the version of software running on the node in the following format:

  Software +*n.nlpn*+

sysObjectID (system 2) read-only

The values returned for sysObjectID identify the platform of the Ascend product as defined by the object identifiers in the products group of the Enterprise MIB.

- multiband (1.3.6.1.4.1.529.1.1)
- max (1.3.6.1.4.1.529.1.2)
- pipeline (1.3.6.1.4.1.529.1.3)

sysUpTime (system 3) read-only

The time in hundredths of a second since the Ascend unit was last re-initialized by turning power on or by software boot.

sysContact (system 4) read-write

The contact responsible for this managed node. The value returned comes from the Contact parameter in the Ascend products’ menu-driven interface. Ascend products limit this field to 80 characters.

sysName (system 5) read-write

The name of the node as assigned by the user. The value returned comes from the Name parameter in the Sys Config branch of the Ascend products’ menu-driven interface. Ascend products limit this field to 16 characters.

sysLocation (system 6) read-write

The location of the node as assigned by the user. The value returned comes from the Location parameter in the Ascend products' menu-driven interface. Ascend products limit this field to 80 characters.

sysServices (system 7) read-only

Fully supports the RFC 1213 definition, without any Ascend-specific changes. The functionality of the Ascend nodes cannot be described simply in terms of the ISO Reference Model. therefore, the method described in RFC 1213 to compute the value of this object does not correlate exactly to the functions of the Ascend nodes.

The values returned for sysServices depend upon the platform (see "sysDescr (system 1) read-only" on page 5-2) of the Ascend product and are as follows:

- 8 (multiband)
- 14 (max and pipeline)

Multibands support TELNET and SNMP which are end-to-end services.

Interfaces group (MIB-2 2)

The interfaces group contains a count of the number of interfaces on the unit and a table with an entry for each interface. Many of the objects in the ifTable are not meaningful for serial host ports. The objects supported for the various interface types follow:

ifNumber (interfaces 1) read-only

The number of interfaces available on the node. The number of interfaces in the product are detected at power-up. Each of the following (if present on the managed node) is counted as an interface:

- Every port to a WAN line (BRI, T1/PRI, or Switched-56) is an interface.
- Every port to a local line (Host/BRI) is an interface.
- Every serial host port (V.35, RS-449, X.21) is an interface.
- Every serial WAN port is an interface.
- Each console port is an interface.
- Each digital modem is an interface.
- The Ethernet port is an interface.
- The internal (software) loopback defined by MIB-2 is an interface.
- Each potential bridge/router session, whether active or not, is an interface. See "ifDescr (ifEntry 2) read-only" on page 5-4.

ifTable (interfaces 2) not-accessible

The interface table.

ifEntry (ifTable 1) not-accessible

Interface table entries.

ifIndex(ifEntry 1) read-only

The index listing the entries in ifTable.

ifDescr (ifEntry 2) read-only

Ascend nodes return values for ifDescr in the following format:

interface-type interface-number

If *interface-type* is "ie0", "lo0", or "wanidle0", *interface-number* is blank:

- ie0 (Ethernet interface)
- lo0 (software loopback interface)
- wanidle0 (interface to idle bridge/router sessions)

If *interface-type* is "wan", *interface-number* is any number from 0 to the number of bridge/router session interfaces:

- wan (interfaces to active bridge/router sessions)

If *interface-type* is "Console", *interface-number* is any number from 1 to the number of console interfaces:

- Console (interfaces to consoles)

If *interface-type* is a type of line that connects to the WAN or a local line that the Ascend unit provides to TE/TAs, *interface-number* is an index giving the line's slot and line number. In addition, if *interface-type* is an RS232-like port that the Ascend unit provides to the local DTEs (serial hosts) or that connects to the WAN (Serial WAN, MAX HP only), *interface-number* is an index giving the ports's slot and port number. The following *interface-types* are lines indexed by slot and line number:

- T1 (T1/PRI lines - WAN interfaces)
- E1/DPNSS (E1/PRI lines - WAN interfaces)
- BRI (BRI lines - WAN and Host interfaces)
- S56/4 Wire (4-Wire Switched-56 Lines - WAN interfaces)
- S56/2 Wire (2-Wire Switched-56 Lines - WAN interfaces)

The following *interface-types* are serial ports indexed by slot and port number:

- AIM6 (Serial host ports in an AIM6 [Host/6] card - Host I/Fs)
- Dual Host (Serial host ports in an Dual Host card - Host I/Fs)
- Quad Host (Serial host ports in an Quad Host card - Host I/Fs)
- Serial WAN (Serial WAN ports - WAN I/Fs)

An empty slot is also an *interface-type* indexed by slot and number:

- Empty (Interface to an empty slot)

If *interface-type* is one of the preceding types of line, serial host port, or an empty slot, *interface-number* is an index in the following format:

slot slot-number port/line item-number

where the following definitions apply:

Slot-number is the slot index of the slot containing the interface.

Item-number is the item index of the item on the slot containing the interface. See Chapter 4, "Ascend Enterprise MIB," for details.

ifType (if Entry 3) read-only

The type of interface, defined according to the physical/link protocol(s) below the network layer in the protocol stack. Interface types not shown below, such as fddi(15), are not currently supported. Ascend products return the following:

- ethernet-csmacd(6) (Ethernet interface)

- ds1(18) (T1 line WAN interface using robbed-bit signaling)
- e1(19) (E1 line WAN interface using DPNSS signaling)
- basicISDN(20) (BRI line Host or WAN interface)
- primaryISDN(21) (T1 or E1 WAN interface using ISDN signaling)
- ppp(23) (bridge/router interface running PPP or MPP)
- softwareLoopback(24) (software loopback interface)
- other(1) (all other interface types including interfaces to serial host ports, Serial WAN ports, consoles, or empty slots)

ifMTU (ifEntry 4) read-only

The value of ifMTU specifies the largest datagram (in octets) that can be sent/received through the interface. This variable applies only to the Ethernet interface and other interfaces bridging or routing packets. No value (noSuchName) is returned for interfaces not processing packets such as serial host ports, consoles, empty slots.

ifSpeed (ifEntry 5) read-only

An estimate of the interface's current bandwidth in bits per second. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object contains the nominal bandwidth. Serial host port interfaces return the current bandwidth in use by the port.

ifPhysAddress (ifEntry 6) read-only

For Ethernet interfaces, ethernet-csmacd(6), ifPhysAddress returns the six byte Ethernet physical address. For softwareLoopback (24), this object returns a 0 value. For all other Ascend products' interfaces, this object is not supported and returns no value (noSuchName).

ifAdminStatus (ifEntry 7) read-write

The desired state of the interface. The read/write value is either Up(1) or Down(2).

ifOperStatus (ifEntry 8) read-only

The current operational state of the interface. The returned value is either Up(1) or Down(2).

ifLastChange (ifEntry 9) read-only

Not supported, noSuchName returned on queries.

ifInOctets, (ifEntry 10) read-only

The total number of octets received on the interface, including framing characters. This variable applies to the Ethernet interface and other interfaces bridging or routing packets. No value (noSuchName) is returned for interfaces not processing packets such as serial host ports, consoles, empty slots.

ifInUcastPkts (ifEntry 11) read-only

The number of subnetwork-unicast packets delivered to a higher-layer protocol. This variable applies to the Ethernet interface and other interfaces bridging or routing packets. No value (noSuchName) is always returned for interfaces not processing packets such as serial host ports, consoles, empty slots.

ifInNUcastPkts (ifEntry 12) read-only

The number of non-unicast (i.e., subnetwork-broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol. This variable applies to the Ethernet interface and other interfaces bridging or routing packets. No value (noSuchName) is always returned for interfaces not processing packets such as serial host ports, consoles, empty slots.

ifInDiscards (ifEntry 13) read-only

The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. This variable applies to the Ethernet interface and other interfaces bridging or routing packets. No value (noSuchName) is always returned for interfaces not processing packets such as serial host ports, consoles, empty slots.)

ifInErrors (ifEntry 14) read-only

The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. This variable applies to the Ethernet interface and other interfaces bridging or routing packets. No value (noSuchName) is always returned for interfaces not processing packets such as serial host ports, consoles, empty slots.

