

Lucent Technologies
Bell Labs Innovations



Stinger[®]

SNMP Management of the ATM Stack Supplement

Part Number: 7820-0732-006
For software version 9.3.170
May 2002

Copyright © 2001, 2002 Lucent Technologies Inc. All rights reserved.

This material is protected by the copyright laws of the United States and other countries. It may not be reproduced, distributed, or altered in any fashion by any entity (either internal or external to Lucent Technologies), except in accordance with applicable agreements, contracts, or licensing, without the express written consent of Lucent Technologies. For permission to reproduce or distribute, please email your request to techcomm@lucent.com.

Notice

Every effort was made to ensure that the information in this document was complete and accurate at the time of printing, but information is subject to change.

European Community (EC) RTTE compliance

CE Hereby, Lucent Technologies, declares that the equipment documented in this publication is in compliance with the essential requirements and other relevant provisions of the Radio and Telecommunications Technical Equipment (RTTE) Directive 1999/5/EC.

To view the official *Declaration of Conformity* certificate for this equipment, according to EN 45014, access the Lucent INS online documentation library at <http://www.lucentdocs.com/ins>.

Safety, compliance, and warranty information

Before handling any Lucent Access Networks hardware product, read the *Edge Access and Broadband Access Safety and Compliance Guide* included in your product package. See that guide also to determine how products comply with the electromagnetic interference (EMI) and network compatibility requirements of your country. See the warranty card included in your product package for the limited warranty that Lucent Technologies provides for its products.

Security statement

In rare instances, unauthorized individuals make connections to the telecommunications network through the use of access features.

Trademarks

Lucent, the Lucent logo, and all Lucent brand and product names are trademarks or registered trademarks of Lucent Technologies Inc. Other brand and product names are trademarks of their respective holders.

Ordering information

You can order the most up-to-date product information and computer-based training online at <http://www.lucentdocs.com/bookstore>.

Feedback

Lucent Technologies appreciates customer comments about this manual. Please send them to techcomm@lucent.com.

Customer Service

Product and service information, and software upgrades, are available 24 hours a day. Technical assistance options accommodate varying levels of urgency.

Finding information and software

To obtain software upgrades, release notes, and addenda for this product, log in to Lucent OnLine Customer Support at <http://www.lucent.com/support>.

Lucent OnLine Customer Support also provides technical information, product information, and descriptions of available services. The center is open 24 hours a day, seven days a week. Log in and select a service.

Obtaining technical assistance

Lucent OnLine Customer Support at <http://www.lucent.com/support> provides easy access to technical support. You can obtain technical assistance through email or the Internet, or by telephone. If you need assistance, make sure that you have the following information available:

- Active service or maintenance contract number, entitlement ID, or site ID
- Product name, model, and serial number
- Software version or release number
- Software and hardware options
- If supplied by your carrier, service profile identifiers (SPIDs) associated with your line
- Your local telephone company's switch type and operating mode, such as AT&T 5ESS Custom or Northern Telecom National ISDN-1
- Whether you are routing or bridging with your Lucent product
- Type of computer you are using
- Description of the problem

Obtaining assistance through email or the Internet

If your services agreement allows, you can communicate directly with a technical engineer through Email Technical Support or a Live Chat. Select one of these sites when you log in to <http://www.lucent.com/support>.

Calling the technical assistance center (TAC)

If you cannot find an answer through the tools and information of Lucent OnLine Customer Support or if you have a very urgent need, contact TAC. Access Lucent OnLine Customer Support at <http://www.lucent.com/support> and click **Contact Us** for a list of telephone numbers inside and outside the United States.

Alternatively, call 1-866-LUCENT8 (1-866-582-3688) from any location in North America for a menu of Lucent services. Or call +1 510-769-6001 for an operator. If you do not have an active services agreement or contract, you will be charged for time and materials.

Contents

Customer Service	iii
About This Supplement.....	xiii
Documentation conventions.....	xiii
Stinger documentation set.....	xiv
ITU-T and ANSI standards.....	xv
Structure of management information definitions	xv
Industry-standard documents.....	xvi
Change history	xvii
Chapter 1 Introduction to ATM Components	1-1
ATM stack components	1-1
Physical and ATM interfaces.....	1-2
Physical interfaces in the Stinger unit.....	1-2
ATM interfaces	1-2
ATM interface type 37	1-2
ATM adaptation layer 5 (AAL5) interfaces type 49	1-2
Stacked interfaces	1-3
Virtual links and connections.....	1-3
Virtual links	1-3
ATM PVCs	1-4
ATM soft PVCs (SPVCs).....	1-4
PNNI static route configuration	1-5
PNNI node configuration.....	1-5
PNNI node settings	1-5
Peer group leader election	1-6
Mapping UNI scope to PNNI (not currently supported)	1-6
SVCC-based routing control channel settings (not currently supported).....	1-7
Timing and topology state parameters.....	1-7
Node metrics.....	1-7
PNNI static route configuration.....	1-7
Chapter 2 Typical Stinger Configurations.....	2-1
Network configuration guidelines.....	2-1
Configuring a physical interface.....	2-2
Configuring an ATM interface	2-3
VPI-VCI ranges on ATM interfaces	2-3
Finding the ATM interface ifIndex.....	2-3
Creating traffic descriptors.....	2-4
MIB table overview for traffic descriptor entries	2-4
Default traffic descriptors	2-5

Configuring a descriptor	2-5
Getting the next available index number	2-5
Creating a descriptor entry	2-5
Minimum set of objects for a traffic descriptor	2-6
Configuring an ATM circuit	2-6
Overview of required steps	2-6
MIB table overview for cross-connects	2-7
Creating VCL entries	2-8
Getting the ifIndex values of the circuit's ATM interfaces	2-8
Configuring VCL entries	2-8
Minimum set of objects for a VCL entry	2-9
Creating and activating a virtual circuit cross-connect.....	2-9
Getting the next available index number	2-9
Configuring a cross-connect.....	2-9
Activating the cross-connect	2-9
Deleting cross-connects	2-9
Configuring an SPVC	2-10
Overview of required steps	2-10
MIB table overview for SPVC configuration	2-11
Configuring an ATM address	2-11
Creating a VCL entry for the initiating SPVC.....	2-12
Getting the ifIndex value of the ATM interface for the PVC on the LIM side	2-12
Configuring the VCL entry.....	2-12
Minimum set of objects for an SPVC VCL entry	2-12
Configuring an SPVC entry in the initiating switch	2-13
System-generated entries in the initiating switch	2-13
System-generated entries in the target switch.....	2-14
Configuring the SPVC notification interval	2-15
Configuring a terminating PVC	2-15
MIB table overview for a terminating PVC configuration	2-15
Configuring a terminating connection	2-17
Minimal required objects to create a profile.....	2-17
Sample MibInternetProfileEntry for a terminating PVC.....	2-17
Configuring PNNI routing	2-18
MIB table overview for PNNI node configuration	2-19
Example of a PNNI configuration	2-19
Conditions under which set requests are accepted in PNNI tables.....	2-20

Chapter 3 Management Information Bases (MIBs)..... 3-1

Modifications to standard settings	3-1
Proprietary MIBs for Stinger physical interfaces	3-3
DS1-ATM MIB.....	3-4
DS3-ATM MIB.....	3-7
OC3-ATM MIB	3-9
Switch-Config MIB	3-11
RFC 2515 ATM management objects	3-13
ATM interface configuration	3-13
ATM traffic descriptor configuration	3-14
ATM interface VPL and VCL configuration.....	3-15
ATM cross-connects for PVCs	3-16
AAL5 VCC performance statistics	3-17
SNMP M4 network element view MIB	3-18

ATM Forum Soft PVC MIB	3-18
Managing SPVCs at the originating switch	3-18
SPVC addresses	3-20
Monitoring failing SPVCs	3-20
Supplemental managed objects for ATM management	3-21
ATM interface configuration extensions	3-21
ATM cross-connects for SVCs and SPVCs.....	3-22
ATM interface signaling statistics	3-23
Monitoring active ATM circuits	3-24
Overview of the ATM status MIB.....	3-24
Overview of ATM status variables.....	3-25
ATM status variables.....	3-26
ATM ILMI tables.....	3-27
Currently failing permanent VPLs and VCLs	3-27
ATM Forum PNNI MIB	3-28
PNNI node configuration.....	3-28
PNNI interface configuration.....	3-32
PNNI metrics configuration.....	3-32
PNNI route configuration	3-33
Read-only tables of dynamic link and node information.....	3-35
Deprecated and unsupported PNNI tables	3-39
ATM interface status MIB	3-41
DS3-ATM line status	3-42
E3-ATM line status.....	3-43
OC3-ATM line status.....	3-44
SONET extension MIB	3-48
sonetExtSection group	3-48
sonetExtLine group.....	3-49
sonetExtFarEndLine group	3-49
sonetExtPath group	3-50
sonetExtFarEndPath group	3-51
Appendix A Stinger MIB Load Sequence	A-1
Index.....	Index-1

Figures

Figure 1-1	Interfaces and higher-layer components of the ATM stack.....	1-1
Figure 1-2	Virtual channel links (VCLs) forming a virtual channel connection (VCC).....	1-3
Figure 1-3	Virtual path links (VPLs) forming a virtual path connection (VPC).....	1-3
Figure 1-4	PVCs	1-4
Figure 1-5	SPVCs	1-4
Figure 2-1	slotIfTable entry for a physical interface in slot 17, port 1.....	2-2
Figure 2-2	Locating the ifIndex value for ATM interface ifType 37 (with sample values)	2-4
Figure 2-3	RFC 2515: Traffic descriptor table.....	2-5
Figure 2-4	Example of a trunk-to-trunk ATM circuit with two PVC interfaces	2-6
Figure 2-5	RFC 2515: Cross-connect tables.....	2-7
Figure 2-6	LIM-to-trunk circuit (DSLAM) using an SPVC.....	2-10
Figure 2-7	PNNI Version 1.0 Addendum (Soft PVC MIB) tables.....	2-11
Figure 2-8	Terminating PVC	2-15
Figure 2-9	Lucent Access Internet MIB objects for a terminating PVC	2-16
Figure 2-10	ATM Interface extensions and PNNI Node table	2-19
Figure 2-11	PNNI lowest-level node with its ATM address	2-20

Tables

Table 1-1	Semantics of UNI scope hierarchy	1-6
Table 2-1	Valid network configurations in an SPVC environment	2-2
Table 2-2	Invalid network configurations in an SPVC environment	2-2
Table 2-3	PnniNodeTable conditions for acceptance of set requests	2-20
Table 2-4	PnniNodePglEntry table conditions for acceptance of set requests	2-21
Table 2-5	PnniNodeTimer table conditions for acceptance of set requests	2-21
Table 2-6	PnniNodeSvccEntry table conditions for acceptance of set requests	2-22
Table 2-7	PnniScopeMapping table conditions for acceptance of set requests	2-22
Table 2-8	PnniSummaryAddressTable conditions for acceptance of set requests	2-22
Table 2-9	PnniIfTable conditions for acceptance of set requests	2-23
Table 2-10	PnniMetricsTable conditions for acceptance of set requests	2-23
Table 2-11	PnniRouteAddrTable conditions for acceptance of set requests	2-23
Table 2-12	PnniRouteTnsTable conditions for acceptance of set requests	2-24
Table 3-1	Items in which Stinger units differ from standard MIB defaults	3-1
Table 3-2	mibds1AtmNetworkProfileTable	3-4
Table 3-3	mibds1AtmNetworkProfileTable	3-4
Table 3-4	mibds1AtmNetworkProfileTable	3-6
Table 3-5	mibds1AtmNetworkProfileTable	3-6
Table 3-6	mibds3AtmNetworkProfileTable	3-7
Table 3-7	mibds3AtmNetworkProfileTable	3-8
Table 3-8	mibds3AtmNetworkProfileTable	3-9
Table 3-9	miboc3AtmNetworkProfileTable	3-9
Table 3-10	miboc3AtmNetworkProfileTable	3-10
Table 3-11	miboc3AtmNetworkProfileTable	3-10
Table 3-12	miboc3AtmNetworkProfileTable	3-11
Table 3-13	mibswitchConfigTable	3-11
Table 3-14	mibswitchConfig_atm_parameters_outgoing_shaper table	3-12
Table 3-15	mibswitchConfig_atm_parameters_outgoing_queue table	3-12
Table 3-16	atmInterfaceConfTable	3-13
Table 3-17	atmInterfaceDs3PlcpTable	3-14
Table 3-18	atmInterfaceTCTable	3-14
Table 3-19	atmTrafficDescrParamTable	3-14
Table 3-20	atmVplTable	3-15
Table 3-21	atmVclTable	3-15
Table 3-22	atmVpCrossConnectTable	3-16
Table 3-23	atmVcCrossConnectTable	3-17
Table 3-24	aal5VccTable	3-17
Table 3-25	atmfM4VpNextVpiTable	3-18
Table 3-26	atmfM4VpNextVciTable	3-18
Table 3-27	atmSoftPvcBaseGroup	3-18
Table 3-28	atmSoftPVccTable	3-19
Table 3-29	atmSoftPVpcTable	3-19
Table 3-30	atmInterfaceSoftPvcAddressTable	3-20

Table 3-31	atmCurrentlyFailingSoftPVccTable	3-20
Table 3-32	atmCurrentlyFailingSoftPVpcTable	3-20
Table 3-33	atmInterfaceExtTable	3-21
Table 3-34	atmSvcVpCrossConnectTable	3-23
Table 3-35	atmSvcVcCrossConnectTable	3-23
Table 3-36	atmSigStatTable	3-23
Table 3-37	atmIImiSrvRegTable	3-27
Table 3-38	atmIImiNetworkPrefixTable	3-27
Table 3-39	atmCurrentlyFailingPVplTable	3-28
Table 3-40	atmCurrentlyFailingPVclTable	3-28
Table 3-41	pnniBaseGroup	3-28
Table 3-42	pnniNodeTable	3-29
Table 3-43	pnniNodePglTable	3-29
Table 3-44	pnniNodeTimerTable	3-30
Table 3-45	pnniNodeSvccTable	3-30
Table 3-46	pnniScopeMappingTable	3-31
Table 3-47	pnniSummaryAddressTable	3-31
Table 3-48	pnniIfTable	3-32
Table 3-49	pnniMetricsTable	3-33
Table 3-50	pnniRouteBaseGroup	3-33
Table 3-51	pnniRouteAddrTable	3-33
Table 3-52	pnniRouteTnsTable	3-34
Table 3-53	pnniLinkTable	3-35
Table 3-54	pnniNbrPeerTable	3-36
Table 3-55	pnniNbrPeerPortTable	3-37
Table 3-56	pnniSvccRccTable	3-37
Table 3-57	pnniPtseTable	3-37
Table 3-58	pnniMapTable	3-38
Table 3-59	pnniMapNodeTable	3-38
Table 3-60	pnniMapAddrTable	3-39
Table 3-61	pnniMapTnsTable	3-39
Table 3-62	pnniSummaryTable (deprecated)	3-39
Table 3-63	pnniRouteNodeTable (currently not supported)	3-40
Table 3-64	pnniDTLTable (currently not supported)	3-41
Table 3-65	mibatmIfStatus profile table	3-41
Table 3-66	ds3AtmLineStatus table	3-42
Table 3-67	e3AtmLineStatus profile table	3-43
Table 3-68	oc3AtmLineStatus profile table	3-44
Table 3-69	oc3AtmLineStatus profile table	3-46
Table 3-70	oc3AtmLineStatus profile table	3-47
Table 3-71	sonetExtSectionCurr1DayTable	3-48
Table 3-72	sonetExtSection1DayIntervalTable	3-48
Table 3-73	sonetExtLineCurr1DayTable	3-49
Table 3-74	sonetExtLine1DayIntervalTable	3-49
Table 3-75	sonetExtFarEndLineCurr1DayTable	3-49
Table 3-76	sonetExtFarEndLine1DayIntervalTable	3-50
Table 3-77	sonetExtPathCurr1DayTable	3-50
Table 3-78	sonetExtPath1DayIntervalTable	3-50
Table 3-79	sonetExtFarEndPathCurr1DayTable	3-51
Table 3-80	sonetExtFarEndPath1DayIntervalTable	3-51

About This Supplement

This supplement describes Simple Network Management Protocol (SNMP) management of Asynchronous Transfer Mode (ATM) ports, interfaces, and connections in a Stinger system. Its goal is to provide guidelines for configuring and managing ATM permanent virtual circuits (PVCs), switched virtual circuits (SVCs), and soft PVCs (SPVCs) by means of any SNMP management utility.