ifInUnknownProtos (ifEntry 15) read-only

The number of packets received via the interface which were discarded because of an unknown or unsupported protocol. This variable applies to the Ethernet interface and other interfaces bridging or routing packets. No value (noSuchName) is always returned for interfaces not processing packets such as serial host ports, consoles, empty slots.

ifOutOctets (ifEntry 16) read-only

The total number of octets transmitted out of the interface, including framing characters. This variable applies to the Ethernet interface and other interfaces bridging or routing packets. No value (noSuchName) is always returned for interfaces not processing packets such as serial host ports, consoles, empty slots.

ifOutUcastPkts (ifEntry 17) read-only

The total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or not sent This variable applies to the Ethernet interface and other interfaces bridging or routing packets. No value (noSuchName) is always returned for interfaces not processing packets such as serial host ports, consoles, empty slots.

ifOutNUcastPkts (ifEntry 18) read-only

The total number of packets that higher-level protocols requested be transmitted to a non- unicast (i.e., a subnetwork-broadcast or subnetwork-multicast) address, including those that were discarded or not sent (For the Ethernet interface and other interfaces bridging or routing packets. No value (noSuchName) is always returned for interfaces not processing packets such as serial host ports, consoles, empty slots.)

ifOutDiscards (ifEntry 19) read-only

The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet is to free up buffer space This variable applies to the Ethernet interface and other interfaces bridging or routing packets. No value (noSuchName) is always returned for interfaces not processing packets such as serial host ports, consoles, empty slots.

ifOutErrors (ifEntry 20) read-only

The number of outbound packets that could not be transmitted because of errors This variable applies to the Ethernet interface and other interfaces bridging or routing packets. No value (noSuchName) is always returned for interfaces not processing packets such as serial host ports, consoles, empty slots.

ifOutQLen (ifEntry 21) read-only

The length of the output packet queue (in packets) This variable applies to the Ethernet interface and other interfaces bridging or routing packets. No value (noSuchName) is always returned for interfaces not processing packets such as serial host ports, consoles, empty slots.

ifSpecific (ifEntry 22) read-only

This object indicates a MIB definition for media that supports a given interface. If Ascend currently does not support a MIB specific to that media, noSuchName is returned.

- ds1(18) (T1/PRI WAN line interface)
- e1(19) (E1/PRI WAN line interface)
- rs232(33) (interfaces to serial host ports)
- noSuchName (all other interface types including empty slots)

Address translation group (MIB-2 3)

This group, although no longer required for MIB-2 implementations, is fully supported. The Address translation group is only meaningful in LAN bridging and routing environments. noSuchName is returned for interfaces for which the objects in this group are not meaningful.

IP group (MIB-2 4)

This group is fully supported. The IP group is only meaningful in IP bridging and routing environments. noSuchName is returned for interfaces for which the objects in this group are not meaningful.

ipForwarding (ip 1) read-write

The indication of whether this Ascend unit is acting as an IP gateway in respect to the forwarding of datagrams received by, but not addressed to, this entity. IP gateways forward datagrams. IP hosts do not (except those source-routed via the host).

- forwarding(1) acting as a gateway
- not-forwarding(2) *not* acting as a gateway

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

ipDefaultTTL (ip 2) read-write

The default value inserted into the Time-To-Live field of the IP header of datagrams originating from this entity whenever a TTL value is not supplied by the transport layer protocol. For all relevant Ascend products' interfaces, ipDefault is 64 seconds. Some implementations interpret the seconds as hops, where one hop = 1 second. Allowable values in a set operation are between 1 and 255.

ipInReceives, (ip3) read-only

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

ipInHdrErrors, (ip4) read-only

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

ipInAddrErrors, (ip5) read-only

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

ipForwDatagrams, (ip6) read-only

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

ipInUnknownProtos, (ip7) read-only

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

ipInDiscards, (ip8) read-only

The number of input IP datagrams for which no problems were encountered to prevent their continued processing, but which were discarded (e.g., for lack of buffer space). Note that this counter does not include any datagrams discarded while awaiting re-assembly.

ipInDelivers, (ip9) read-only

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

ipOutRequests, (ip10) read-only

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

ipOutDiscards, (ip11) read-only

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

ipOutNoRoutes, (ip12) read-only

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

ipReasmTimeout, (ip13) read-only

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

ipReasmReqds, (ip14) read-only

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

ipReasmOKs, (ip15) read-only

The number of IP datagrams successfully re-assembled.

ipReasmFails, (ip16) read-only

The number of failures detected by the IP re-assembly algorithm (for whatever reason: timed out, errors, etc). Note that this is not necessarily a count of discarded IP fragments since some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received.

ipFragOKs, (ip17) read-only

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

ipFragFails, (ip18) read-only

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

ipFragCreates (ip19) read-only

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

ipAddrTable (ip 20) not-accessible

The IP address table.

ipAddrEntry (ipAddrTable 1) not-accessible

IP address table entries.

ipAdEntAddr, (ipAddrEntry 1) read-only

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

ipAdEntIfIndex, (ipAddrEntry 2) read-only

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

ipAdEntNetMask, (ipAddrEntry 3) read-only

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

ipAdEntBcastAddr, (ipAddrEntry 4) read-only

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

ipAdEntReasmMaxSize (ipAddrEntry 5) read-only

The size of the largest IP datagram which this entity can re-assemble from incoming IP fragmented datagrams received on this interface.

ipRouteTable (ip 21) not-accessible

The IP routing table.

ipRouteEntry, not-accessible

IP routing table entries.

ipRouteDest, (ipRouteEntry 1) read-write

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

ipRouteIfIndex. (ipRouteEntry 2) read-write

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

ipRouteMetric1, (ipRouteEntry 3) read-write

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

ipRouteMetric2, (ipRouteEntry 4) read-write

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

ipRouteMetric3, (ipRouteEntry 5) read-write

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

ipRouteMetric4, (ipRouteEntry 6) read-write

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

ipRouteNextHop, (ipRouteEntry 7) read-write

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

ipRouteType, (ipRouteEntry 8) read-write

The type of route.

- other(1) none of the following
- invalid(2) an invalidated route
- direct(3) route to directly connected (sub-)network
- indirect(4) route to a non-local host/network/sub-network

Note that the values direct(3) and indirect(4) refer to the notion of direct and indirect routing in the IP architecture.

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

ipRouteProto, (ipRouteEntry 9) read-write

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

- other(1): none of the following
- local(2): non-protocol information, e.g., manually configured entries
- netmgmt(3): set via a network management protocol
- icmp(4): obtained via ICMP, e.g., Redirect
- egp(5)

The remaining values are all gateway routing protocols: ggp(6), hello(7), rip(8), is-is(9), es-is(10), ciscoIgrp(11), bbnSpfIgp(12), ospf(13), bgp(14).

ipRouteAge, (ipRouteEntry 10) read-write

The number of seconds since this route was last updated or otherwise determined to be correct. Note that no semantics of "too old" can be implied except through knowledge of the routing protocol by which the route was learned. Ascend products uses the value 9999 to indicate "infinity" for ipRouteAge.

ipRouteMask, (ipRouteEntry 11) read-write

Indicate the mask to be logical-ANDed with the destination address before being compared to the value in the ipRouteDest field. For those systems that do not support arbitrary subnet masks, an agent constructs the value of the ipRouteMask by determining whether the value of the correspondent ipRouteDest field belong to a class-A, B, or C network, and then using the default netmask.

If the value of the ipRouteDest is 0.0.0.0 (a default route), then the mask value is also 0.0.0.0. It should be noted that all IP routing subsystems implicitly use this mechanism.

ipRouteMetric5 (ipRouteEntry 12) read-write

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

ipRouteInfo (ifRouteEntry 13) read-only

Ascend products do not support any MIBs specific to a particular routing protocol, for any of its products. The returned value of ipRouteInfo is always noSuchName.

ipNetToMediaTable (ip 22) not-accessible

The IP Address Translation table used for mapping from IP addresses to physical addresses.

ipNetToMediaEntry (ipNetToMediaTable 1) not-accessible

IP address translation table entries.

ipNetToMediaIfIndex, (ipNetToMediaEntry 1) read-write

Returns a value indicating a route type. Ascend products supports the following values:

- dynamic(3)
- static(4)

ipNetToMediaPhysAddress, (ipNetToMediaEntry 2) read-write

The media-dependent physical address.

ipNetToMediaNetAddress, (ipNetToMediaEntry 3) read-write

The IpAddress corresponding to the media-dependent physical address.

ipNetToMediaType (ipNetToMediaEntry 4) read-write

- other(1): none of the following
- invalid(2): an invalidated mapping
- dynamic(3): The type of mapping.
- static(4): The type of mapping.

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

ipRoutingDiscards (ip 23) read-only

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

ICMP group (MIB-2 5)

This group is fully supported. The ICMP group is only meaningful in LAN bridging and routing environments. noSuchName is returned for interfaces for which the objects in this group are not meaningful.

TCP group (MIB-2 6)

This group is not currently supported on Ascend products. Requests for values in this group result in noSuchName.

UDP group (MIB-2 7)

This group is fully supported in Ascend products.

udpInDatagrams (udp 1) read-only

The total number of UDP datagrams delivered to UDP users.

udpNoPorts (udp 2) read-only

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

udpInErrors (udp 3) read-only

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

udpOutDatagrams (udp 4) read-only

The total number of UDP datagrams sent from this entity.

udpTable (udp 5) not-accessible

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

udpEntry (udpTable 1) not-accessible

Information about a particular current UDP listener.

udpLocalAddress (udpEntry 1) read-only

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

udpLocalPort (udpEntry 2) read-only

The local port number for this UDP listener.

EGP group (MIB-2 8)

This group is not currently supported on Ascend products. Requests for values in this group result in noSuchName.

Transmission group (MIB-2 10)

This group is a placeholder in the MIB tree to link related MIBs. Only the DS1 and RS232 MIBs are linked to this branch; that is, the Transmission group has a value 1.3.6.2.1.10 with a DS1 branch (1.3.6.2.1.10.18) and a RS232 branch (1.3.6.2.1.10.33).

SNMP group (MIB-2 11)

This group is fully supported in Ascend products.

snmpInPkts (snmp 1) read-only

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

snmpOutPkts (snmp 2) read-only

The total number of SNMP Messages which were passed from the SNMP protocol entity to the transport service.

snmpInBadVersions (snmp 3) read-only

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

snmpInBadCommunityNames (snmp 4) read-only

The total number of SNMP Messages delivered to the SNMP protocol entity which used a SNMP community name not known to said entity.

snmpInBadCommunityUses (snmp 5) read-only

The total number of SNMP Messages delivered to the SNMP protocol entity which represented an SNMP operation which was not allowed by the SNMP community named in the Message.

snmpInASNParseErrs (snmp 6) read-only

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

snmpInTooBigs (snmp 8) read-only

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

snmpInNoSuchNames (snmp 9) read-only

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

snmpInBadValues (snmp 10) read-only

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

snmpInReadOnlys (snmp 11) read-only

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

snmpInGenErrs (snmp 12) read-only

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

snmpInTotalReqVars (snmp 13) read-only

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

snmpInTotalSetVars (snmp 14) read-only

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

snmpInGetRequests (snmp 15) read-only

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

snmpInGetNexts (snmp 16) read-only

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

snmpInSetRequests (snmp 17) read-only

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

snmpInGetResponses (snmp 18) read-only

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

snmpInTraps (snmp 19) read-only

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

snmpOutTooBigs (snmp 20) read-only

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

snmpOutNoSuchNames (snmp 21) read-only

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

snmpOutBadValues (snmp 22) read-only

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

snmpOutGenErrs (snmp 24) read-only

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

snmpOutGetRequests (snmp 25) read-only

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

snmpOutGetNexts (snmp 26) read-only

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

snmpOutSetRequests (snmp 27) read-only

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

snmpOutGetResponses (snmp 28) read-only

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

snmpOutTraps (snmp 29) read-only

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

snmpEnableAuthenTraps (snmp 30) read-write

The value enabled(1) or disabled(2) indicates whether the SNMP agent process is permitted to generate authentication-failure traps. The value of this object overrides any configuration information; as such, it provides a means whereby all authentication-failure traps may be disabled.

DS1 MIB implementation (RFC 1406)

RFC-1406 defines the manageable items in a DS1 interface. The MIB is composed of a Near End group, a Far End group, and a Fractional group. The Near End group and the Fractional group are supported. Ascend products do not have the ability to query the far end for link statistics, so the Far End group is not supported. Requests for items that are part of the Far End group return noSuchName.

MIB objects are read-only at this time. All requests to set any variable in the MIB return noSuchName.

Near end group

The following sections discuss the fields supported in the tables of the Near End group.

dsx1ConfigTable (ds1 6) not-accessible

The dsx1ConfigTable is indexed by dsx1LineIndex. Ascend products use the interface number as a line index. The Enterprise MIB slotItemTable can be used to map slot and item (line number) to interface number. All fields of this table are supported.

dsx1CurrentTable (ds1 7) not-accessible

The dsx1CurrentTable is indexed by a dsx1CurrentIndex. Ascend products use the interface number as a current index. The Enterprise MIB slotItemTable can be used to map slot and item (line number) to interface number. The following read-only objects in this table are supported. Requests for values from any other objects in this table return noSuchName.

- dsx1CurrentESs: returns error seconds for current interval.
- dsx1CurrentSESs: returns severely error seconds for current interval.
- dsx1CurrentUASs: returns unavailable seconds for current interval.
- dsx1CurrentCSSs: returns controlled slip seconds for current interval.
- dsx1CurrentBESs: returns bursty error seconds for current interval.

dsx1IntervalTable (ds1 8) not-accessible

The dsx1IntervalTable is doubly indexed by dsx1IntervalIndex and dsx1IntervalNumber. Ascend products use the interface number as an interval index. The Enterprise MIB slotItemTable can be used to map slot and item (line number) to interface number.

The interval number must be in the range of 1 through the current number of intervals defined in the configuration table. The same gauges supported in the current table are supported in the interval table. They are ES, SES, UAS, CSS, an BES. Requests for any other gauge result in noSuchName being returned.

NOTE: An interval is one 15 minute period. The current number of intervals starts at 0 before the Ascend product is turned on and reaches 96 after uninterrupted operation for 24 hours.

dsx1TotalTable not-accessible

The dsx1TotalTable is indexed by a dsx1TotalIndex. The Ascend products use the interface number as the total index. The Enterprise MIB slotItemTable can be used to map slot and item (line number) to interface number. The same gauges supported in the current table are supported in the total table. They are ES, SES, UAS, CSS, an BES. Requests for any other gauge result in noSuchName being returned.

Fractional group

The following sections discuss the fields supported in the tables of the Fractional group.

dsx1FracTable (ds1 13) not-accessible

The dsx1FracTable maps the line and channel number to the interface number; that is, you can get the currently assigned number (namely, MIB 2 ifIndex) to the interface using that line and channel. Once you get ifIndex the dsx1FracTable, you can use it to look up its status in the MIB 2 status tables or in the Enterprise MIB's slotIfTable.

Note the following example. In this example, the dsx1FracTable returns the interface number 7 for channel 3 of line 2. The second line of this example, looks up the type of interface (ifType) corresponding to interface number 7 (ifIndex) in the MIB 2 ifTable. In the second line, the value 18 is returned indicating a "ds1" type interface corresponds to interface number 7 (hence channel 3 of line 2):

GET ds1.dsx1FracTable.dsx1FracEntry.dsx1FracIfIndex.2.3 = 7

GET ifTable.ifEntry.ifType.7=1

dsx1FracEntry (dsx1FracTable 1) not-accessible

An entry in the dsx1FracTable.

dsx1FracIndex (dsx1FracEntry 1) read-only

The dsx1FracIndex is the line number, and is identical to the line number index dsx1LineIndex.

dsx1FracNumber (dsx1FracEntry 2) read-only

The dsx1FracNumber is the channel number.

dsx1FracIfIndex (dsx1FracEntry 3) read-only

The dsx1FracIfIndex is the index entry into the MIB 2 ifTable; that is, dsx1FracIfIndex is the value of ifIndex in MIB 2 that corresponds to a line and channel combination. If the channel is not in use, dsx1FracIfIndex returns 0. Note, line and channel combinations can point to the same interface because calls to or from that interface can be composed of several channels possibly from more than one line as in a H0 call (384 kbit/s, 6 channels) from a serial host port. Also note, the line and channel combination does not always point to a serial host port interface (AIM6, Dual Host, or Quad Host) — It can also point to a T1 line (as in a T1-to-PRI call) or to any interface type given by ifDescr in MIB 2.