To use the information in this supplement, you must have already installed a Stinger unit. If you have not already finished that task, see the unit's *Getting Started Guide*.






Warning: Before installing your Stinger unit, be sure to read the safety instructions in the *Edge Access and Broadband Access Safety and Compliance Guide*. For information specific to your unit, see the “Safety-Related Physical, Environmental, and Electrical Information” chapter in the *Getting Started Guide* for your Stinger unit.

Documentation conventions

Following are the special characters and typographical conventions that might be used in this manual:

Convention	Meaning
Monospace text	Represents text that appears on your computer's screen, or that might appear on your computer's screen.
Boldface monospace text	Represents characters that you enter exactly as shown (unless the characters are also in <i>italics</i> —see <i>Italics</i> , below). If you could enter the characters but are not specifically instructed to, they do not appear in boldface.
<i>Italics</i>	Represent variable information. Do not enter the words themselves in the command. Enter the information they represent. In ordinary text, italics are used for titles of publications, for some terms that would otherwise be in quotation marks, and to show emphasis.
[]	Indicate an optional argument you might add to a command. To include such an argument, type only the information inside the brackets. Do not type the brackets unless they appear in boldface.
	Separates command choices that are mutually exclusive.
>	Points to the next level in the path to a parameter or menu item. The item that follows the angle bracket is one of the options that appear when you select the item that precedes the angle bracket.
:	Separates levels of profiles, subprofiles, and parameters in a pathname displayed in the command-line interface or referred to in text.

Convention	Meaning
Key1+Key2	Represents a combination keystroke. To enter a combination keystroke, press the first key and hold it down while you press one or more other keys. Release all the keys at the same time. (For example, Ctrl+H means hold down the Control key and press the H key.)
Press Enter	Means press the Enter, or Return, key or its equivalent on your computer.
Note:	Introduces important additional information.
 Caution:	Warns that a failure to follow the recommended procedure might result in loss of data or damage to equipment.
 Warning:	Warns that a failure to take appropriate safety precautions might result in physical injury.
 Warning:	Warns of danger of electric shock.

Stinger documentation set

The Stinger documentation set consists of the following manuals, which can be found at <http://www.lucent.com/support> and <http://www.lucentdocs.com/ins>:

- **Read me first:**
 - *Edge Access and Broadband Access Safety and Compliance Guide*. Contains important safety instructions and country-specific information that you must read before installing a Stinger unit.
 - *TAOS Command-Line Interface Guide*. Introduces the TAOS command-line environment and shows you how to use the command-line interface effectively. This guide describes keyboard shortcuts and introduces commands, security levels, profile structure, and parameter types.
- **Installation and basic configuration:**
 - *Getting Started Guide* for your unit. Shows how to install your Stinger chassis and hardware. This guide also shows you how to use the command-line interface to configure and verify IP access and basic access security on the unit, and how to configure Stinger control module redundancy.
 - Module guides for each Stinger line interface module (LIM), trunk module, or other type of module. An individual guide describes the module's features and provides instructions for configuring the module and verifying its status.
- **Configuration:**
 - *Stinger ATM Configuration Guide*. Describes how to use the command-line interface to configure Asynchronous Transfer Mode (ATM) operations on a Stinger unit. The guide explains how to configure permanent virtual circuits (PVCs), and shows how to use standard ATM features such as quality of service (QoS), connection admission control (CAC), and subtending.

- *Stinger Private Network-to-Network Interface (PNNI) Supplement*. Provides quick-start instructions for configuring PNNI and soft PVCs (SPVCs), and describes the related profiles and commands in the Stinger command-line interface.
- *Stinger SNMP Management of the ATM Stack Supplement*. Describes SNMP management of ATM ports, interfaces, and connections on a Stinger unit to provide guidelines for configuring and managing ATM circuits through any SNMP management utility.
- *Stinger T1000 Routing and Tunneling Supplement*. Describes how to configure the Layer 3 routing and virtual private network (VPN) capabilities supported by a Stinger T1000 module.
- *TAOS RADIUS Guide and Reference*. Describes how to set up a TAOS unit to use the Remote Authentication Dial-In User Service (RADIUS) server and contains a complete reference to RADIUS attributes.
- **Administration and troubleshooting:** *Stinger Administration Guide*. Describes how to administer the Stinger unit and manage its operations. Each chapter focuses on a particular aspect of Stinger administration and operations. The chapters describe tools for system management, network management, and Simple Network Management Protocol (SNMP) management.
- **Reference:**
 - *Stinger Reference*. An alphabetic reference to Stinger profiles, parameters, and commands.
 - *TAOS Glossary*. Defines terms used in documentation for Stinger units.

ITU-T and ANSI standards

This supplement does not provide a detailed explanation of architectures or standards developed by other organizations. Following are some publications that you might find useful:

- **ITU-T recommendations:** Recommendations of the International Telecommunication Union Telecommunication Standardization Sector (ITU-T, formerly CCITT) are available commercially. You can order them at <http://www.itu.ch/publications/>.
- **ANSI standards:** The Stinger unit supports Issue 2 of the American National Standards Institute (ANSI) standard T1.413, *Telecommunications—Network and Customer Installation Interfaces—Asymmetric Digital Subscriber Line (ADSL) Metallic Interface*.

Structure of management information definitions

Network management system (NMS) commands access or change information stored in management information bases (MIBs) in the Stinger unit. The MIBs use structure of management information (SMI) definitions for SNMP version 1 (SNMPv1) and SNMP version 2 (SNMPv2).

SNMPv1 SMI is defined in RFC 1155 and is implemented for the SNMP agent in `rfc1155.smi`. For definitions of the SNMPv1 protocol, data types, and MIB tables, see those documents. SNMPv2 SMI is defined in RFC 1212 and RFC 1215, and implemented for the SNMP agent in `rfc1212.smi` and `rfc1215.smi`. The `snmpv2tov1.smi` file defines SNMPv2 SMI used in SNMPv1.

Following are relevant RFCs:

- RFC 1155, *Structure and Identification of Management Information for TCP/IP-based Internets*, May 1990.
- 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 2233, *The Interfaces Group MIB using SMIv2*, November 1997.

Industry-standard documents

Industry organizations such as the Internet Engineering Task Force (IETF) and ATM Forum have specified MIBs that are the basis for SNMP management of the ATM stack.

- **IETF defined MIBs.** The Stinger unit is compliant with the following RFCs and IETF drafts.
 - RFC 2514, *Definitions of Textual Conventions and OBJECT-IDENTITIES for ATM Management*, February 1999.
 - RFC 2515, *Definitions of Managed Objects for ATM Management*, February 1999.
 - draft-ietf-atommib-mib2-13, *Definitions of Supplemental Managed Objects for ATM Management*, March 1998.
- **ATM specifications.** The Stinger unit is compliant with the ATM specifications in the following publications:
 - af-tm-0056.000, *ATM Forum Traffic Management Specification Version 4.0*, April 1996.
 - af-uni-0010.003, *ATM User-Network-Interface (UNI) Signaling Specification Version 4.0*.
 - af-pnni-0055.000, *Private Network-Network Interface Specification Version 1.0 (PNNI 1.0)*, March 1996.
 - af-pnni-0081.000, *PNNI v1.0 Errata and PICS*, May 1997.
 - af-pnni-0066.000, *Private Network-Network Interface Specification Version 1.0 (PNNI 1.0) Addendum (Soft PVC MIB)*, September 1996.
 - af-nm-0095_001_mib.txt, *SNMP M4 Network Element View MIB*, July 1998.

Note: af-ilmi-0065.000, *Integrated Local Management Interface (ILMI) Specification*, is not currently supported.

- **Physical layer MIBs.** The following RFCs define physical layer MIB objects for Stinger trunk interfaces:
 - RFC 1407, *Definitions of Managed Objects for the DS3/E3 Interface Type*, January 1993.
 - RFC 2558, *Definitions of Managed Objects for the SONET/SDH Interface Type*, March 1994.

Change history

November 2000. The default value of the `atmInterfaceConfMinSvccVci` object in the `atmInterfaceExtTable` was corrected from 0 (zero) to 32. See “`atmInterfaceExtTable`” on page 3-21.

January 2001. Editorial changes and addition of guidelines for the number of supported virtual connections. See “Network configuration guidelines” on page 2-1.

March 2001. Editorial changes.

May 2002. Editorial changes and addition of the DS1-ATM, Switch-Config, ATM interface status, and SONET extension MIBs. See “Proprietary MIBs for Stinger physical interfaces” on page 3-3.

Introduction to ATM Components

ATM stack components 1-1

Physical and ATM interfaces 1-2

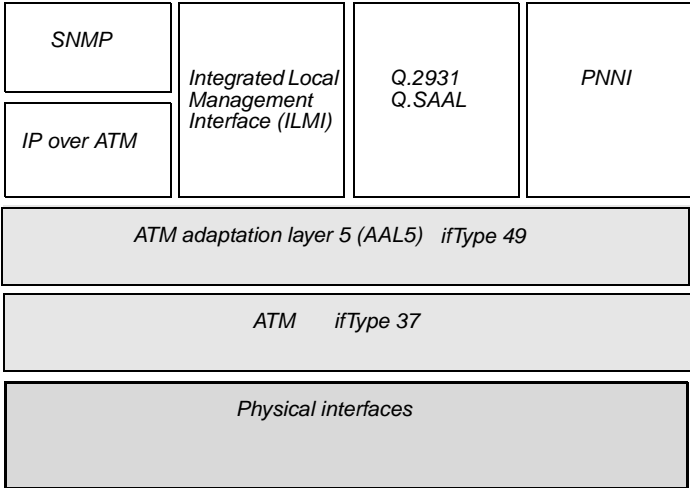
Virtual links and connections 1-3

PNNI static route configuration 1-5

ATM stack components

Figure 1-1 shows the layers of the ATM stack and how the upper layers of the stack interact with the lower-layer physical and logical ports. The shaded items have entries in the interface type (ifType) table. Each type of physical interface in the system has a unique ifType value. For example, OC3 is ifType 39, DS3 is ifType 30, asymmetric DSL (ADSL) is ifType 94, symmetric DSL (SDSL) is ifType 96, and so on.

Figure 1-1. Interfaces and higher-layer components of the ATM stack



- To configure the ATM components of the Stinger unit, you must complete the following steps:
- 1 Configure the physical interfaces.
 - 2 Configure the ATM interface associated with each physical interface.
 - 3 Configure the traffic descriptors.
 - 4 Configure virtual links.
 - 5 Configure permanent virtual circuits (PVCs) or soft PVCs (SPVCs).

Physical and ATM interfaces

Each physical interface is associated with one ATM interface. An `ifIndex` value is used as the instance ID to uniquely identify the interface in the Stinger unit. The index has the same value as the `ifIndex` object defined in RFC 1573, *Evolution of the Interfaces Group of MIB-II*.

Physical interfaces in the Stinger unit

At the lowest layer of the stack, the Stinger unit supports a variety of physical interface types. For details about the MIBs for ATM trunk interfaces, see “Proprietary MIBs for Stinger physical interfaces” on page 3-3.

At system startup, the physical interfaces are configured with default values. After you enable a line by setting the enabled object to `yes` (1) in the appropriate proprietary MIB, the physical interface can become active without further configuration.

ATM interfaces

The following tables contain most of the settings used to configure an ATM interface:

- `atmInterfaceConfTable`, defined in the RFC 2515 MIB. See “ATM interface configuration” on page 3-13.
- `atmInterfaceExtTable`, defined in draft-ietf-atommib-mib2-13. See “Supplemental managed objects for ATM management” on page 3-21.

ATM interface type 37

ATM interface type 37 represents a logical ATM layer on top of a physical port. The Stinger unit switches cells between ATM interfaces via a cross-connect function according to the cells' virtual path identifier-virtual channel identifier (VPI-VCI) values or VPI value alone.

On ATM interfaces of this type, Stinger units support permanent virtual circuits (PVCs), as well as the soft PVCs (SPVCs) and switched virtual circuits (SVCs) used with Private Network-to-Network Interface (PNNI) routing. For more information about PNNI, see “PNNI static route configuration” on page 1-5.

Both Q.2931 and Q.SAAL signaling are supported on the interface with configurable timers and retry values associated with their functionality.

ATM adaptation layer 5 (AAL5) interfaces type 49

Stinger units support AAL5 circuits for packet traffic. SNMP management of the unit and other terminating connections are established on AAL5 interfaces. For details about setting up a terminating PVC on an ATM interface, see “Configuring an SPVC” on page 2-10. For details about configuring the unit to enable SNMP access, see the *Stinger Administration Guide*.