RS232 MIB implementation (RFC-1317)

RFC-1317 defines the manageable items for RS232 and RS232-like interfaces. Ascend products' serial host ports are similar to synchronous RS232 ports. Ascend products' physical console ports are asynchronous RS232 ports.

The following sections define the MIB objects implemented for the various Ascend products' ports. noSuchName is returned when any non-implemented MIB object is queried. RS232 MIB objects are read-only. Any attempt to set read-only objects results in noSuchName being returned.

rs232Number

The number of ports (regardless of their current state) in the RS-232-like general port table. In the Ascend products implementation, this is the same as the total number of interfaces supported by the unit, however port table entries only exist for serial host and console ports.

rs232PortIndex

A unique value for each port. Its value ranges between 1 and the value of rs232Number. In the Ascend products implementation, this index is the same as the interface number. The Enterprise MIB can be used to map between slot and item and interface number. This value is the same value as contained in ifIndex.

rs232PortType

The port's hardware type. Console ports return the value rs232(2). Serial host ports with V.35 cables installed return the value v35(5). Serial host ports with no cable or RS-449 cables return the value other(1). All other types of interfaces, such as bridge/routing interfaces, WAN ports return noSuchName.

rs232PortInSigNumber

The number of input signals for the port in the input signal table rs232PortInSigTable. The table contains entries only for those signals the software can detect. Serial host ports can detect two input signals, DTR and RTS. Console ports can only detect DTR.

rs232PortOutSigNumber

The number of output signals for the port in the output signal table rs232PortOutSigTable. The table contains entries only for those signals the software can assert. Serial host ports can assert three signals, DSR, CD, and RI. Console ports can not assert any signals.

rs232PortInSpeed

The input speed of the port in bits per second. This is always the same as the output speed.

rs232PortOutSpeed

The output speed of the port in bits per second. This is always the same as the input speed.

rs232AsyncPortIndex

The only asynchronous ports supported are the console ports. Attempting to access this table for any interface other than an interface associated with a console port returns noSuchName.

rs232AsyncPortBits

The Ascend unit's console ports only support 8 bit data.

rs232AsyncPortStopBits

The Ascend unit's console ports only support 1 stop bit.

rs232AsyncPortParity

The Ascend unit's console ports only support none(1) parity.

rs232AsyncPortAutobaud

The Ascend unit's console ports do not support autobaud and always return disabled(2).

rs232AsyncPortParityErrs

Not supported. noSuchName returned on queries.

rs232AsyncPortFramingErrs

Not supported. noSuchName returned on queries.

rs232AsyncPortOverrunErrs

Not supported. noSuchName returned on queries.

rs232SyncPortIndex

A unique value for each synchronous port. Its value is the same as rs232PortIndex for the port. Entries in the rs232SyncPortTable only exist for serial host ports. Indexing this table by the interface number associated with any other type of port returns noSuchName.

rs232SyncPortClockSource

Source of the port's bit rate clock. The Ascend unit's RS232-like ports always report internal(1) because external clock source is not supported.

rs232SyncPortFrameCheckErrs

Not supported. noSuchName returned on queries.

rs232SyncPortTxUnderrunErrs

Not supported. noSuchName returned on queries.

rs232SyncPortRxOverrunErrs

Not supported. noSuchName returned on queries.

rs232SyncPortInterFrames

Not supported. noSuchName returned on queries.

rs232SyncPortAbortedFrames

Not supported. noSuchName returned on queries.

rs232InSigPortIndex

The value of rs232PortIndex for the port to which this entry belongs. This is the first of two indices into the input signal table.

rs232InSigName

The second index into the input signal table is the signal name. As noted above, the only input signals supported on serial host ports are dtr(4) and rts(1). Console ports can only detect dtr(4).

rs232InSigState

The current signal state, either on(2) or off(3).

rs232InSigChanges

Supported. Returns the number of times each associated signal changes state.

rs232OutSigPortIndex

The value of rs232PortIndex for the port to which this entry belongs. This is the first of two indices into the output signal table.

rs232OutSigName

Identification of a hardware signal. Only dsr(3), dcd(6), and ri(5) are supported.

rs232OutSigState

The current signal state, either on(2) or off(3).

rs232OutSigChanges

Supported. Returns the number of times each associated signal changes state.

Frame Relay MIB implementation (RFC-1315)

The MIB is composed of three groups, one defining the Data Link Connection Management Interface (DLCMI), one describing the Circuits, and a third describing errors.

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

DLCMI table

The DLCMI table contains entries only for interfaces that are active; that is, the connection between the entity (MAX or Pipeline) and the frame relay switch must be online. If the connection (switched or nailed-up circuit) goes offline, the entry for that interface is removed from this table.

Note: An interface becomes an entry in the DLCMI table only after you create a Frame Relay Profile for it and its Active parameter has been set to Yes. Setting Active=No removes the interface from the DLCMI table.

Warning: Do not create more than one Frame Relay Profile for an interface. The Frame Relay MIB currently assumes only one DLCMI (frame relay service) exists for a given interface. Creating multiple Frame Relay Profiles for the same interface violates that assumption.

frDlcmiTable (frame-relay 1) not-accessible

The frDlcmiTable holds the parameters for the Data Link Connection Management Interface for the frame relay service on this interface.

frDlcmiEntry (frDlcmiTable 1) not-accessible

The parameters for a particular Data Link Connection Management Interface.

frDlcmiIfIndex (frDlcmiEntry 1) read-only

The ifIndex (the interface as indexed by MIB 2) value of the corresponding ifEntry. See Chapter 4, "Ascend Enterprise MIB," for information about how the ifIndex is related to interfaces on Ascend products. frDlcmiIfIndex points to online frame relay interfaces.

frDlcmiState (frDlcmiEntry 2) read-write

This variable states which Data Link Connection Management scheme is active on the Frame Relay interface (and by implication, what DLCI it uses). This variable can have the following values:

- noLmiConfigured (1)
- ansiT1-617-D (3): ANSI T1.617 Annex D

The values lmiRev1 (2) and ansiT1-617-B (4) are currently not supported.

frDlcmiAddress (frDlcmiEntry 3) read-write

This variable states which address format is in use on the frame relay interface. This variable can have the value q922(4). The values q921 (1), q922March90 (2) and q922November90 (3) are currently not supported

frDlcmiAddressLen (frDlcmiEntry 4) read-write

This variable states which address length in octets. In the case of Q922 format, the length indicates the entire length of the address including the control portion. The value can be two-octets (2). The values three-octets (3) and four-octets (4) are currently not supported

frDlcmiPollingInterval (frDlcmiEntry 5) read-write

This is the number of seconds (5 to 30) between successive status enquiry messages. frDlcmiPollingInterval is the T391 parameter in the MAX/Pipeline menu-driven user interface. Must be less than T392 (parameter in the MAX/Pipeline menu-driven user interface) whenever T392 is not 0.

frDlcmiFullEnquiryInterval (frDlcmiEntry 6) read-write

Number of status enquiry intervals (1 to 255) that pass before issuance of a full status enquiry message. frDlcmiFullEnquiryInterval is the N391 parameter in the MAX/Pipeline menu-driven user interface.

frDlcmiErrorThreshold (frDlcmiEntry 7) read-write

This is the maximum number of unanswered Status Enquiries (1 to 10) the equipment shall accept before declaring the interface down. frDlcmiErrorThreshold is the N392 parameter in the Ascend unit menu-driven user interface. Must be less than or equal to frDlcmiMonitoredEvents.

frDlcmiMonitoredEvents (frDlcmiEntry 8) read-write

This is the number of status polling intervals (1 to 10) over which the error threshold is counted. For example, if within MonitoredEvents number of events the station receives ErrorThreshold number of errors, the interface is marked as down. frDlcmiMonitoredEvents is the N393 parameter in the Ascend unit menu-driven user interface. Must be greater than or equal to frDlcmiErrorThreshold.

frDlcmiMaxSupportedVCs (frDlcmiEntry 9) read-write

The maximum number of Virtual Circuits allowed for this interface. The Ascend unit fixes this variable at the maximum number of Connection Profiles it can support. Usually dictated by the Frame Relay network.

In response to a SET, if a value less than or higher than the The Ascend unit's maximal capability is configured, the Ascend unit responds badValue.

frDlcmiMulticast (frDlcmiEntry 10) read-write

This indicates whether the Frame Relay interface is using a multicast service. Currently supports only nonbroadcast service and broadcast (2) is not supported.

- nonBroadcast (1)

Circuit table

The circuit table describes the use of the Data Link Connection Identifiers (DLCIs) attached to each frame relay interface.

A Frame Relay service is a multiplexing service. DLCIs enumerate virtual circuits (permanent or dynamic) which are layered onto the underlying circuit, represented by ifEntry. Zero or more virtual circuits are layered onto this interface and provide interconnection with various remote destinations. Each such virtual circuit is represented by an entry in this circuit table.

Note: The Ascend unit can multiplex other types of circuits (such as PPP, voice, etc.) beside frame relay virtual circuits over the same interface. In this document, frame relay interface does not imply exclusive use of that interface by DLCIs.

The Circuit table contains entries only for circuits that are active; that is, the connection between the entity (MAX or Pipeline) and the frame relay switch must be online. If the connection (switched or nailed-up circuit) goes offline, the entry for that interface is removed from this table.

Note: A circuit becomes an entry in the Circuit table only after you have created a Frame Relay Profile and Connection Profile for it and have set the Active parameters in each to Yes. Setting Active=No in either profile, removes the circuit from the Circuit table.

frCircuitTable (frame-relay 2) not-accessible

A table containing information about specific DLCIs and corresponding virtual circuits.

frCircuitEntry (frCircuitTable 1) not-accessible

The information regarding a single DLCI.

frCircuitIfIndex (frCircuitEntry 1) read-only

The ifIndex value (the interface as indexed by MIB 2) of the ifEntry this virtual circuit is layered onto. See Chapter 4, "Ascend Enterprise MIB," for information about how the ifIndex is related to interfaces on Ascend products. frCircuitIfIndex points to online frame relay interfaces.

frCircuitDlci (frCircuitEntry 2) read-only

The DLCI for this virtual circuit. frCircuitDlci can have an integer value from 16 to 991. The value returned here is set up by the DLCI parameter in a Connection Profile. Furthermore, that Connection Profile has a FR Prof parameter that points to the Frame Relay Profile which sets up the interface (frCircuitIfIndex) used by the virtual circuit.

frCircuitState (frCircuitEntry 3) read-write

invalid (1) (write-only) Closes circuit and causes this entry to be removed from frCircuitTable. However, if the frame relay interface is nailed-up, the Ascend unit restores this entry to frCircuitTable after resetting this entry's counters (FECNs etc.).

active (2) (read) Circuit is active and end-to-end data transfer is possible. (write) Activates circuit if circuit is inactive. Updates last change time.

inactive (3) (read) Circuit is inactive. (write) Deactivates circuit if circuit is active. Updates last change time. Does not change the setting of the Active parameter in the Connection Profile.

Note: If your are running Annex D, initially, for a brief period of time after the frame relay interface goes online, the DLCI are not active.

frCircuitReceivedFECNs (frCircuitEntry 4) read-only

Number of frames received from the network indicating forward congestion since the virtual circuit was created.

frCircuitReceivedBECNs (frCircuitEntry 5) read-only

Number of frames received from the network indicating backward congestion since the virtual circuit was created.

frCircuitSentFrames (frCircuitEntry 6) read-only

The number of frames sent from this virtual circuit since it was created.

frCircuitSentOctets (frCircuitEntry 7) read-only

The number of octets sent from this virtual circuit since it was created.

frCircuitReceivedFrames (frCircuitEntry 8) read-only

Number of frames received over this virtual circuit since it was created.

frCircuitReceivedOctets (frCircuitEntry 9) read-only

Number of octets received over this virtual circuit since it was created.

frCircuitCreationTime (frCircuitEntry 10) read-only

The value of sysUpTime (MIB 2) when the virtual circuit was created.

frCircuitLastTimeChange (frCircuitEntry 11) read-only

The value of sysUpTime (MIB 2) when last there was a change in the virtual circuit state.

frCircuitCommittedBurst (frCircuitEntry 12) read-write

This variable indicates the maximum amount of data, in bits, that the network agrees to transfer under normal conditions, during the measurement interval. MAX/Pipeline support only the fixed value of 0, which indicates no commitment.

frCircuitExcessBurst (frCircuitEntry 13) read-write

This variable indicates the maximum amount of uncommitted data bits that the network will attempt to deliver over the measurement interval. MAX/Pipeline support a fixed value which is calculated as 64kbit/s * the number of channels in the frame relay service.

For example, suppose a DLCI is set up by Connection Profile "TomsPC" and Frame Relay Profile "WorkGroupA." In the Frame Relay Profile "WorkGroupA", the parameter Nailed Grp=5. Since group 5 in the Line Profile has 6 nailed-up channels, frCircuitExcessBurst=6*64=384. (For switched connections to the frame relay switch, frCircuitExcessBurst is always given by the Data Svc parameter in the Frame Relay Profile.

frCircuitThroughput (frCircuitEntry 14) read-write

Throughput is the average number of "Frame Relay Information Field" bits transferred per second across a user network interface in one direction, measured over the measurement interval. MAX/Pipeline support only the fixed value of 0, which indicates no commitment.

Error table

The Error Table describes errors encountered on each frame relay interface.

frErrTable (frame-relay 3) not-accessible

A table containing information about errors on the frame relay interface.

frErrEntry (frErrTable 1) not-accessible

The error information for a single frame relay interface.

frErrIfIndex (frErrEntry 1) read-only

The ifIndex value (the interface as indexed by MIB 2) of the ifEntry this virtual circuit is layered onto. See Chapter 4, "Ascend Enterprise MIB," for information about how the ifIndex is related to interfaces on Ascend products. This index points to online frame relay interfaces.

frErrType (frErrEntry 2) read-only

The type of error that was last seen on this interface.

- unknownError(1): Error was detected but isn't any of the following.
- receiveShort(2): Frame too short.
- receiveLong(3): This error is not supported.
- illegalDLCI(4): EA bits don't specify a 2-octet header.
- unknownDLCI(5): Frame relay switch targeted a DLCI which doesn't exist or is currently inactive (frCircuitState=inactive(1)).
- dlcmiProtoErr(6): This error is not supported.
- dlcmiUnknownIE(7): Bad/unknown IE fields.
- dlcmiSequenceErr(8): This error is not supported.
- dlcmiUnknownRpt(9): Unknown/not supported request or report.
- noErrorSinceReset(10): No errors detected since the start of this virtual circuit.

frErrData (frErrEntry 3) read-only

An octet string containing up to 20 bytes of the errored packet. frErrData returns null if no errors detected since start of connection.

frErrTime (frErrEntry 4) read-only

The value of sysUpTime (MIB 2) at which the error was detected. frErrTime returns 0 if no errors detected since start of connection.

Frame relay globals

frame-relay-globals (frame-relay 4)

frTrapState (frame-relay-globals 1) read-write

This variable indicates/sets whether the system produces the frDLCIStatusChange trap. The setting affects all traps for all DLCIs. The value can be enabled(1) or disabled(2).

Note: Setting rfTrapState is not stored in nonvolatile RAM (NVRAM) and resets to enabled(1) whenever the system is reset, such as when switching power-off/power-on.

DLCMI-related traps

frDLCIStatusChange (trap-type 1)

A DLCIStatusChange trap signifies that the unit (MAX or Pipeline) sending the trap recognizes that one of the virtual circuits (to which a DLCI number has been assigned) has changed state; that is, the link has either been created, invalidated, or it has toggled between the active and inactive states.

To identify the virtual circuit that had a status change, this trap binds to the variables frCircuitIfIndex and frCircuitDlci. To identify the state of the virtual circuit, this trap binds to the variable frCircuitState.