Stacked interfaces

Stinger units support the `ifStackTable`, which defines relationships between the multiple sublayers of a network interface and how interfaces are stacked. For each interface, there is an entry in the `ifTable` with an `ifIndex` value. Each entry in the `ifStackTable` specifies a lower and higher interface index, defining the entry's position in the stack and associating it with relevant interfaces. For example, a physical SDSL port is a physical interface `ifType 96`, which is associated with a physical address (slot number and item number). The port is also associated with an ATM interface (`ifType 37`) at the next layer of the stack, followed by an AAL5 interface (`ifType 49`).

For an example of how to locate associated interfaces, see "Finding the ATM interface `ifIndex`" on page 2-3.

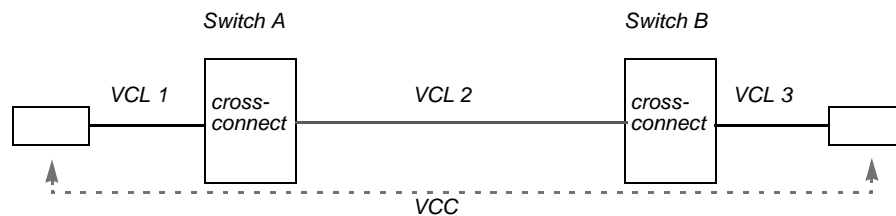
Virtual links and connections

An ATM virtual link is a connection between two switches. Multiple virtual links are concatenated to form ATM virtual connections between two end points, with each concatenation occurring at an ATM switch.

Virtual links

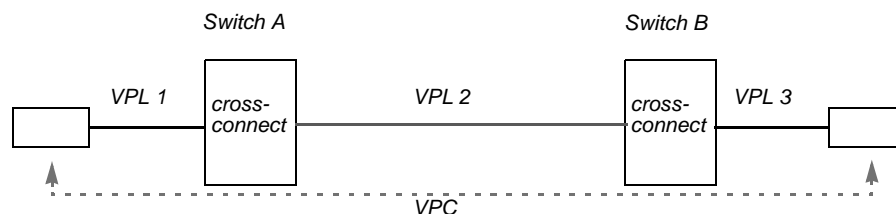
The individual links of a virtual channel connection (VCC) are called virtual channel links (VCLs). The individual links of a virtual path connection (VPC) are called virtual path links (VPLs). For example, Figure 1-2 shows multiple VCLs concatenated to form a VCC.

Figure 1-2. Virtual channel links (VCLs) forming a virtual channel connection (VCC)



Similarly, Figure 1-3 shows multiple VPLs concatenated to form a VPC.

Figure 1-3. Virtual path links (VPLs) forming a virtual path connection (VPC)



A VCL or VPL inherits its traffic characteristics (such as service category) from the VCC or VPC of which it is a part.

You configure virtual links by using the following tables of the ATM MIB, which is defined in `rfc2515.mib`:

- `atmVplTable` (ATM interface virtual path link table)
- `atmVclTable` (ATM interface virtual channel link table)

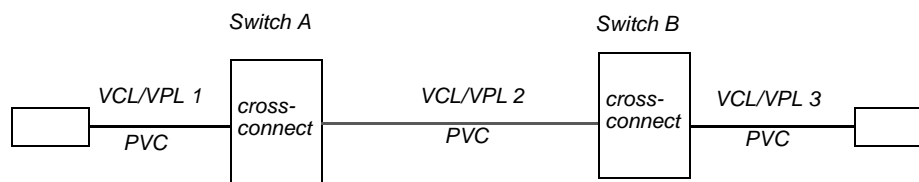
For details about these tables, see “ATM interface VPL and VCL configuration” on page 3-15.

Note: For each virtual link, the `atmVclReceiveTrafficDescriptorIndex` and `atmVclTransmitTrafficDescriptorIndex` objects refer to the entries of the `atmTrafficDescrParamTable` used to specify the traffic descriptor used by the link.

ATM PVCs

ATM permanent virtual circuits (PVCs) require manually configured connections. For an end-to-end PVC, the originating virtual link is cross-connected to another PVC virtual link. From one end point to another, each intervening switch must be manually configured with PVCs. For example, Figure 1-4 shows an end-to-end PVC connection.

Figure 1-4. PVCs



You configure PVCs by using one of the following tables of the ATM MIB, which is defined in `rfc2515.mib`:

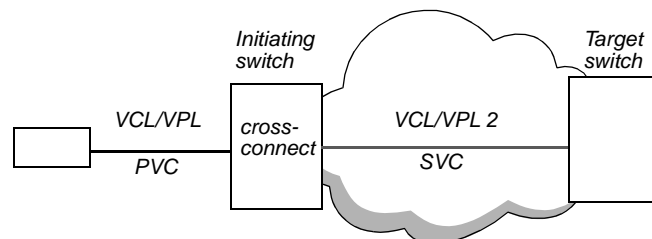
- `atmVcCrossConnectTable` (for virtual channel links)
- `atmVpCrossConnectTable` (for virtual path links)

For details about these tables, see “ATM cross-connects for PVCs” on page 3-16.

ATM soft PVCs (SPVCs)

SPVCs are hybrid connections that are made of a permanent configured part (a PVC) and a part that the switch sets up dynamically (an SVC). The establishment of a connection through the network is done via signaling between switches. An example is shown in Figure 1-5.

Figure 1-5. SPVCs



The switch at one end of an SPVC initiates the signaling to set up the switch-to-switch connection. This switch is known as the `spvcInitiator`. The end of the SPVC that

receives the connection setup is known as the `spvcTarget`. Via network management, the `spvcInitiator` is configured with an ATM address identifying the egress interface from the network. The `spvcInitiator` is responsible for establishing, releasing, and reestablishing the SVC part of the call.

A Stinger unit can be either the `spvcInitiator` or the `spvcTarget` of a given SPVC. The PVC part of an SPVC on a Stinger unit is always the connection from a LIM port to the customer premises equipment (CPE) device. The SVC part of the SPVC is the connection to another switch via a trunk port. The SPVC is initiated when the virtual link at the PVC side of the cross-connect becomes active. After computing the most efficient path to the connection destination, the `spvcInitiator` transmits a signaling request to the `spvcTarget` to set up an SVC.

You configure SPVCs by using one of the following tables, defined in `af-pnni-0066.mib` prepared by the ATM Forum:

- `atmSoftPVccTable` (for virtual channel links)
- `atmSoftPVpcTable` (for virtual path links)

For details about these tables, see “ATM Forum Soft PVC MIB” on page 3-18.

You model and control SPVCs by using one of the `atmSvcVcCrossConnectTable` (for virtual channel links) or the `atmSvcVpCrossConnectTable` (for virtual path links), defined in the supplemental AToM MIB.

PNNI static route configuration

PNNI provides a standards-based dynamic routing protocol for ATM. It allows a group of switches to exchange routing topology information that includes ATM addressing reachability and link and node quality of service (QoS) metrics. The initial release of PNNI on Stinger units supports the minimum subset of PNNI capabilities as specified in Annex G of the ATM Forum PNNI specification. For details, see the *Stinger Private Network-to-Network Interface (PNNI) Supplement*.

PNNI node configuration

When running the PNNI protocol, the Stinger unit performs the functions of a lowest-level node. Configuring a PNNI node includes the following tasks:

- Configuring the node itself
- Configuring Hello protocol timers, topology distribution parameters, and significant change thresholds

PNNI node settings

PNNI node configuration affects the operation of the Stinger unit as a PNNI logical node within a given PNNI peer group. If the unit is eligible to become a member of more than one peer group, you must configure the unit as a node in each group. For details about node configuration, see “`pnniNodeTable`” on page 3-29.

Peer group leader election

A logical node is in the same peer group as other logical nodes when the nodes have the same peer group ID. Within a peer group, each node is assigned and then advertises a peer group leadership priority value. The advertised value is used to elect a peer group leader (PGL), which performs some functions of the logical group node.

Note: With the current software version, the `pnniNodePglLeadershipPriority` object (see “`pnniNodePglTable`” on page 3-29) is always set to 0, so the Stinger unit cannot become peer group leader. However, the unit does participate in the election process.

Mapping UNI scope to PNNI (not currently supported)

ATM User-Network Interface (UNI) Signaling Specification Version 4.0 describes *connection scope* selection, which enables a caller to indicate a routing range for a connection, and *membership scope*, which specifies a routing hierarchy. The scope, which is negotiated during the ILMI client address registration procedure, defines the routing range of a connection without requiring a direct mapping to a network’s routing tables.

The scope values shown in Table 1-1 are used at UNI interfaces (connections to end systems). These values are taken from *ATM User-Network Interface (UNI) Signaling Specification Version 4.0*.

Table 1-1. Semantics of UNI scope hierarchy

UNI scope	Description
Local network	Represents a physical network.
Local plus one Local plus two Site minus one	Levels of granularity between Local and Intrasite scopes for ATM subnetworks that do not include interbuilding or wide-area links.
Intrasite	Identifies an inclusive routing hierarchy that is not geographically separated. This scope confines traffic within a local location to avoid interbuilding or wide-area links.
Intrasite plus one Organization minus one	Levels of granularity between Intrasite and Intraorganization scopes for ATM networks that can include interbuilding links or wide-area links.
Intraorganization	Identifies the inclusive routing hierarchy of an autonomous organization, which is a single administrative authority of the network. The network can include interbuilding and wide-area links.
Organization plus one Community minus one	Levels of granularity between Intraorganization and Intracommunity scopes for ATM networks that include more than one organization.
Intracommunity Community plus one Interregional	Levels of granularity between Intracommunity and Global scopes for collections of autonomous organizations organized by a provider or organizational partnership.
Global	Represents all autonomous organizations that form a connected private ATM network.

Note: With the current software version, the Stinger unit does not support UNI signaling for PNNI. The UNI scope mapping settings (described in “pnniScopeMappingTable” on page 3-31) do not currently apply.

SVCC-based routing control channel settings (not currently supported)

A switched virtual channel connection (SVCC) is a VCC established and taken down dynamically through control signaling. When a PNNI logical link becomes operational, the nodes attached to the link initiate an exchange of information via a well-known VCC used as a PNNI routing control channel (RCC).

Note: With the current software version, the Stinger unit operates only as a lowest-level node. SVCC-RCC settings (described in “pnniSvccRccTable” on page 3-37) do not apply to lowest-level nodes.

Timing and topology state parameters

When a node initializes, it sends a Hello packet on a VCC used as the routing control channel (RCC). The Hello packet contains state and nodal information (such as link status and peer group ID). After the node has started within its peer group, it periodically updates the information in PNNI topology state elements (PTSEs). The updated PTSEs are flooded among the nodes of the peer group by means of PNNI topology state packets (PTSPs), so that the topology database is synchronized for all nodes in the group.

For details about configuring timing, topology state, and significant change settings, see “pnniNodeTimerTable” on page 3-30.

Node metrics

PNNI metrics apply to the connectivity from a node or port to another node or port, or to one or more reachable address prefixes or transit networks, for one or more specific service categories. The `pnniMetricsClasses` object specifies a bit mask in which each bit represents a service category. For example:

Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
CBR	rtVBR	nrtVBR	ABR	UBR

The `pnniMetricsGcacClp` value applies to traffic with a cell loss priority (CLP) of 0 or 0+1. The objects `pnniMetrics1` to `pnniMetrics8` (described in “pnniMetricsTable” on page 3-33) specify traffic characteristics.

PNNI static route configuration

The Stinger currently supports a subset of the PNNI static route tables.

The `pnniRouteAddrTable` and `pnniRouteTnsTable` tables are used to add static routes to reachable address prefixes and transit networks, respectively. For details about these tables, see “pnniRouteAddrTable” on page 3-33 and “pnniRouteTnsTable” on page 3-34.

Introduction to ATM Components

PNNI static route configuration

The `pnniSummaryAddressTable` (described in “`pnniSummaryTable (deprecated)`” on page 3-39) has been deprecated. The PNNI route node (described in “`pnniRouteNodeTable (currently not supported)`” on page 3-40) and the designated transit list (DTL) stack (described in “`pnniDTLTable (currently not supported)`” on page 3-41) are not supported.

Typical Stinger Configurations

Network configuration guidelines	2-1
Configuring a physical interface	2-2
Configuring an ATM interface	2-3
Creating traffic descriptors	2-4
Configuring an ATM circuit	2-6
Configuring an SPVC	2-10
Configuring a terminating PVC	2-15
Configuring PNNI routing	2-18

You configure Stinger ATM components by creating, deleting, or modifying rows in MIB tables. With the current software version, you create a new entry by setting the appropriate RowStatus object to `createAndGo`. Setting RowStatus to `createAndGo` enables the NMS software to configure the entry in one step, by sending a set request that contains all required parameter values. With a one-step procedure, the set request packet contains all of the commands to create a complete entry and set RowStatus to `active`.

Note: With the current software version, setting RowStatus to `createAndWait` for a negotiated configuration procedure is not supported.

The SNMP agent in the Stinger unit allocates resources for configured entities. If the entity is no longer needed (for example, if a VCL is retired), you release associated resources by setting the appropriate RowStatus object to `destroy`.

Network configuration guidelines

To run PNNI, your system must have the Lucent Technologies PNNI software license enabled. The following command verifies that the license has been enabled:

```
admin> get base pnni
[in BASE:pnni-enabled]
pnni-enabled = yes
```

In addition, Lucent Technologies recommends the following virtual circuit and network configuration guidelines:

- In a pure permanent virtual circuit (PVC) environment, the Stinger unit supports up to 5376 active PVCs (14 LIMs × 48 ports maximum per LIM × 8 PVCs per port).

- In an environment that supports both PVCs and soft PVCs (SPVCs), the total virtual circuit limitation is still 5376. The total number of SPVC configurations is limited to 2K within the maximum limitation.
- In an environment that supports both switched virtual circuits (SVCs) and SPVCs, up to 2K SPVCs are recommended, with a maximum of 1000 SPVC initiators. Target SPVCs and initiating SPVCs are supported in the same switch, as long as the total number of SPVCs is no more than 2K and the total number of initiating SPVCs is no more than 1K.
- The Stinger unit supports up to 1500 transit (trunk-to-trunk) calls as well, which contribute to the 2K maximum calls allowed for SPVCs.

Table 2-1 shows examples of valid network configurations in an SPVC environment.

Table 2-1. Valid network configurations in an SPVC environment

Target SPVC	Initiator SPVC	Transit calls	Totals (acceptable)
2000	0	0	2000
0	1000	0	1000
1000	1000	0	2000
1500	300	200	2000

Table 2-2 shows examples of invalid network configurations in an SPVC environment.