Modem MIB implementation (RFC 1696)

This section describes how Ascend units have implemented the Modem MIB (RFC 1696). Pipeline 25/50 and Multiband Plus do not implement this MIB. This implementation is for SNMP version 1 only and includes a subset of the full RFC 1696 recommendations for SNMP version 2.

You can retrieve the Ascend version of the Modem MIB from Ascend's ftp site at ftp.ascend.com. The object names used in the following descriptions are the same as in the Ascend version of the Modem MIB which in some cases differ from RFC 1696.

As a group relative to MIB-2, the Modem MIB is the 38th branch; that is, MIB-2 38.

Modem objects group (mdmMib 1)

The modem objects group contains objects that identify the modems of the managed node, namely the Ascend product. The objects of the modem objects group and their supported values follow.

mdmNumber (mdmMIBObjects 1) read-only

The number of modems present and functional in the Ascend unit. Each modem has a single entry in the modem ID table. If no modems exist, all of the following queries return noSuchName. This value defines the maximum value of the mdmIndex object.

mdmIDTable (mdmMIBObjects 2) not-accessible

This table, called the modem ID table, is base table for the modems managed by this MIB. The mdmLineTable, mdmDTEInterfaceTable, mdmCallControlTable, mdmCCStoredDialStringTable, mdmECTable, mdmDCTable, mdmSCTable, and mdmStatsTable all augment the rows defined in this table.

mdmIDEntry (mdmIDTable 1) not-accessible

One entry exists for each modem managed by the agent. Entries in this table are created only by the agent.

mdmIndex (mdmIDEntry 1) not-accessible

The index to the modem ID table. A unique number for each modem that ranges from 1 to mdmNumber. The value must remain constant at least from one reinitialization of the network management agent to the next. This index is used by all tables in the modem MIB.

mdmIDManufacturerOID (mdmIDEntry 2) read-only

This value is intended to identify the manufacturer, model, and version of this modem. This entry in the modem ID table may be used to identify the existence of enterprise-specific functions and behaviors. Ascend's OID is used: 1.3.6.1.4.1.529. (Reference: V.58 attribute manufacturerID subfield ManufacturerOI)

mdmIDProductDetails (mdmIDEntry 3) read-only

A textual description of this device, including the manufacturer's name, modem model name, hardware revision, firmware revision, and optionally, its serial number. mdmIDProductDetails is an entry in the modem ID table. (Reference: V.58 attribute manufacturerID subfield productDetails.)

get ...mdmIDProductDetails... returns one of the two following values depending on which modem is installed: (The V34 modem is available only on the MAX models.)

- Ascend Comm. Inc., V32bis digital modem
- Ascend Comm. Inc., V34 digital modem

mdmLineTable (mdmMIBObjects 3) not-accessible

This table, called the modem Line Interface table, augments the modem ID table.

mdmLineEntry (mdmLineTable 1) not-accessible

One entry exists for each modem managed by the agent, and since each modem interfaces to only one channel, one entry exists for each channel interfaced. The index to the modem Line Interface table is mdmIndex. Entries in this table are created only by the agent.

mdmLineCarrierLossTime (mdmLineEntry 1) read-write

Duration in 10ths of a second the modem waits after loss of carrier before hanging up. The Ascend unit fix this object at 14 (that is, 1.4 seconds). This entry in the modem Line Interface table allows the modem to distinguish between a momentary lapse in line quality and a true disconnect. (Reference: V.58 lineSignalFailDisconnectTimer)

mdmLineState (mdmLineEntry 2) read-write

An entry in the modem Line Interface table that reads or writes the state of the modem's interface to a channel. get ...mdmLineState... can return any of the following values:

- onHook(2): The modem is on hook (that is, not in use).
- offHook(3): The modem is off hook and not connected (read-only value indicating the modem is awaiting RLDS and/or decoding result codes).
- connected(4): The modem is connected (read-only).
- reset(6):The modem is resetting.

mdmLineState can be set to any of the following values:

set ...mdmLineState...onHook(2) causes the call to be hung up and resets the modem.

set ...mdmLineState...busiedOut(5) causes the same action as onHook(2).

set ...mdmLineState...reset(6) the same action as onHook(2).

Any other sets than those listed above result in a badValue error. Statistics are not cleared when the modem is reset.

mdmLineCapabilitiesTable (mdmMIBObjects 4) not-accessible

This table, called the modem Capabilities table, augments the modem ID table. This table lists the protocol capabilities for the modems managed by the agent.

mdmLineCapabilitiesEntry (mdmLineCapabilitiesTable 1) not-accessible

One entry exists for each modem protocol capability for each modem, regardless of whether that protocol is enabled or not. The two-dimensional indices to this table are mdmIndex, mdmLineCapabilitiesIndex.

This table is useful for providing an inventory of the capabilities on a modem, and allowing the manager to enable or disable capabilities from the menu of available possibilities. Row creation is not required to enable or disable capabilities.

mdmLineCapabilitiesIndex (mdmLineCapabilitiesEntry 1) not-accessible

A unique index for this capabilities entry. A maximum of 10 capabilities can be indexed for each modem.

The Ascend unit modems do not support all capabilities. See mdmLineCapabilies 1 through 10 below for which capabilities are supported. If a modem capability is not supported, the modem Capabilities table has no entry for that index. For example, mdmLineCapabiliesBell208 is not supported.

mdmLineCapabilitiesID (mdmLineCapabilitiesEntry 2) read-only

An entry in the modem Capabilities table that identifies one capability of a modem. Standard protocol capabilities will have identifiers registered in this document or other companion standards documents. Proprietary protocol capabilities will be registered by their respective organization. All capabilities, standard or vendor-specific, shall be registered in this table.

mdmLineCapabilitiesEnaRqst (mdmLineCapabilitiesEntry 3) read-write

An entry in the modem Capabilities table that reads or writes the requested configuration of a capability of a modem. This object has the following values:

- disabled(1): disable capability
- perhaps(2): Does not apply to Ascend units
- preferred(3):enable capability

The Ascend unit modem capabilities cannot be changed through SNMP. All modem capabilities are constantly enabled except for the following:

The user interlace can enable/disable V42bis (mdmLineCapabilies 16) and MNP5 (mdmLineCapabilies 21) through the V42/MNP parameter for all digital modems in the unit. For example, looking at modem #1, the set enabling MNP5 (index 10) is allowed if and only if V42/MNP=Yes: set ...mdmLineCapabilitiesEnaRqst.1.10.preferred(3), while set ...mdmLineCapabilitiesEnaRqst.1.10.disabled(1) is not allowed.

The V34 (mdmLineCapabilies 14) is enabled only on V.34 modem modules in the MAX; otherwise, it is disabled.

mdmLineCapabilitiesEnaGrant (mdmLineCapabilitiesEntry 4) read-only

An entry in the modem Capabilities table that reads the actual configuration of this capability.

- disabled(1): capability disabled
- perhaps(2): Does not apply to Ascend units
- preferred(3): capability enabled

mdmLineCapabilities (mdmMIBObjects 5) not-accessible

This branch of the modem MIB includes the modem capabilities returned by the mdmLineCapabilitiesID object. Currently there are 29 capabilities in this list. For example, mdmLineCapabilitesV21, which is the object 1 in the mdmLineCapabilities branch, identifies V.21 capability.

mdmLineCapabilitiesV21 (mdmLineCapabilities 1)

mdmLineCapabilitiesV21 is indexed as the 1st entry in the modem Capabilities table.

mdmLineCapabilitiesV22 (mdmLineCapabilities 2)

mdmLineCapabilitiesV22 is indexed as the 2nd entry in the modem Capabilities table.

mdmLineCapabilitiesV22bis (mdmLineCapabilities 3)

mdmLineCapabilitiesV22bis is indexed as the 3rd entry in the modem Capabilities table.

mdmLineCapabilitiesV32 (mdmLineCapabilities 10)

mdmLineCapabilitiesV32 is indexed as the 4th entry in the modem Capabilities table.

mdmLineCapabilitiesV32bis (mdmLineCapabilities 11)

mdmLineCapabilitiesV32bis is indexed as the 5th entry in the modem Capabilities table.

mdmLineCapabilitiesV34 (mdmLineCapabilities 14)

mdmLineCapabilitiesV34 is indexed as the 6th entry in the modem Capabilities table. Enabled only in the MAX digital modem modules with V.34 option; otherwise disabled.

mdmLineCapabilitiesV42 (mdmLineCapabilities 15)

mdmLineCapabilitiesV42 is indexed as the 7th entry in the modem Capabilities table.

mdmLineCapabilitiesV42bis (mdmLineCapabilities 16)

mdmLineCapabilitiesV42bis is indexed as the 8th entry in the modem Capabilities table. Enabled/disabled by the V42/MNP parameter in the user interface.

mdmLineCapabilitiesMNP4 (mdmLineCapabilities 20)

mdmLineCapabilitiesMNP4 is indexed as the 9th entry in the modem Capabilities table.

mdmLineCapabilitiesMNP5 (mdmLineCapabilities 21)

mdmLineCapabilitiesMNP5 is indexed as the 10th entry in the modem Capabilities table. Enabled/disabled by the V42/MNP parameter in the user interface.

mdmDTEInterfaceTable (mdmMIBObjects 6) not-accessible

This table, called the modem DTE Interface table, augments the modem ID table.

mdmDTEInterfaceEntry (mdmDTEInterfaceTable 1) not-accessible

One entry exists for each modem managed by the agent, and since each modem interfaces to only one DTE, one entry exists for each DTE interfaced. The index to the modem Line Interface table is mdmIndex. Entries in this table are created only by the agent.

mdmDTEActionDTROnToOff (mdmDTEInterfaceEntry 1) read-write

The action the Ascend unit modem takes when DTR drops is ignore(1) and cannot be changed through SNMP.

- ignore(1): The modem takes no action when DTR drops.
- escapeToCommandMode(2): Does not apply to Ascend units' modems.
- disconnectCall(3): Does not apply to Ascend units' modems.
- resetModem(4): Does not apply to Ascend units' modems.

set ...mdmDTEActionDTROnToOff...ignore(1) is allowed, while all other values return badValue. The command get ...mdmDTEActionDTROnToOff... always returns ignore(1).

mdmDTEActionDTROffToOn (mdmDTEInterfaceEntry 2) read-write

The action the Ascend unit modem takes when DTR is raised is ignore(1) and cannot be changed through SNMP.

- ignore(1): The modem takes no action when DTR is raised.
- enableDial(2): Does not apply to Ascend units' modems.
- autoAnswerEnable(3): Does not apply to Ascend units' modems.
- establishConnection(4): Does not apply to Ascend units' modems.

set ...mdmDTEActionDTROffToOn...ignore(1) is allowed, while all other values return badValue. The command get ...mdmDTEActionDTROffToOn... always returns ignore(1).

mdmDTESyncTimingSource (mdmDTEInterfaceEntry 3) read-write

The clock source for synchronous transmissions is external(2) and cannot be changed through SNMP. The Ascend unit modems operate only in the asynchronous mode and this object has no effect.

- internal(1): Does not apply to Ascend units' modems.
- external(2): The transmit clock signals are provided by the DTE.
- loopback(3): Does not apply to Ascend units' modems.
- network(4): Does not apply to Ascend units' modems.

set ...mdmDTESyncTimingSource...external(2) is allowed, while all other values return badValue. The command get ...mdmDTESyncTimingSource... always returns external(2).

mdmDTESyncAsyncMode (mdmDTEInterfaceEntry 4) read-write

The operational mode of the modem. The Ascend unit modems operate only in the asynchronous mode and cannot be changed through SNMP.

- async(1): The Ascend unit modems are fixed at this value.
- sync(2): Does not apply to Ascend units.
- syncAfterDial(3): Does not apply to Ascend units.

set ...mdmDTESyncAsyncMode...async(1) is allowed, while all other values return badValue. The command get ...mdmDTESyncAsyncMode... always returns async(1).

mdmDTEInactivityTimeout (mdmDTEInterfaceEntry 5) read-write

The amount of idle time in minutes that the modem will wait before disconnecting a connection. The Ascend unit modems are fixed at the value 0, and no idle disconnect occurs.

set ...mdmDTEInactivityTimeout...0 is allowed, while all other values return badValue. The command get ...mdmDTEInactivityTimeout... always returns 0.

mdmCallControlTable (mdmMIBObjects 7) not-accessible

This table, called the modem Call Control table, augments the modem ID table.

mdmCallControlEntry (mdmCallControlTable 1) not-accessible

One entry exists for each modem managed by the agent. The index to the modem Line Interface table is mdmIndex. Entries in this table are created only by the agent.

mdmCCRingsBeforeAnswer (mdmCallControlEntry 1) read-write

Determines which ring the modem will wait to answer the phone on. The Ascend unit modems are fixed at the value 0, and the modem does not go offhook to answer a call when a ring signal is detected. When MAX/Pipeline determine an incoming call should be answered by a modem, the modem is commanded to answer immediately. (Reference: ringsBeforeAnswer)

set ...mdmCCRingsBeforeAnswer...0 is allowed, while all other values return badValue. The command get ...mdmCCRingsBeforeAnswer... always returns 0.

mdmCCCallSetUpFailTimer (mdmCallControlEntry 2) read-write

This parameter specifies the amount of time, in seconds, that the modem shall allow between either answering a call (automatically or manually) or completion of dialing, and establishment of a connection with the remote modem. If no connection is established during this time, the modem disconnects from the line and returns a result code indicating the cause of the disconnection. The Ascend unit modems are fixed at the value 50 seconds. (Reference: V.58 callSetUpFailTimer)

set ...mdmCCCallSetUpFailTimer...50 is allowed, while all other values return badValue. The command get ...mdmCCCallSetUpFailTimer... always returns 50.

mdmCCResultCodeEnable (mdmCallControlEntry 3) read-write

The Ascend unit modems always return error codes in numeric format, numericEnabled(2) and this action cannot be changed through SNMP. (Reference: V.58 responseModeSelect)

- disabled(1): Does not apply to Ascend units.
- numericEnabled(2): The DCE issues result codes in numeric form.
- verboseEnabled(3): Does not apply to Ascend units.

set ...mdmCCResultCodeEnable...numericEnabled(2) is allowed, while all other values return badValue. The command get ...mdmCCResultCodeEnable... always returns numericEnabled(2).

mdmCCEscapeAction (mdmCallControlEntry 4) read-write

The Ascend unit modems ignore the "escape to command mode" character sequence. This behavior cannot be changed by through SNMP.

- ignoreEscape(1): The Ascend unit modems are fixed at this value.
- hangUp(2): Does not apply to Ascend units.
- enterCommandMode(3): Does not apply to Ascend units.

set ...mdmCCEscapeAction...ignoreEscape(1) is allowed, while all other values return badValue. The command get ...mdmCCEscapeAction... always returns ignoreEscape(1).

mdmCCCallDuration (mdmCallControlEntry 5) read-only

Present or last completed connection time in seconds. If there have been no previous connections, this value should be -1.

mdmCCConnectionFailReason (mdmCallControlEntry 6) read-only

Indicates the reason that the last connection or attempt failed. The reason code for the last failure is not cleared upon a successful connection. The meaning of each reason code is explained below.

- unknown(1): This code means there has been no previous call.
- other(2): This code used when no other code is applicable.
- managementCommand(3): SNMP manager or other application level causes (such as disabling a Connection Profile) terminated the call.
- inactivityTimeout(4): Does not apply to Ascend units.

- mnpIncompatibility(5): Does not apply to Ascend units.
- protocolError(6): Does not apply to Ascend units.
- powerLoss(10): Does not apply to Ascend units.
- equipmentFailure(11): Does not apply to Ascend units.
- dtrDrop(20): Does not apply to Ascend units.
- noDialTone(30): Does not apply to Ascend units.
- lineBusy(31): Does not apply to Ascend units.
- noAnswer(32): Does not apply to Ascend units.
- voiceDetected(33): Does not apply to Ascend units.
- carrierLost(40): Indicates that the modem has disconnected due to detection of loss of carrier. DCD (data carrier detect) dropped.
- trainingFailed(41): Indicates that DCD (data carrier detect) failed to go active or that the modem failed to decode result codes
- faxDetected(42): Does not apply to Ascend units.

mdmCCStoredDialStringTable (mdmMIBObjects 8) not-accessible

This table, called the modem Dial String table, augments the modem ID table. The table of stored dial strings. (Reference: V.58 telephoneNumbers)

mdmCCStoredDialStringEntry (mdmCCStoredDialStringTable 1) not-accessible

A stored dial string. Two entries exist for each modem managed by the agent. The two-dimensional index to the modem Line Interface table is mdmIndex, mdmCCStoredDialStringIndex. Entries in this table are created only by the agent.

mdmCCStoredDialStringIndex (mdmCCStoredDialStringEntry 1) not-accessible

The unique index of a particular dial string. This index accepts only the values 1 and 2.

mdmCCStoredDialString (mdmCCStoredDialStringEntry 2) read-write

A dial string stored in the modem. Ascend unit dial strings are constant and cannot be changed by SNMP. Any attempt to set this object to a value different from what is stored in the modem results in a badValue error.