Table 2-2. Invalid network configurations in an SPVC environment

Target SPVC	Initiator SPVC	Transit calls	Totals (unacceptable)
1500	1000	0	2500
0	1500	0	1500
1000	1000	1000	3000

Configuring a physical interface

At system startup, the unit's physical interfaces are configured with default values. You must activate the line by setting the enabled object to `yes (2)` in the appropriate proprietary MIB. The line can then become active without any further configuration.

Each physical interface has an entry in the `slotIfTable`. The `slotIfSlotIndex` object specifies the physical slot in which the interface resides, and the `slotIfItemIndex` object identifies the item (port) of the interface. For example, the `slotIfTable` entry for a trunk installed in port 1 of slot 17 has the values shown in Figure 2-1.

Figure 2-1. slotIfTable entry for a physical interface in slot 17, port 1

```
slotIfTable  
slotIfEntry:  
slotIfIndex = 22  
slotIfSlotIndex = 17  
slotIfSlotItemIndex = 1
```

The `slotIfIndex` value identifies an entry in the `ifTable`.

Configuring an ATM interface

An ATM interface is stacked on top of a physical interface such as an OC3, DS3, or xDSL port. The default values for an ATM interface depend on which type of physical device it is associated with.

VPI-VCI ranges on ATM interfaces

For trunk modules, VPIs are 8 bits in length, and by default, VCIs are 13 bits in length. For information about configuring the VPI-VCI range on a trunk module, see the *Stinger ATM Configuration Guide* or “Proprietary MIBs for Stinger physical interfaces” on page 3-3.

For LIMs, the default VPI size is 4 bits with a 7-bit VCI. For information about changing the default assignment, see the *Stinger ATM Configuration Guide*.

Finding the ATM interface ifIndex

An ATM interface is identified by its `ifIndex` value. Any configuration of the ATM interface, including VCL and VPL entries for links that traverse the interface, must specify the proper `ifIndex` value.

The following example demonstrates one useful method for locating the ATM interface `ifIndex` value associated with a particular physical port by using entries in the `ifTable` and `ifStackTable` (`rfc2233.mib`), and `slotIfTable`.

To locate the ATM interface `ifIndex` value associated with port 1 of a trunk module in slot 17:

- 1 Get the `slotIfTable` and find the entry with the following values:

```
slotIfSlotIndex = 17
slotIfItemIndex = 1
```

Note: The `slotIfIndex` of this entry gives the number used as `ifIndex` in the `ifTable`. For example, Figure 2-2 shows `slotifIndex = 9`.

- 2 Get the `ifStackTable` and find the entry in which the `ifStackLowerLayer` corresponds to the `slotIfIndex` for slot 17, port 1.
- 3 Use the `ifStackHigherLayer` value for that entry as the `ifIndex` value for the ATM interface.

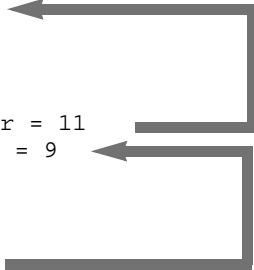
Sample values are shown in Figure 2-2.

Figure 2-2. Locating the `ifIndex` value for ATM interface `ifType 37` (with sample values)

```
IfTable ATM interface (ifType 37)
ifIndex = 11

IfStackTable
ifStackEntry:
ifStackHigherLayer = 11
ifStackLowerLayer = 9

slotIfTable
slotIfEntry:
slotifIndex = 9
slotIfSlotIndex = 17
slotIfSlotItem = 1
```



Creating traffic descriptors

ATM traffic management specifications define service categories for different types of traffic that must have particular characteristics. For example, voice traffic requires a constant amount of bandwidth and cannot tolerate delays, whereas file transfer can tolerate delay and variable bandwidth. ATM switches that reside between two end systems use the ATM service categories and related traffic management settings to alter the characteristics of a cell stream to meet service requirements.

Stinger units support configurable traffic descriptors to specify a service category and define related traffic management parameters. Once you have specified the descriptors you need, you can apply a descriptor to the upstream or downstream traffic (or both) of any number of connections.

To create a traffic descriptor, follow these steps:

- 1 Get the traffic descriptor index by reading the object `atmTrafficDescrParamIndexNext`.
- 2 Create an entry for the traffic descriptor in `atmTrafficDescrParamTable`.

To apply a traffic descriptor to a link, specify the traffic descriptor index in the `atmVplTable` or `atmVclTable` entry. To apply it to incoming traffic, use the `atmVclReceiveTrafficDescrIndex` object. To apply it in the transmit direction, use the `atmVclTransmitTrafficDescrIndex` object.

Note: You must supply both traffic descriptor indexes when creating a VCL or VPL entry. Otherwise, the `set` request fails.

MIB table overview for traffic descriptor entries

RFC 2515, *Definitions of Managed Objects for ATM Management*, includes the traffic descriptor table (`atmTrafficDescrParamTable`) for specifying traffic descriptor entries. The objects in `atmTrafficDescrParamTable` are shown in Figure 2-3.

Figure 2-3. RFC 2515: Traffic descriptor table

```
atmTrafficDescrParamIndexNext
atmTrafficDescrParamTable
  atmTrafficDescrParamEntry
  atmTrafficDescrType
  atmTrafficDescrParamIndex
  atmTrafficDescrType
  atmTrafficDescrParam1
  atmTrafficDescrParam2
  atmTrafficDescrParam3
  atmTrafficDescrParam4
  atmTrafficDescrParam5
  atmTrafficQoSClass
  atmTrafficDescrRowStatus
  atmServiceCategory
  atmTrafficFrameDiscard
```

Default traffic descriptors

Stinger units have a default traffic descriptor (`atmTrafficDescrParamIndex 1`), which defines an unspecified bit rate. It also has two default traffic descriptors for setting up signaling control channels, `atmTrafficDescrParamIndex 2` and `atmTrafficDescrParamIndex 3`.

Configuring a descriptor

A traffic descriptor entry is identified by the traffic descriptor index. To make sure that no fields in the `AtmTrafficDescrParamTable` can be modified when the row is active, remove the row from service by using the following command:

```
SET RowStatus = notInService
```

You can then set objects to define the entry. When the objects have been modified, you can activate the row by using the following command:

```
SET RowStatus = active
```

Getting the next available index number

To obtain the next available traffic descriptor index, issue a get request:

```
GET atmTrafficDescrParamIndexNext
```

Creating a descriptor entry

A traffic descriptor entry is identified by the traffic descriptor index. The index value is shown in bold in the following sample entry. `Index-#` represents the index value obtained from the get request. The value of `atmTrafficDescrType`, which determines the valid settings for `atmTrafficDescrParam1` through `atmTrafficDescrParam5`, must be consistent with `atmServiceCategory`.

```
atmTrafficDescrParamIndex      = Index-#
  atmTrafficDescrType          = consistent with atmServiceCategory
  atmTrafficDescrParam1       = as required by atmTrafficDescrType
```

Typical Stinger Configurations

Configuring an ATM circuit

```
atmTrafficDescrParam2      = as required by atmTrafficDescrType
atmTrafficDescrParam3      = as required by atmTrafficDescrType
atmTrafficDescrParam4      = as required by atmTrafficDescrType
atmTrafficDescrParam5      = as required by atmTrafficDescrType
atmServiceCategory         = as required
atmTrafficFrameDiscard     = true
atmTrafficDescrRowStatus   = createAndGo
```

You can create as many traffic descriptors as you need for supported services. For more details about traffic descriptors and QoS settings in the Stinger unit, see the *Stinger ATM Configuration Guide*.

Minimum set of objects for a traffic descriptor

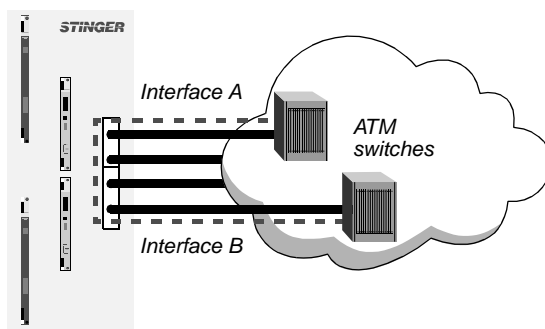
The following setting with a valid `atmTrafficDescrParamIndex` value creates a traffic descriptor that uses default settings:

```
atmTrafficDescrRowStatus = createAndGo
```

Configuring an ATM circuit

An ATM circuit consists of two virtual links that cross-connect. The two virtual links are VCLs or VPLs. Figure 2-4 shows an example of an ATM trunk-to-trunk circuit using VCLs. The system also supports LIM-to-LIM and LIM-to-trunk circuits that use either VCLs or VPLs.

Figure 2-4. Example of a trunk-to-trunk ATM circuit with two PVC interfaces



Note: If a VCL or VPL does not terminate in the unit, you must use the virtual circuit or virtual path cross-connect table to switch a virtual link to a virtual link in another interface. This process is shown in the example entries that follow.

Overview of required steps

Following are the steps for configuring an ATM circuit:

- 1 If necessary, create a traffic descriptor, or select a descriptor to use for the link. (Select two descriptors if the traffic characteristics of the link are asymmetric.)
- 2 Get the `ifIndex` value of the ATM interface for the first side of the circuit. See “Finding the ATM interface `ifIndex`” on page 2-3 for an example.
- 3 Create a VCL entry for the first side of the circuit in `atmVclTable`. (If using VPLs, create an entry in the `atmVplTable`.)

- 4 Get the `ifIndex` value of the ATM interface for the second side of the circuit. See “Finding the ATM interface `ifIndex`” on page 2-3 for an example.
- 5 Create a VCL entry for the second side of the circuit in `atmVclTable`. (If using VPLs, create an entry in the `atmVplTable`.)
- 6 Get the next index of the `VcCrossConnectTable` by reading the object `atmVcCrossConnectIndexNext`.
- 7 Create a virtual circuit cross-connect entry in `atmVcCrossConnectTable`.
- 8 Activate the virtual circuit cross-connect by setting `atmVcCrossConnectAdminStatus` up.

Note: The `atmVclAdminStatus` object is instantiated only for a VCL that terminates a VCC. It is not used for cross-connected links.

MIB table overview for cross-connects

RFC 2515, *Definitions of Managed Objects for ATM Management*, includes the cross-connect tables for specifying ATM circuits shown in Figure 2-5.

Figure 2-5. RFC 2515: Cross-connect tables

atmVcCrossConnectIndexNext

atmVcCrossConnectEntry

```
atmVcCrossConnectIndex
atmVcCrossConnectLowIfIndex
atmVcCrossConnectLowVpi
atmVcCrossConnectLowVci
atmVcCrossConnectHighIfIndex
atmVcCrossConnectHighVpi
atmVcCrossConnectHighVci
atmVcCrossConnectAdminStatus
atmVcCrossConnectL2HOperStatus
atmVcCrossConnectH2LOperStatus
atmVcCrossConnectL2HLastChange
atmVcCrossConnectH2LLastChange
atmVcCrossConnectRowStatus
```

AtmSvcVcCrossConnectEntry

```
atmSvcVcCrossConnectIndex
atmSvcVcCrossConnectLowIfIndex
atmSvcVcCrossConnectLowVpi
atmSvcVcCrossConnectLowVci
atmSvcVcCrossConnectHighIfIndex
atmSvcVcCrossConnectHighVpi
atmSvcVcCrossConnectHighVci
atmSvcVcCrossConnectCreationTime
atmSvcVcCrossConnectRowStatus
```

atmVpCrossConnectIndexNext

atmVpCrossConnectEntry

```
atmVpCrossConnectIndex
atmVpCrossConnectLowIfIndex
atmVpCrossConnectLowVpi
atmVpCrossConnectHighIfIndex
atmVpCrossConnectHighVpi
atmVpCrossConnectAdminStatus
atmVpCrossConnectL2HOperStatus
atmVpCrossConnectH2LOperStatus
atmVpCrossConnectL2HLastChange
atmVpCrossConnectH2LLastChange
atmVpCrossConnectRowStatus
```

AtmSvcVpCrossConnectEntry

```
atmSvcVpCrossConnectIndex
atmSvcVpCrossConnectLowIfIndex
atmSvcVpCrossConnectLowVpi
atmSvcVpCrossConnectHighIfIndex
atmSvcVpCrossConnectHighVpi
atmSvcVpCrossConnectCreationTime
atmSvcVpCrossConnectRowStatus
```

Creating VCL entries

A VCL entry is identified by the ATM interface `ifIndex` value and a valid VPI-VCI pair.

Getting the `ifIndex` values of the circuit's ATM interfaces

For an example of how to obtain the interface index for an ATM interface, see "Finding the ATM interface `ifIndex`" on page 2-3.

Configuring VCL entries

The VCL entry index values are shown in bold in the following sample entry, where `ifIndexA` represents the ATM interface index of the first side of the circuit (Interface A in Figure 2-4). `VpiA` and `VciA` represent VPI and VCI values for the link, `RxTDesc` represents the index of the receive traffic descriptor, and `TxTDesc` represents the index of the transmit traffic descriptor.

```
ifIndex                               = ifIndexA
atmVclVpi                             = VpiA
atmVclVci                             = VciA
  atmVclReceiveTrafficDescrIndex       = RxTDesc
  atmVclTransmitTrafficDescrIndex      = TxTDesc
  atmVccAalType
  atmVccAal5CpcsTransmitSduSize
  atmVccAal5CpcsReceiveSduSize
  atmVccAal5EncapsType
  atmVclCastType
  atmVclRowStatus                       = createAndGo
```

Note: You must supply both traffic descriptor indexes when creating a VCL or VPL entry. Otherwise, the `set` request fails.

In the next sample entry, `ifIndexB` represents the ATM interface index of the second side of the circuit (Interface B in Figure 2-4). `VpiB` and `VciB` represent VPI and VCI values for the link, `RxTDesc` represents the index of the receive traffic descriptor, and `TxTDesc` represents the index of the transmit traffic descriptor.

```
ifIndex                               = ifIndexB
atmVclVpi                             = VpiB
atmVclVci                             = VciB
  atmVclReceiveTrafficDescrIndex       = TxTDesc
  atmVclTransmitTrafficDescrIndex      = RxTDesc
  atmVccAalType
  atmVccAal5CpcsTransmitSduSize
  atmVccAal5CpcsReceiveSduSize
  atmVccAal5EncapsType
  atmVclCastType
  atmVclRowStatus                       = createAndGo
```

Note that for the second interface of the circuit, the traffic descriptors are reversed from their application on the first side of the circuit: the transmit descriptor is applied to the receive direction, and vice versa.