If mdmCCStoredDialStringIndex=1, the string corresponds to the V42/MNP disabled mode (V42/MNP parameter set to No). If mdmCCStoredDialStringIndex=2, the string corresponds to the V42/MNP enabled mode (V42/MNP parameter set to Yes).

mdmECTable (mdmMIBObjects 9) not-accessible

This table, called the modem Error Correcting table, augments the modem ID table.

mdmECEntry (mdmECTable 1) not-accessible

One entry exists for each modem managed by the agent. The index to the this table is mdmIndex. Entries in this table are created only by the agent.

mdmECErrorControlUsed (mdmECEntry 1) read-only

Indicates the error control method used during the current or previous call. This shall be one of the values for error control protocols registered in the capabilities table for this modem. If no error control protocol is in use, this object shall have the value {0 0}. (Reference: V.58 errorControlActive)

Since Ascend unit modems support only the V42 LAPM and MNP4 error control modes, get ...mdmECErrorControlUsed... can return the following values:

- mdmLineCapabilitiesV42
- mdmLineCapabilitiesMNP4
- {0.0} (the modem at the far end uses no error control, or there is no previous call)

mdmDCTable (mdmMIBObjects 10) not-accessible

This table, called the modem data compression table, augments the modem ID table.

mdmDCEntry (mdmDCTable 1) not-accessible

Entries in this table are created only by the agent. One entry exists for each modem managed by the agent.

mdmDCCompressionTypeUsed (mdmDCEntry 1) read-only

Indicates the data compression method used during the current or previous call. This shall be one of the values for compression protocols registered in the capabilities table for this modem. If no compression protocol is in use, this object shall have the value {0 0}.

Since Ascend unit modems support only the V42bis and MNP5 compression modes, get ...mdmDCCompressionTypeUsed... can return the following values:

- mdmLineCapabilitiesV42bis
- mdmLineCapabilitiesMNP5
- {0.0} (no compression, or there is no previous call)

mdmSCTable (mdmMIBObjects 11) not-accessible

The modem signal convertor table augments the modem ID table.

mdmSCEntry (mdmSCTable 1) not-accessible

Entries in this table are created only by the agent. One entry exists for each modem managed by the agent.

mdmSCCurrentLineXmitRate (mdmSCEntry 1) read-only

The current link transmit rate of a connection, or the last link transmit rate of the last connection in bits per second. This object is set to 0 at startup. (Reference: V.58 transmissionSignallingRateActive)

mdmSCCurrentLineRcvRate (mdmSCEntry 2) read-only

The current link receive rate of a connection, or the last link receive rate of the last connection in bits per second. This object is set to 0 at startup. For Ascend unit modems, this object has the same value as mdmSCCurrentLineXmitRate. (Reference: V.58 transmissionSignallingRateActive)

mdmSCInitialLineXmitRate (mdmSCEntry 3) read-only

The initial link transmit rate of the current connection, or the initial link transmit rate of the last connection in bits per second. For Ascend unit modems, this object has the same value as mdmSCCurrentLineXmitRate.

mdmSCInitialLineRcvRate (mdmSCEntry 4) read-only

The initial link receive rate of the current connection, or the initial link receive rate of the last connection in bits per second. This object is set to 0 at startup. For Ascend unit modems, this object has the same value as mdmSCCurrentLineXmitRate.

mdmSCModulationSchemeUsed (mdmSCEntry 5) read-only

The modulation scheme of the current or previous call. This shall be one of the values for modulation protocols registered in the capabilities table for this modem. (Reference: V.58 gstnModulationSchemeActive)

Since Ascend unit modems support only certain modulation schemes, get ...mdmSCModulationSchemeUsed... can return the following values:

- {0.0} (no previous call)
- mdmLineCapabilitiesV21
- mdmLineCapabilitiesV22
- mdmLineCapabilitiesV22bis
- mdmLineCapabilitiesV32
- mdmLineCapabilitiesV32bis
- mdmLineCapabilitiesV34 (requires MAX V34 modem module)

mdmStatsTable (mdmMIBObjects 12) not-accessible

The modem statistics table augments the modem ID table. Unless otherwise noted, the values stored in this table are cumulative and are not cleared except when the Ascend unit resets.

mdmStatsEntry (mdmStatsTable 1) not-accessible

Entries in this table are created only by the agent. One entry exists for each modem managed by the agent.

mdmStatsRingNoAnswers (mdmStatsEntry 1) read-only

The number of events in which the Ascend unit had answered a call and determined which modem should get the call, but could not give the call to that modem due to buffer problems.

mdmStatsIncomingFailures (mdmStatsEntry 2) read-only

The number of events in which the Ascend unit had answered a call and gave the call to that modem, but the call failed because the modem did not detect DCD going active, or the modem could not train with the far end (decode the line rate, determine the error control protocol used, or determine the compression protocol used).

mdmStatsIncomingCompletions (mdmStatsEntry 3) read-only

The number of incoming connection requests that this modem answered and successfully trained with the other DCE.

mdmStatsFailedDialAttempts (mdmStatsEntry 4) read-only

The number of call attempts that failed because the modem didn’t go off hook, or there was no dial tone. This object always has a value 0 because the Ascend unit modems do not dial out.

mdmStatsOutgoingFailures (mdmStatsEntry 5) read-only

The number of outgoing calls from this modem which successfully went off hook and dialed, in which it could not train with the other DCE. This object always has a value 0 because the Ascend unit modems do not dial out.

mdmStatsOutgoingCompletions (mdmStatsEntry 6) read-only

The number of outgoing calls from this modem which resulted in successfully training with the other DCE. This object always has a value 0 because the Ascend unit modems do not dial out.

mdmStatsRetrains (mdmStatsEntry 7) read-only

The number of retrains experienced on connections on this line. This object has a fixed value, 0.

mdmStats2400OrLess (mdmStatsEntry 8) read-only

The number of connections initially established at a modulation speed of 2400 bits per second or less.

mdmStats2400To14400 (mdmStatsEntry 9) read-only

The number of connections initially established at a modulation speed of greater than 2400 bits per second and less than 14400 bits per second.

mdmStatsGreaterThan14400 (mdmStatsEntry 10) read-only

The number of connections initially established at a modulation speed of greater than 14400 bits per second.

mdmStatsErrorControlled (mdmStatsEntry 11) read-only

The number of established connections using an error control protocol.

mdmStatsCompressed (mdmStatsEntry 12) read-only

The number of established connections using a compression protocol.

mdmStatsCompressionEff (mdmStatsEntry 13) read-only

The Ascend unit cannot measure compression efficiency and always return 100. (Reference: V.58 compressionEfficiency)

mdmStatsSentOctets (mdmStatsEntry 14) read-only

The number of octets presented to the modem by the DTE.

mdmStatsReceivedOctets (mdmStatsEntry 15) read-only

The number of octets presented to the DTE by the modem.

mdmStatsSentDataFrames (mdmStatsEntry 16) read-only

The Ascend unit cannot count the data frames sent and always return 0.

mdmStatsReceivedDataFrames (mdmStatsEntry 17) read-only

The Ascend unit cannot count the data frames received and always return 0.

mdmStatsResentFrames (mdmStatsEntry 18) read-only

The Ascend unit cannot count the data frames sent and always return 0.

mdmStatsErrorFrames (mdmStatsEntry 19) read-only

The Ascend unit cannot count the data frames sent or received and always return 0.

L

M