Minimum set of objects for a VCL entry

With valid index values, the following settings create a VCL entry that uses default settings:

```
atmVclReceiveTrafficDescrIndex = TxTDesc
atmVclTransmitTrafficDescrIndex = RxTDesc
atmVclRowStatus = createAndGo
```

TxTDesc and RxTDesc are indexes of an atmTrafficDescrParamEntry with atmTrafficDescrRowStatus set to active.

Creating and activating a virtual circuit cross-connect

A virtual circuit cross-connect entry is identified by the cross-connect index itself, as well as the index values of the two VCLs it connects.

Getting the next available index number

To obtain the next available cross-connect index, issue a get request:

```
GET atmVcCrossConnectIndexNext
```

Configuring a cross-connect

A cross-connect entry is identified by the cross-connect index and the index values of the two VCLs it cross-connects. The indexes are shown in bold in the following sample entry. Index-# represents the index value obtained from the get request. The ifIndexA, VpiA, and VciA values match those in the VCL entry for interface A. The ifIndexB, VpiB, and VciB values match those in the VCL entry for interface B, as described in the preceding sections.

```
atmVcCrossConnectIndex = use Index-#
atmVcCrossConnectLowIfIndex = ifIndexA
atmVcCrossConnectLowVpi = VpiA
atmVcCrossConnectLowVci = VciA
atmVcCrossConnectHighIfIndex = ifIndexB
atmVcCrossConnectHighVpi = VpiB
atmVcCrossConnectHighVci = VciB
atmVcCrossConnectAdminStatus = down
atmVcCrossConnectRowStatus = createAndGo
```

Activating the cross-connect

To make the cross-connect functional, you must activate it as follows:

```
SET atmVcCrossConnectAdminStatus = up
```

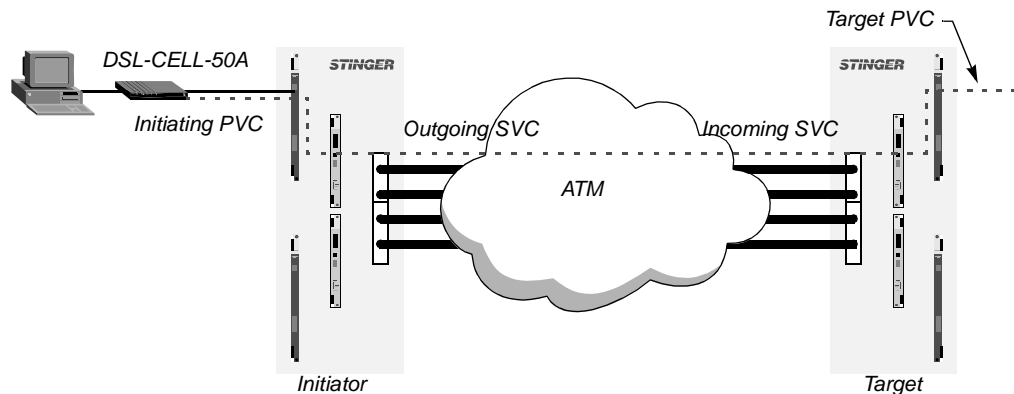
Deleting cross-connects

To release a cross-connect, identify the entry by its index value and set the RowStatus object to destroy. After deleting a cross-connect entry, the SNMP manager can set the RowStatus object to destroy for the associated virtual links. The SNMP agent then releases the resources and removes the associated rows in the atmVclTable or atmVplTable.

Configuring an SPVC

SPVCs are hybrid connections that consist of a permanent configured part (a PVC) and a part that the switch sets up dynamically (an SVC). The PVC part is always the connection from a LIM port to the CPE device. The SVC part is a connection to another ATM switch on a trunk port. Figure 2-6 shows an SPVC that initiates on a LIM port connection to CPE. When the SPVC on the LIM port becomes active, the system brings up a trunk-side SVC.

Figure 2-6. LIM-to-trunk circuit (DSLAM) using an SPVC



Overview of required steps

Following are the steps you perform and those performed by call control procedures to configure an initiating SPVC. To configure an SPVC:

- 1 If necessary, create a traffic descriptor, or select a descriptor to use for the link. (Select two descriptors if the traffic characteristics of the link are asymmetric.)
- 2 Get the `ifIndex` value of the ATM interface for the incoming link. See “Finding the ATM interface `ifIndex`” on page 2-3 for an example.
- 3 Configure the ATM address for the SPVC interface in `atmInterfaceSoftPvcAddressTable`, if necessary. The system generates a default SPVC target address for each ATM interface in the system.
- 4 Create a VCL entry for the PVC in `atmVclTable`. (If using VPLs, create an entry in the `atmVplTable`.)
- 5 Create an entry in `atmSoftPvcTable` for the initiating SPVC.

Note: Call control procedures perform the following steps as part of establishing the SPVC. These entries are not created by the SNMP manager.

- 1 The initiating switch creates a VCL entry for the outgoing SVC in the `atmVclTable`.
- 2 The initiating switch creates cross-connect entry in `atmSvcVcCrossConnectTable`.
- 3 The target switch creates a VCL entry for the incoming SVC in the `atmVclTable`.
- 4 The target switch creates an outgoing VCL entry for the SPVC target (the destination of the connection or another switch along the path).
- 5 The target switch creates a cross-connect entry in `atmSvcVcCrossConnectTable` between the incoming SVC and the outgoing SVC.

MIB table overview for SPVC configuration

The ATM Forum *Private Network-to-Network Interface Specification Version 1.0 Addendum (Soft PVC MIB)*, af-pnni-0066.00, includes the tables for specifying SPVCs shown in Figure 2-7.

Figure 2-7. PNNI Version 1.0 Addendum (Soft PVC MIB) tables

itmSoftPVccTable	atmSoftPVpcTable
atmSoftPVccLeafReference	atmSoftPVpcLeafReference
atmSoftPVccTargetAddress	atmSoftPVpcTargetAddress
atmSoftPVccTargetSelectType	atmSoftPVpcTargetSelectType
atmSoftPVccTargetVpi	atmSoftPVpcTargetVpi
atmSoftPVccTargetVci	atmSoftPVpcLastReleaseCause
atmSoftPVccLastReleaseCause	atmSoftPVpcLastReleaseDiagnostic
atmSoftPVccLastReleaseDiagnostic	atmSoftPVpcOperStatus
atmSoftPVccOperStatus	atmSoftPVpcRestart
atmSoftPVccRestart	atmSoftPVpcRetryInterval
atmSoftPVccRetryInterval	atmSoftPVpcRetryTimer
atmSoftPVccRetryTimer	atmSoftPVpcRetryThreshold
atmSoftPVccRetryThreshold	atmSoftPVpcRetryFailures
atmSoftPVccRetryFailures	atmSoftPVpcRetryLimit
atmSoftPVccRetryLimit	atmSoftPVpcRowStatus
atmSoftPVccRowStatus	
atmInterfaceSoftPvcAddressTable	
atmInterfaceSoftPvcAddress	
atmInterfaceSoftPvcAddressRowStatus	
atmCurrentlyFailingSoftPVccTable	
atmCurrentlyFailingSoftPVccTimeStamp	
atmCurrentlyFailingSoftPVpcTable	
atmCurrentlyFailingSoftPVpcTimeStamp	
TRAPS	
atmSoftPvcCurrentlyFailingSoftPVccs	
atmSoftPvcCurrentlyFailingSoftPVpcs	

Configuring an ATM address

Each trunk port and LIM port that supports ATM has an associated ATM interface. Each ATM interface in the system is assigned a unique SPVC target address, which is used in the signaling required to establish both source and target SPVCs. The SPVC target address is defined in the `atmInterfaceSoftPvcAddressTable`.

The Stinger generates default SPVC target addresses for each ATM interface in the unit. You can also cause the system to generate additional sets of target SPVC addresses if required. For details about how the system generates the default addresses, see the *Stinger Private Network-to-Network Interface (PNNI) Supplement*.

To change a system-generated ATM address for an interface, you modify the appropriate entry in the `atmInterfaceSoftPvcAddressTable`. To make sure that no fields in the `atmInterfaceSoftPvcAddressTable` can be modified when the row is active, remove the row from service by using the following command:

```
SET RowStatus = notInService
```

You can then set objects to define the entry. When the objects have been modified, you can activate the row by using the following command:

```
SET RowStatus = active
```

Following is a sample entry:

```
ifIndex = ifNumber  
atmInterfaceSoftPvcAddress = ATM Address  
atmInterfaceSoftPvcAddressRowStatus = createAndGo
```

Creating a VCL entry for the initiating SPVC

A VCL entry is identified by the ATM interface `ifIndex` value and a valid VPI-VCI pair.

Getting the ifIndex value of the ATM interface for the PVC on the LIM side

For an example of how to obtain the proper interface index for an ATM interface, see “Finding the ATM interface `ifIndex`” on page 2-3.

Configuring the VCL entry

The VCL entry index values are shown in bold in the following sample entry, where `ifIndexLIM` represents the interface index of the LIM port of the initiating PVC (see Figure 2-8). `VpiLIM` and `VciLIM` represent VPI and VCI values for the link, `RxTDesc` represents the index of the receive traffic descriptor, and `TxTDesc` represents the index of the transmit traffic descriptor.

```
ifIndex = ifIndexLIM  
atmVclVpi = VpiLIM  
atmVclVci = VciLIM  
atmVclReceiveTrafficDescrIndex = TxTDesc  
atmVclTransmitTrafficDescrIndex = RxTDesc  
atmVccAalType  
atmVccAal5CpcsTransmitSduSize  
atmVccAal5CpcsReceiveSduSize  
atmVccAal5EncapsType  
atmVclCastType  
atmVclConnKind = spvcInitiator  
atmVclRowStatus = createAndGo
```

Note: The `atmVclConnKind` object must be set to `spvcInitiator`.

Minimum set of objects for an SPVC VCL entry

With valid index values, the following settings create a VCL entry for the initiating PVC:

```
atmVclReceiveTrafficDescrIndex = TxTDesc  
atmVclTransmitTrafficDescrIndex = RxTDesc  
atmVclConnKind = spvcInitiator  
atmVclRowStatus = createAndGo
```

`TxTDesc` and `RxTDesc` are indexes of an active `atmTrafficDescrParamEntry` (an entry in which `atmTrafficDescrRowStatus` is set to `active`).

Configuring an SPVC entry in the initiating switch

Each initiating PVC requires an entry in `atmSoftPVccTable`. You must create the `spvcInitiator` entry in the `atmVclTable` and configure an ATM address for the egress interface before creating the SPVC entry.

To make sure that no fields in the `AtmSoftPVccTable` or `AtmSoftPVpcTable` can be modified when the row is active, remove the row from service by using the following command:

```
SET RowStatus = notInService
```

You can then set objects to define the entry. When the objects have been modified, you can activate the row by using the following command:

```
SET RowStatus = active
```

Following is a sample entry:

```
ifIndex                = ifIndexLIM
atmVclVpi             = VpiLIM
atmVclVci            = VciLIM
  atmSoftPVccLeafReference = arbitrary number
  atmSoftPVccTargetAddress = Target ATM Address
  atmSoftPVccTargetSelectType = required
  atmSoftPVccTargetVpi     = requested target Vpi
  atmSoftPVccTargetVci     = requested target Vci
  atmSoftPVccRestart
  atmSoftPVccRetryInterval
  atmSoftPVccRetryThreshold
  atmSoftPVccRetryLimit
  atmSoftPVccRowStatus     = createAndGo
```

System-generated entries in the initiating switch

When the initiating SPVC becomes active, the system transmits a signaling request to the target switch to create the SVC, and then cross-connects it to the initiating PVC.

Following is a sample VCL entry created by the initiating switch for the outgoing SVC. The system sets the `ifIndex` to the index of the trunk interface used to create the SVC. Notice that `atmVclConnKind` is set to `spvcOutgoing`.

```
ifIndex                = ifIndexOutgoing
atmVclVpi             = VpiOutgoing
atmVclVci            = VciOutgoing
  atmVclReceiveTrafficDescrIndex = nTDesc
  atmVclTransmitTrafficDescrIndex = nTDesc
  atmVccAalType
  atmVccAal5CpcsTransmitSduSize
  atmVccAal5CpcsReceiveSduSize
  atmVccAal5EncapsType          = llcEncapsulation
  atmVclCastType                = p2p
  atmVclConnKind                = spvcOutgoing
  atmVclRowStatus               = active
```

Following is a sample SVC cross-connect entry created by the system for the established SPVC. In the following sample, assume that the `ifIndexLIM` value is lower than the `ifIndexOutgoing` value.

```
atmSvcVcCrossConnectIndex           = allocated by switch
atmSvcVcCrossConnectLowIfIndex      = ifIndexLIM
atmSvcVcCrossConnectLowVpi         = VpiLIM
atmSvcVcCrossConnectLowVci         = VciLIM
atmSvcVcCrossConnectHighIfIndex    = IfIndexOutgoing
atmSvcVcCrossConnectHighVpi        = VpiOutgoing
atmSvcVcCrossConnectHighVci        = VciOutgoing
  atmSvcVcCrossConnectCreationTime
  atmSvcVcCrossConnectRowStatus
```

System-generated entries in the target switch

On the target switch, the signaling request received from the initiator to set up an SVC must be successfully negotiated before the link is established.

Following is a sample VCL entry created by the target switch for the incoming SVC. The system sets the `ifIndex` to the index of the interface used to create the SVC in response to the signaling request from the initiating switch. Notice that `atmVclConnKind` is set to `svcIncoming`.

```
ifIndex                             = ifIndexIncoming
atmVclVpi                           = VpiIncoming
atmVclVci                           = VciIncoming
  atmVclReceiveTrafficDescrIndex     = allocated nTDesc
  atmVclTransmitTrafficDescrIndex    = allocated nTDesc
  atmVccAalType
  atmVccAal5CpcsTransmitSduSize
  atmVccAal5CpcsReceiveSduSize
  atmVccAal5EncapsType               = llcEncapsulation
  atmVclCastType                     = p2p
  atmVclConnKind                     = svcIncoming
  atmVclRowStatus                    = active
```

The target switch must also create a VCL entry for an outgoing SVC on the path to the target destination of the SPVC. Following is a sample VCL entry created by the SPVC target switch for its outgoing SVC. The system sets the `ifIndex` to the index of the interface used to create the outgoing SVC. Notice that `atmVclConnKind` is set to `spvcTarget`.

```
ifIndex                             = ifIndexTarget
atmVclVpi                           = VpiTarget
atmVclVci                           = VciTarget
  atmVclReceiveTrafficDescrIndex     = nTDesc
  atmVclTransmitTrafficDescrIndex    = nTDesc
  atmVccAalType
  atmVccAal5CpcsTransmitSduSize
  atmVccAal5CpcsReceiveSduSize
  atmVccAal5EncapsType               = llcEncapsulation
  atmVclCastType                     = p2p
  atmVclConnKind                     = spvcTarget
  atmVclRowStatus                    = active
```

Following is a sample SVC cross-connect entry created by the target switch to connect its incoming and outgoing SVCs for the established SPVC. In the following sample, assume that the `ifIndexIncoming` value is lower than the `ifIndexTarget` value.

```
atmSvcVcCrossConnectIndex      = allocated by switch
atmSvcVcCrossConnectLowIfIndex = ifIndexIncoming
atmSvcVcCrossConnectLowVpi     = VpiIncoming
atmSvcVcCrossConnectLowVci     = VciIncoming
atmSvcVcCrossConnectHighIfIndex = ifIndexTarget
atmSvcVcCrossConnectHighVpi    = VpiTarget
atmSvcVcCrossConnectHighVci    = VciTarget
  atmSvcVcCrossConnectCreationTime
  atmSvcVcCrossConnectRowStatus
```

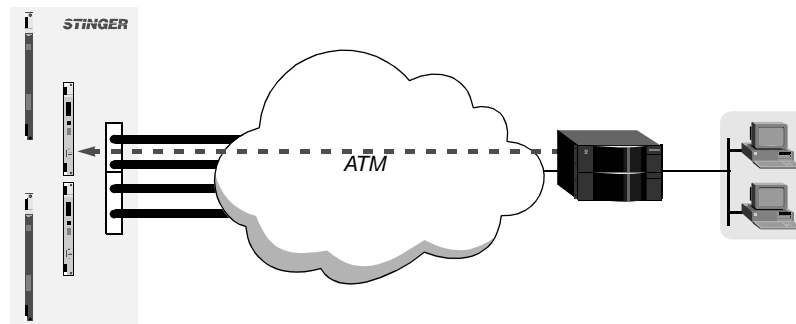
Configuring the SPVC notification interval

You can change the `atmSoftPvcNotificationInterval` by issuing a standard `set` request. The default interval is 30 seconds.

Configuring a terminating PVC

A terminating PVC is not switched through, but terminates in the Stinger controller, as shown in Figure 2-8.

Figure 2-8. Terminating PVC



This type of connection is used for SNMP management or Telnet access to the unit. The Stinger brings up the PVC on the basis of the VPI-VCI assignment and other ATM parameters. This section shows only the ATM configuration aspects of terminating PVCs.

MIB table overview for a terminating PVC configuration

To create a terminating connection, use the `internetProfileTable` defined in `mibinet.mib`. Configuring a PVC in this table creates a `Connection` profile in the Stinger unit's command-line interface. Figure 2-9 shows objects that might require nondefault settings for a terminating PVC.

Figure 2-9. Lucent Access Internet MIB objects for a terminating PVC

mibinternetProfileTable

```
internetProfile__station
internetProfile__active
internetProfile__encapsulation_protocol
internetProfile__ip_options__remote_address
internetProfile__telco_options__nailed_groups
internetProfile__atm_options__vpi
internetProfile__atm_options__vci
internetProfile__atm_options__nailed_group
internetProfile__atm_options__vp_switching
internetProfile__atm_qos_options__usr_up_stream_contract
internetProfile__atm_qos_options__usr_dn_stream_contract
internetProfile__action
```

Note: The system does not check values you specify in `mibinet.mib`, so verifying these numbers is important. For example, notice that two settings in the profile specify the nailed-group number for the physical interface of the PVC. The system does not reject the profile if the nailed-group numbers are inconsistent, or if the specified traffic contracts do not exist, but the inconsistent information causes the connection to fail.

Object	Specifies
<code>internetProfile__station</code>	Name of the far-end device. This will also be the name of the Connection profile in the command-line interface.
<code>internetProfile__active</code>	Activate/deactivate the terminating connection. Set to <code>yes</code> to activate the connection.
<code>internetProfile__encapsulation_protocol</code>	Encapsulation protocol for the connection. This object must be set to <code>atm</code> for a terminating PVC.
<code>internetProfile__ip_options__remote_address</code>	Destination IP address, which lies at the end of a PVC whose first hop is known by the specified VPI-VCI pair.
<code>internetProfile__telco_options__nailed_groups</code>	Nailed-group number of the physical interface used for the terminating PVC. Note: This object and the <code>internetProfile__atm_options__nailed-group</code> object must specify the same nailed group, or the connection cannot be established.
<code>internetProfile__atm_options__vpi</code>	VPI-VCI pair for the terminating PVC. Be sure to use VPI and VCI assignments that are within the valid range for the physical interface. A VPI-VCI assignment that is not compatible with the physical interface configuration causes the connection to fail with an error message.
<code>internetProfile__atm_options__vci</code>	
<code>internetProfile__atm_options__nailed_group</code>	Nailed-group number of the physical interface used for the terminating PVC. Note: This object and the <code>internetProfile__telco_options__nailed-groups</code> object must specify the same nailed group, or the connection cannot be established.

Object	Specifies
<code>internetProfile__atm_options__vp_switching</code>	Enable/disable VP switching. A value of <code>no</code> means the PVC is a virtual channel connection. A value of <code>yes</code> means it is a virtual path connection.
<code>internetProfile__atm_qos_options__usr_up_stream_contract</code>	Name of an ATM-QOS profile that defines a traffic descriptor to be applied to upstream traffic on the PVC. If no name is specified, the default ATM-QOS profile (traffic descriptor 1) is used.
<code>internetProfile__atm_qos_options__usr_dn_stream_contract</code>	Name of an ATM-QOS profile that defines a traffic descriptor to be applied to downstream traffic on the PVC. If no name is specified, the default ATM-QOS profile (traffic descriptor 1) is used.
<code>internetProfile__action</code>	Create or destroy a <code>Connection</code> profile that is visible in the command-line interface.

Configuring a terminating connection

To create terminating connection, follow these steps:

- 1 Create a `MibInternetProfileEntry` with the default settings for `Connection` profiles.
- 2 Modify the objects described in “MIB table overview for a terminating PVC configuration” on page 2-15.
- 3 Set the `internetProfile__active` object to `yes` to activate the connection.

Minimal required objects to create a profile

You create the `MibInternetProfileEntry` with default settings by naming the profile and specifying the `createProfile` action. For example:

```
internetProfile__station = ip-device-1
internetProfile__action = createProfile
```

Sample MibInternetProfileEntry for a terminating PVC

Following is a sample `MibInternetProfileEntry` for a terminating PVC. The remote device is named `ip-device-1` and has the IP address `1.1.1.1/28`. The PVC is a virtual channel connection using the VPI-VCI pair 3-35, associated with the first LIM port in slot 2 (nailed group 51).

```
mibinternetProfileTable
internetProfile__station = ip-device-1
internetProfile__active = yes
internetProfile__encapsulation_protocol = atm
internetProfile__ip_options__remote_address = 1.1.1.1/28
internetProfile__telco_options__nailed_groups = 51
internetProfile__atm_options__vpi = 3
internetProfile__atm_options__vci = 35
internetProfile__atm_options__nailed_group = 51
internetProfile__atm_options__vp_switching = no
internetProfile__atm_qos_options__usr_up_stream_contract = default
```

```
internetProfile__atm_qos_options__usr_dn_stream_contract = default
internetProfile__action = createProfile
```

This entry creates a Connection profile in the Stinger interface with the following nondefault settings:

```
[in CONNECTION/ip-device-1]
station* = ip-device-1
active = yes
encapsulation-protocol = atm

[in CONNECTION/ip-device-1:ip-options]
remote-address = 1.1.1.1/28

[in CONNECTION/ip-device-1:telco-options]
nailed-groups = 51

[in CONNECTION/ip-device-1:atm-options]
vpi = 3
vci = 35
nailed-group = 51
vp-switching = no

[in CONNECTION/ip-device-1:atm-qos-options]
usr-up-stream-contract = default
usr-dn-stream-contract = default
```

For details about these parameters, see the *Stinger Reference*. For information about configuring terminating PVCs in the Stinger interface, see the *Stinger ATM Configuration Guide*.

Configuring PNNI routing

With the current software version, the system supports one PNNI node in the Stinger unit, which is always a lowest-level node. All PNNI-enabled trunk ports are bound to the same node. The system assigns default values to the objects in the `pnniNodeTable`, including a default PNNI node ATM address. However, you must ensure that the PNNI node address, node ID, and peer group ID are valid for your network. Otherwise, the system cannot begin operating as a PNNI node.

Note: For details about ATM addresses, see *ATM Forum Addressing: User Guide version 1.0*, AF-RA-0105.000, January, 1999 and *ATM Forum Addressing: Reference Guide*, AF-RA-0106.000, February, 1999.

A *PNNI node ID* is a 22-byte, 44-digit hexadecimal number used to identify the system as a logical node within a PNNI peer group. The first byte of the node ID is the node level—for example, 96 (0x60). The second byte is set to 160 (0xA0) by convention, and the remaining 20 bytes are set to the PNNI node ATM end system address (AESA).

A *PNNI peer group ID* is a 14-byte, 28-digit hexadecimal number used to group nodes into a PNNI peer group. All members of the same PNNI peer group have the same peer group ID. The first byte of the peer group ID is the node level—for example, 96 (0x60). The remaining 13 bytes are set to the PNNI node AESA prefix.

Following are the steps for configuring an ATM interface for PNNI:

- 1 Set the `atmIntConfigType` object in the `atmInterfaceExtTable` to `atmfppnni1Dot0` to enable PNNI on the ATM interface.

- 2 For the PNNI node, set the `pnniNodeId`, `pnniNodeAtmAddress`, and `pnniNodePeerGroupId` in the `pnniNodeTable` to values that are consistent with the addressing scheme in use on your network.

Note: For details about the system-generated default values for the `pnniNodeId`, `pnniNodeAtmAddress`, and `pnniNodePeerGroupId` objects, see the *Stinger Private Network-to-Network Interface (PNNI) Supplement*.

- 3 Configure PNNI protocol parameters and timers if needed.

MIB table overview for PNNI node configuration

The `atmInterfaceConfTable` from an IETF draft entitled *Definitions of Supplemental Managed Objects for ATM Management* (draft-ietf-atommib-atm2-13.txt), September 1999, contains ATM interface configuration and monitoring information that is not defined in the `atmInterfaceConfTable` from the ATM-MIB. The `pnniNodeTable` is defined in *Private Network-Network Interface Specification Version 1.0 (PNNI 1.0)*, af-pnni-0055.000, March 1996. Figure 2-10 shows objects related to PNNI node configuration:

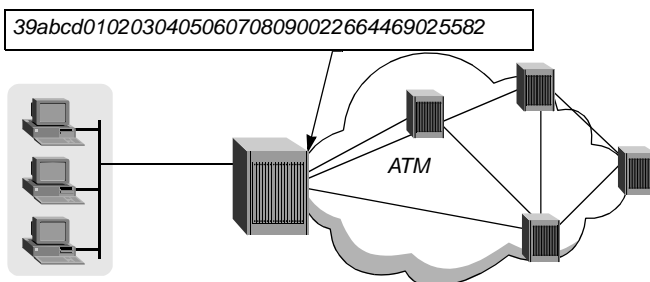
Figure 2-10. ATM Interface extensions and PNNI Node table

atmInterfaceExtTable	pnniNodeTable
atmIntfConfigType	<code>pnniNodeIndex</code>
<code>atmIntfActualType</code>	pnniNodeLevel
<code>atmIntfConfigSide</code>	pnniNodeId
<code>atmIntfActualSide</code>	pnniNodeLowest
<code>atmIntfIlmiAdminStatus</code>	pnniNodeAdminStatus
<code>atmIntfIlmiOperStatus</code>	<code>pnniNodeOperStatus</code>
<code>atmIntfIlmiFsmState</code>	<code>pnniNodeDomainName</code>
<code>atmIntfIlmiEstablishConPollIntvl</code>	pnniNodeAtmAddress
<code>atmIntfIlmiCheckConPollIntvl</code>	pnniNodePeerGroupId
<code>atmIntfIlmiConPollInactFactor</code>	<code>pnniNodeRestrictedTransit</code>
<code>atmIntfIlmiPublicPrivateIndctr</code>	<code>pnniNodeComplexRep</code>
<code>atmInterfaceConfMaxSvpcVpi</code>	<code>pnniNodeRestrictedBranching</code>
<code>atmInterfaceCurrentMaxSvpcVpi</code>	<code>pnniNodeDatabaseOverload</code>
<code>atmInterfaceConfMaxSvccVpi</code>	<code>pnniNodePtses</code>
<code>atmInterfaceCurrentMaxSvccVpi</code>	pnniNodeRowStatus
<code>atmInterfaceConfMinSvccVci</code>	
<code>atmInterfaceCurrentMinSvccVci</code>	
<code>atmIntfSigVccRxTrafficDescrIndex</code>	
<code>atmIntfSigVccTxTrafficDescrIndex</code>	
<code>atmIntfPvcFailures</code>	
<code>atmIntfCurrentlyFailingPVpls</code>	
<code>atmIntfCurrentlyFailingPVcls</code>	
<code>atmIntfPvcFailuresTrapEnable</code>	
<code>atmIntfPvcNotificationInterval</code>	

Example of a PNNI configuration

Figure 2-11 shows a PNNI node with a 20-byte ATM end system address (AESAs). A group of lowest-level nodes pass data to each other and to end systems across physical links.

Figure 2-11. PNNI lowest-level node with its ATM address



For an example of how to obtain the proper interface index for an ATM interface, see “Finding the ATM interface ifIndex” on page 2-3.

Following are sample settings for the lowest-level node:

```

ifIndex                = ifIndex
atmIntfConfigType     = atmfpnnildot0
  atmIntfSigVccRxTrafficDescrIndex = 2
  atmIntfSigVccTxTrafficDescrIndex = 2

PnniNodeIndex         = 1
  pnniNodeAtmAddress   =
39abcd0102030405060708090022664469025582
  pnniNodeId           =
60a039abcd0102030405060708090022664469025582
  pnniNodePeerGroupId  = 6039abcd01020304050607080900
  pnniNodeRestrictedTransit = false
  pnniNodeComplexRep   = false
  pnniNodeRowStatus    = createAndGo
  
```

Conditions under which set requests are accepted in PNNI tables

Table 2-3 through Table 2-12 identify the PNNI MIB tables and list the conditions under which set requests are accepted on each of the MIB objects.

Table 2-3. PnniNodeTable conditions for acceptance of set requests

Object	set request accepted	set request rejected
PnniNodeIndex	Not applicable (read-only)	Not applicable (read-only)
PnniNodeLevel	RowStatus=Active AdminStatus=Down	RowStatus = Active
PnniNodeId	RowStatus=Active AdminStatus=Down	RowStatus = Active
PnniNodeLowest	RowStatus=Active AdminStatus=Down	RowStatus = Active
PnniNodeOperStatus	Not applicable (read-only)	Not applicable (read-only)
PnniNodeDomainName	Under all conditions	Never rejected
PnniNodeAtmAddress	RowStatus=Active AdminStatus=Down	RowStatus = Active
PnniNodePeerGroupId	RowStatus=Active, AdminStatus=Down	RowStatus = Active

Table 2-3. PnniNodeTable conditions for acceptance of set requests (continued)

Object	set request accepted	set request rejected
PnniNodeRestrictedTransit	Under all conditions	Never rejected
PnniNodeComplexRep	Under all conditions	Never rejected
PnniNodeRestrictedBranching	Not applicable (read-only)	Not applicable (read-only)
PnniNodeDatabaseOverload	Not applicable (read-only)	Not applicable (read-only)
PnniNodePtses	Not applicable (read-only)	Not applicable (read-only)

Table 2-4. PnniNodePglEntry table conditions for acceptance of set requests

Object	set request accepted	set request rejected
PnniNodePglLeadershipPriority	Value = 0	Any value other than 0
PnniNodeCfgParentNodeIndex	Value = 0	Any value other than 0
PnniNodePglInitTime	Under all conditions	Never rejected
PnniNodePglOverrideDelay	Under all conditions	Never rejected
PnniNodePglReelectTime	Under all conditions	Never rejected
PnniNodePglState	Not applicable (read-only)	Not applicable (read-only)
PnniNodePreferredPgl	Not applicable (read-only)	Not applicable (read-only)
PnniNodePeerGroupLeader	Not applicable (read-only)	Not applicable (read-only)
PnniNodePglTimeStamp	Not applicable (read-only)	Not applicable (read-only)
PnniNodeActiveParentNodeId	Not applicable (read-only)	Not applicable (read-only)

Table 2-5. PnniNodeTimer table conditions for acceptance of set requests

Object	set request accepted	set request rejected
PnniNodePtseHolddown	Value ≥ 0	Value < 0
PnniNodeHelloHolddown	Value ≥ 0	Value < 0
PnniNodeHelloInterval	Value ≥ 0	Value < 0
PnniNodeHelloInactivityFactor	Value ≥ 0	Value < 0
PnniNodeHlinkInact	Value ≥ 0	Value < 0
PnniNodePtseRefreshInterval	Value ≥ 0	Value < 0
PnniNodePtseLifetimeFactor	Value ≥ 101 and value ≤ 1000	Value < 101 and value > 1000
PnniNodeRxmtInterval	Value ≥ 0	Value < 0
PnniNodePeerDelayedAckInterval	Value ≥ 0	Value < 0
PnniNodeAvcrPm	Value ≥ 1 and value ≤ 99	Value < 1 and value > 99
PnniNodeAvcrMt	Value ≥ 1 and value ≤ 99	Value < 1 and value > 99
PnniNodeCdvPm	Value ≥ 1 and value ≤ 99	Value < 1 and value > 99
PnniNodeCdtPm	Value ≥ 1 and value ≤ 99	Value < 1 and value > 99

Typical Stinger Configurations
Configuring PNNI routing

Table 2-6. PnniNodeSvccEntry table conditions for acceptance of set requests

Object	set request accepted	set request rejected
PnniNodeSvccInitTime	Value = DEFVAL (4)	Value ≠ DEFVAL
PnniNodeSvccRetryTime	Value ≥ 0	Value ≤ 0
PnniNodeSvccCallingIntegrityTime	Value ≥ 0	Value ≤ 0
PnniNodeSvccCalledIntegrityTime	Value ≥ 0	Value ≤ 0
PnniNodeSvccTrafficDescriptorIndex	Value = TrafficDescrIndex	Value ≤ 0

Table 2-7. PnniScopeMapping table conditions for acceptance of set requests

Object	set request accepted	set request rejected
PnniScopeLocalNetwork	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniScopeLocalNetworkPlusOne	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniScopeLocalNetworkPlusTwo	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniScopeSiteMinusOne	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniScopeIntraSite	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniScopeSitePlusOne	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniScopeOrganizationMinusOne	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniScopeIntraOrganization	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniScopeOrganizationPlusOne	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniScopeCommunityMinusOne	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniScopeIntraCommunity	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniScopeCommunityPlusOne	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniScopeRegional	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniScopeInterRegional	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniScopeGlobal	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104

Table 2-8. PnniSummaryAddressTable conditions for acceptance of set requests

Object	set request accepted	set request rejected
PnniSummaryAddressType	Not applicable (read-only)	Not applicable (read-only)
PnniSummaryAddress	Not applicable (read-only)	Not applicable (read-only)
PnniSummaryAddressPrefixLength	Not applicable (read-only)	Not applicable (read-only)
PnniSummaryAddressSuppress	RowStatus = Active	Never rejected
PnniSummaryAddressState	Not applicable (read-only)	Not applicable (read-only)

Table 2-9. PnniIfTable conditions for acceptance of set requests

Object	set request accepted	set request rejected
PnniIfNodeIndex	Value > 0 and value <= 65535	Value <= 0 and value > 65535
PnniIfPortId	Not applicable (read-only)	Not applicable (read-only)
PnniIfAggrToken	Value >= 0	Value < 0
PnniIfVPCapability	Under all conditions	Never rejected
PnniIfAdmWeightCbr	Value >= 1 and value <=16777215	Value < 1 and value > 16777215
PnniIfAdmWeightRtVbr	Value >= 1 and value <=16777215	Value < 1 and value > 16777215
PnniIfAdmWeightNrtVbr	Value >= 1 and value <=16777215	Value < 1 and value > 16777215
PnniIfAdmWeightAbr	Value >= 1 and value <=16777215	Value < 1 and value > 16777215
PnniIfAdmWeightUbr	Value >= 1 and value <=16777215	Value < 1 and value > 16777215
PnniIfRccServiceCategory	Under all conditions	Never rejected
PnniIfRccTrafficDescrIndex	Under all conditions	Never rejected

Table 2-10. PnniMetricsTable conditions for acceptance of set requests

Object	set request accepted	set request rejected
PnniMetricsTag	Not applicable (read-only)	Not applicable (read-only)
PnniMetricsDirection	Not applicable (read-only)	Not applicable (read-only)
PnniMetricsIndex	Not applicable (read-only)	Not applicable (read-only)
PnniMetricsClasses	Not applicable (read-only)	Not applicable (read-only)
PnniMetricsGcacClp	Value = 1 or 2	Value ≠ 1 or 2
PnniMetricsAdminWeight	Value >= 1 and value <= 167772215	Value < 1 and value > 167772215
PnniMetrics1 ... PnniMetrics8	Value >= 0	Value < 0

Table 2-11. PnniRouteAddrTable conditions for acceptance of set requests

Object	set request accepted	set request rejected
PnniRouteAddrAddress	Not applicable (read-only)	Not applicable (read-only)
PnniRouteAddrPrefixLength	Not applicable (read-only)	Not applicable (read-only)
PnniRouteAddrIndex	Not applicable (read-only)	Not applicable (read-only)
PnniRouteAddrIfIndex	RowStatus = Active	Not rejected
PnniRouteAddrAdvertisedNodeId	RowStatus = Active	Not rejected
PnniRouteAddrAdvertisedPortId	RowStatus = Active	Not rejected
PnniRouteAddrType	RowStatus = Active	Not rejected
PnniRouteAddrProto	Not applicable (read-only)	Not applicable (read-only)
PnniRouteAddrPnniScope	Value >= 0 and value <= 104	Value < 0 and value > 104
PnniRouteAddrVpCapability	Value = True or False	Value ≠ True or False
PnniRouteAddrMetricsTag	Value >= 0 and value <=2147483647	Value < 0 and value >2147483647
PnniRouteAddrPtseId	Not applicable (read-only)	Not applicable (read-only)

Typical Stinger Configurations

Configuring PNNI routing

Table 2-11. PnniRouteAddrTable conditions for acceptance of set requests (continued)

Object	set request accepted	set request rejected
PnniRouteAddrOriginateAdvertise	Value = True or False	Value ≠ True or False
PnniRouteAddrInfo	Not applicable (read-only)	Not applicable (read-only)
PnniRouteAddrOperStatus	Not applicable (read-only)	Not applicable (read-only)
PnniRouteAddrTimeStamp	Not applicable (read-only)	Not applicable (read-only)

Table 2-12. PnniRouteTnsTable conditions for acceptance of set requests

Object	set request accepted	set request rejected
PnniRouteTnsType	Not applicable (read-only)	Not applicable (read-only)
PnniRouteTnsPlan	Not applicable (read-only)	Not applicable (read-only)
PnniRouteTnsId	During the creation of a new entry when the value of RouteTnsId is 4 characters long	During the creation of a new entry when the value of RouteTnsId is not 4 characters long
PnniRouteTnsIndex	Not applicable (read-only)	Not applicable (read-only)
PnniRouteTnsIfIndex	Value ≥ 0	Value < 0
PnniRouteTnsAdvertisingNodeId	Value corresponding to NodeIndex	Never rejected
PnniRouteTnsRouteType	RowStatus = Active	Never rejected
PnniRouteTnsProto	Not applicable (read-only)	Not applicable (read-only)
PnniRouteTnsPnniScope	Value ≥ 0 and value ≤ 104	Value < 0 and value > 104
PnniRouteTnsVPCapability	RowStatus = Active	Never rejected
PnniRouteTnsMetricsTag	Value ≥ 0 and value ≤ 2147483647	Value < 0 and value > 2147483647
PnniRouteTnsPtseId	Not applicable (read-only)	Not applicable (read-only)
PnniRouteTnsOriginateAdvertisement	Value = True or False	Value ≠ True or False
PnniRouteTnsInfo	Not applicable (read-only)	Not applicable (read-only)

Management Information Bases (MIBs)

3

Modifications to standard settings	3-1
Proprietary MIBs for Stinger physical interfaces	3-3
RFC 2515 ATM management objects	3-13
SNMP M4 network element view MIB	3-18
ATM Forum Soft PVC MIB	3-18
Supplemental managed objects for ATM management	3-21
ATM Forum PNNI MIB	3-28
ATM interface status MIB	3-41
SONET extension MIB	3-48

The tables in this chapter show the Value type or range of valid values and the default setting for the MIBs supported in the Stinger. In most cases, the values supported in the Stinger conform with the Value type or ranges and defaults stated in the MIBs. When the Stinger default value differs from a MIB default value, the object entry includes two rows, with the MIB default stated in the first row and the Stinger default below it.

Modifications to standard settings

Table 3-1 shows the SNMP objects that deviate from standard MIB defaults on Stinger units.

Table 3-1. Items in which Stinger units differ from standard MIB defaults

MIB table	Object name	Stinger-specific default
ATM Soft PVC base group	atmSoftPvcCallFailuresTrapEnable	true
PNNI node	pnniNodePeerGroupId	0
	pnniNodeComplexRep	false
	pnniNodeRestrictedBranching	false
	pnniNodeIndex	1
PNNI interface	pnniIfRccServiceCategory	nrt-Vbr
	pnniIfRccTrafficDescrIndex	2

Management Information Bases (MIBs)

Modifications to standard settings

Table 3-1. Items in which Stinger units differ from standard MIB defaults (continued)

MIB table	Object name	Stinger-specific default
PNNI reachable addresses	pnniRouteAddrIfIndex	0
	pnniRouteAddrPnniScope	Level of lowest-level node
	pnniRouteAddrInfo	Read-only
PNNI reachable transit networks	pnniRouteTnsInfo	Read-only
PNNI PGL election <i>Not applicable in current version.</i>	pnniNodePglState	starting
	pnniNodePglTimeStamp	0
PNNI nodal SVCC-based RCC <i>Not applicable in current version.</i>	pnniNodeSvccTrafficDescriptorIndex	0
PNNI scope mapping table <i>Not applicable in current version.</i>	Not applicable.	
PNNI route node	Not supported: All read-create objects are set to read-only.	
PNNI DTL stack	Not supported: All read-create objects are set to read-only.	
ATM VPL SVC cross-connect	atmSvcVpCrossConnectRowStatus	Read-only
ATM VCL SVC cross-connect	atmSvcVcCrossConnectRowStatus	Read-only

Table 3-1. Items in which Stinger units differ from standard MIB defaults (continued)

MIB table	Object name	Stinger-specific default
ATM interface configuration	atmInterfaceMaxVpcs	oc3-255, ds3-255, xdsl-15
	atmInterfaceMaxVccs	oc3-8192, ds3-8192, xdsl-95
	atmInterfaceMaxActiveVpiBits	oc3-8, ds3-8, xdsl-4 (read-only)
	atmInterfaceMaxActiveVciBits	oc3-13, ds3-13, xdsl-7 (read-only)
	atmInterfaceIlmiVpi	Not currently supported (ILMI)
	atmInterfaceIlmiVci	Not currently supported (ILMI)
	atmInterfaceAddressType	Not currently supported (ILMI)
	atmInterfaceAdminAddress	Not currently supported (ILMI)
	atmInterfaceMyNeighborIpAddress	Not currently supported (ILMI)
	atmInterfaceMyNeighborIfName	Not currently supported (ILMI)
	atmInterfaceSubscrAddress	Not currently supported (ILMI)
ATM interface configuration extension	atmIntfConfigType	PVC. Must set to atmFpnni1Dot0 to support PNNI on an interface.
	atmIntfConfigSide	other (1)
	atmIntfIlmiAdminStatus	down
	atmIntfIlmiEstablishConPollIntvl	Not currently supported (ILMI)
	atmInterfaceConfMinSvccVci	32
	atmIntfSigVccRxTrafficDescrIndex	2
	atmIntfSigVccTxTrafficDescrIndex	2

Proprietary MIBs for Stinger physical interfaces

Stinger units support DS1, DS3, and OC3 trunk modules for connecting to the ATM network. Each physical interface has an associated ATM interface. When the system initially starts up, the physical interfaces are configured automatically to their default settings.

DS1-ATM MIB

You can modify DS1-ATM interface settings by writing values in the DS1 MIB, which is accessible under the following MIB tree structure:

```
private.ascend.configuration
```

The MIB tables shown in Table 3-2 through Table 3-5 identify a profile for a specific trunk port. All object names in Table 3-2 are preceded by the following string:

```
ds1AtmNetworkProfile__
```

The profile is indexed by the port's physical address in the Stinger unit.

Table 3-2. mibds1AtmNetworkProfileTable

MIB object	Value type or range	Default
name	Display string	<i>physical-address</i>
physical_address__shelf physical_address__slot physical_address__item_number	Integers	N/A
enabled	no(1), yes(2)	no
spare_physical_address__shelf spare_physical_address__slot spare_physical_address__item_number	Integers	0 0 0
sparing_mode	inactive(1), manual(2), automatic(3)	inactive
ignore_lineup	system_defined (1), no (2), yes (3)	system_defined

All objects in Table 3-3 are preceded by the following string:

```
ds1AtmNetworkProfile__line_config__
```

Table 3-3. mibds1AtmNetworkProfileTable

MIB object	Value type or range	Default
frame_type	d4 (1) esf (2) g703 (3) n-2ds (4)	esf
encoding	ami (1) b8zs (2) hdb3 (3) none (4)	b8zs
nailed_group	Integer	Unique value
loopback	noLoopback (1) feLoopback (2) lineLoopback (3) remoteLoopback (4) localLoopback (5)	noLoopback
clock_source	eligible (1),notEligible (2)	eligible

Table 3-3. *mibds1AtmNetworkProfileTable* (continued)

MIB object	Value type or range	Default
clock_priority	highPriority (2), middlePriority (3), lowPriority (4)	middlePriority
oFDL	none (1) atNT (2) ansi (3) sprint (4)	N/A
send_code	noCode (1) lineCode (2) payloadCode (3) resetCode (4) qrsCode (5) n-511Pattern (6) n-3In24Pattern (7) n-1In16Pattern (8) allOnesPattern (9) allZerosPattern (10) altOnesZerosPattern (11) dblAltOnesZerosPattern (12) n-2Pow20Pattern (13)	N/A
front_end_type	shortHaul (1), longHaul (2)	shortHaul
line_length	n-1133 (1) n-134266 (2) n-267399 (3) n-400533 (4) n-534655 (5)	n-1133
line_build_out	n-0Db (1) n-75Db (2) n-15Db (3) n-2255Db (4)	n-0Db
pcm_mode	isdn (24), clearChannel (25)	N/A
coset_enabled	no (1), yes (2)	yes
scrambling_enabled	no (1), yes (2)	N/A
hec_correction_enabled	no (1), yes (2)	no
vp_switching_vpi	Integer	15

Management Information Bases (MIBs)
Proprietary MIBs for Stinger physical interfaces

Table 3-3. *mibds1AtmNetworkProfileTable* (continued)

MIB object	Value type or range	Default
vpi_vci_range	n-01-3232767 (1) n-03-3216383 (2) n-07-328191 (3) n-015-324095 (4) n-031-322047 (5) n-063-321023 (6) n-0127-32511 (7) n-0255-32255 (8) vpi0255Vci32255 (9) vpi0255Vci32511 (10) vpi0255Vci321023 (11) vpi0255Vci322047 (12) vpi0255Vci324095 (13) vpi0255Vci328191 (14) vpi0255Vci3216383 (15)	N/A
status_change_trap_enable	no (1), yes (2)	no

All objects in Table 3-4 are preceded by the following string:

ds1AtmNetworkProfile__line_config__ima_option_config__txlink_config__

Table 3-4. *mibds1AtmNetworkProfileTable*

MIB object	Value type or range	Default
ne_tx_lid	Integer	N/A
add_link_cond_time	Integer	N/A
link_recovery_type	manual (2) slow (3) fast (4)	fast
fault_clearing_type	manual (2), auto (3)	auto
fault_clearing_time	Integer	N/A
priority	Integer	N/A

All objects in Table 3-5 are preceded by the following string:

ds1AtmNetworkProfile__line_config__ima_option_config__rxlink_config__

Table 3-5. *mibds1AtmNetworkProfileTable*

MIB object	Value type or range	Default
add_link_cond_time	Integer	N/A
link_recovery_type	manual (2) slow (3) fast (4)	
rec_link_cond_time	Integer	N/A
rx_lid_learning_time	Integer	N/A
fault_clearing_type	manual (2), auto (3)	
fault_clearing_time	Integer	N/A

Table 3-5. *mibds1AtmNetworkProfileTable* (continued)

MIB object	Value type or range	Default
in_defect_int_time	Integer	N/a
out_defect_int_time	Integer	N/A
defect_ratio	Integer	N/A

DS3-ATM MIB

You can modify DS3-ATM interface settings by writing values in the DS3 MIB, which is accessible under the following MIB tree structure:

```
private.ascend.configuration
```

The MIB tables shown in Table 3-6 through Table 3-8 identify a profile for a specific trunk port. All object names in Table 3-6 are preceded by the following string:

```
ds3AtmNetworkProfile__
```

The profile is indexed by the port's physical address in the Stinger unit.

Table 3-6. *mibds3AtmNetworkProfileTable*

MIB object	Value type or range	Default
name	Display String	<i>physical-address</i>
physical_address__shelf physical_address__slot physical_address__item_number	Integers	N/A
enabled	no(1), yes(2)	no
spare_physical_address__shelf spare_physical_address__slot spare_physical_address__item_number	Integers	0 0 0
sparing_mode	inactive(1) manual(2) automatic(3)	inactive
profile_number	Integer	0
ignore_lineup	system_defined (1) no (2) yes (3)	system_defined
action	noAction (1) createProfile (2) deleteProfile (3)	noAction

All object names in Table 3-7 are preceded by the following string:

```
ds3AtmNetworkProfile__line_config__
```

Management Information Bases (MIBs)
Proprietary MIBs for Stinger physical interfaces

Table 3-7. *mibds3AtmNetworkProfileTable*

MIB object	Value type or range	Default
trunk_group	0 through 65535 <i>Not currently used.</i>	0
nailed_group	Integer	Unique value
route_port__slot_number__slot_number route_port__slot_number__shelf_number route_port__relative_port_number__ relative_port_number	Integers	N/A
activation	Not supported on Stinger units	
call_route_info__shelf call_route_info__slot call_route_info__item_number	Integers	0 0 0
loopback	no_loopback(1) facility_loopback(2) local_loopback(3)	no_loopback
high_tx_output	no(1), yes(2)	no
receive_equalization	no (1), yes (2)	
framer_mode	c_bit_ADM(1), c_BIT_PLCP(2), c_bit_adm_loop_timed(3), c_bit_plcp_loop_timed(4), c_bit_adm_frame_locked(5), c_bit_plcp_frame_locked(6)	c_BIT_PLCP
vpi_vci_range	n-0_1-32_32767(1), n-0_3-32_16383(2), n-0_7-32_8191(3), n-0_15-32_4095(4), n-0_31-32_2047(5), n-0_63-32_1023(6), n-0_127-32_511(7), n-0_255-32_255(8), vpi_0_255_vci_32_255(9), vpi_0_255_vci_32_511(10), vpi_0_255_vci_32_1023(11), vpi_0_255_vci_32_2047(12), vpi_0_255_vci_32_4095(13), vpi_0_255_vci_32_8191(14), vpi_0_255_vci_32_16383(15)	vpi-0-255- vci-32-8191
clock_source	eligible(1), not_eligible(2)	not_eligible
clock_priority	high_priority(2), middle_priority(3), low_priority(4)	middle_priority
cell_payload_scramble	no(1), yes(2)	no
status_change_trap_enable	no (1) yes (2)	

The MIB table shown in Table 3-8 specifies the VPI numbers that are used for virtual circuit switching, in addition to the default VPI zero, which is always used for this purpose. All object names in the table are preceded by the following string:

`ds3AtmNetworkProfile__line_config__vc_switching_vpi`

Table 3-8. mibds3AtmNetworkProfileTable

MIB object	Value type or range	Default
shelf__ slot__ item__ index__	Integers	N/A
vc_switching_vpi	Integer	0

OC3-ATM MIB

You can modify OC3-ATM interface settings by writing values in the OC3 MIB, which is accessible under the following MIB tree structure:

`private.ascend.configuration`

The MIB table shown in Table 3-9 identifies a profile for a specific trunk port. All object names in the table are preceded by the following string:

`oc3AtmNetworkProfile__`

The profile is indexed by the port's physical address in the Stinger unit.

Table 3-9. miboc3AtmNetworkProfileTable

MIB object	Value type or range	Default
name	DisplayString	<i>physical-address</i>
physical_address__shelf physical_address__slot physical_address__item_number	Integers	N/A
enabled	no(1), yes(2)	no
spare_physical_address__shelf spare_physical_address__slot spare_physical_address__item_number	Integers	N/A
sparing_mode	inactive(1), manual(2), auto-matic(3)	inactive
profile_number	Integer	0
ignore_lineup	system_defined (1) no (2) yes (3)	system-defined
aps_config_name	DisplayString	N/A
action	noAction (1), createProfile (2), deleteProfile (3)	noAction

All object names in the network profile table shown in Table 3-10 are preceded by the following string:

`oc3AtmNetworkProfile__line_config__`

Management Information Bases (MIBs)
Proprietary MIBs for Stinger physical interfaces

Table 3-10. *miboc3AtmNetworkProfileTable*

MIB object	Value type or range	Default
trunk_group	0 through 65535	0
nailed_group	Integer	Unique value
activation	Not supported on Stinger units	
call_route_info_shelf call_route_info_slot call_route_info_item_number	Integers	0 0 0
loopback	no_loopback(1), facility_loopback(2), local_loopback(3)	no_loopback
framer_mode	sonet(1), sdh(2)	sonet
framer_rate	sTS_3c(1), sTS_1(2)	sTS_3c
rx_descramble_disabled	Not supported on Stinger units	
tx_scramble_disabled	no(1), yes(2)	no
rx_cell_payload_descramble_disabled	Not supported on Stinger units	
tx_cell_payload_scramble_disabled	no(1), yes(2)	no
loop_timing	no(1), yes(2)	yes
vpi_vci_range	n-0_1-32_32767(1), n-0_3-32_16383(2), n-0_7-32_8191(3), n-0_15-32_4095(4), n-0_31-32_2047(5), n-0_63-32_1023(6), n-0_127-32_511(7), n-0_255-32_255(8), vpi_0_255_vci_32_255(9), vpi_0_255_vci_32_511(10), vpi_0_255_vci_32_1023(11), vpi_0_255_vci_32_2047(12), vpi_0_255_vci_32_4095(13), vpi_0_255_vci_32_8191(14), vpi_0_255_vci_32_16383(15)	vpi-0-255-vci-3 2-8191
clock_source	eligible(1), not_eligible(2)	not_eligible
clock_priority	high_priority(2), middle_priority(3), low_priority(4)	middle_priority

Table 3-11 contains parameters for the `traffic_shapers` subprofile. All object names in the table are preceded by the following string:

`oc3AtmNetworkProfile__line_config__traffic_shapers`

Table 3-11. *miboc3AtmNetworkProfileTable*

MIB object	Value type or range	Default
shelf__ slot__ item__index__	Integer	N/A

Table 3-11. *miboc3AtmNetworkProfileTable (continued)*

MIB object	Value type or range	Default
enabled	no (1), yes (2)	N/A
bit_rate	Integer	N/A
peak_rate	Integer	N/A
max_burst_size	Integer	N/A
aggregate	no (1), yes (2)	N/A
priority	Integer	N/A

Table 3-12 specifies the VPI numbers to be used for virtual circuit switching, in addition to the default VPI zero, which is always used for this purpose. All object names in the table are preceded by the following string:

oc3AtmNetworkProfile__line_config__vc_switching__vpi

Table 3-12. *miboc3AtmNetworkProfileTable*

MIB object	Value type or range	Default
shelf__ slot__ item__ index__	Integers	N/A
vc_switching_vpi	Integer	0

Switch-Config MIB

The MIB tables shown in Table 3-13 contains configurable parameters for an ATM cell processor (the switch fabric). This profile can be used to configure the queues and shape the outgoing traffic. For example, a trunk port might use zero or more shapers to shape outgoing ATM traffic with certain VPIs.

This profile is indexed by the switch name. All object names in the table are preceded by the following string:

switchConfig__

Table 3-13. *mibswitchConfigTable*

MIB object	Value type or range	Default
switch_name	Display string	controller
action-o__	noAction (1), createProfile (2), deleteProfile (3)	noAction

Table 3-14 contains parameters for the outgoing-shaper profile. The object names in the table are preceded by the following string:

switchConfig__atm_parameters__outgoing_shaper

Management Information Bases (MIBs)
Proprietary MIBs for Stinger physical interfaces

Table 3-14. *mibswitchConfig_atm_parameters_outgoing_shaper* table

MIB object	Value type or range	Default
switch_name	DisplayString	N/A
index__	Integer	N/A
queue_index	Integer	N/A
vpi	Integer	N/A
bandwidth	Integer	N/A

Table 3-15 contains parameters for the *outgoing-queue* profile. The object names in the table are preceded by the following string:

`switchConfig_atm_parameters_outgoing_queue`

Table 3-15. *mibswitchConfig_atm_parameters_outgoing_queue* table

MIB object	Value type or range	Default
switch_name	DisplayString	N/A
index__	Integer	N/A
active	no (1), yes (2)	
name	DisplayString	N/A
physical_address__shelf physical_address__slot physical_address__item_number	Integers	N/A
cbr	no (1), yes (2)	N/A
real_time_vbr	no (1), yes (2)	N/A
non_real_time_vbr	no (1), yes (2)	N/A
ubr	no (1), yes (2)	N/A
high_priority_weight	Integer	N/A
low_priority_weight	Integer	N/A
source_port__shelf source_port__slot source_port__item_number	Integers	N/A
hop_level	n-0_level (1) n-1_level (2) n-2_level (3) n-3_level (4) any_level (5)	N/A

RFC 2515 ATM management objects

ATM interface configuration

Table 3-16 contains ATM-specific configuration information beyond that supported in the `ifTable`. It contains one entry for each ATM interface in the system.

Table 3-16. `atmInterfaceConfTable`

MIB object	Value type or range	Default
<code>atmInterfaceMaxVpcs</code>	0 through 4096	N/A (MIB default)
		oc3-255, ds3-255, xdsl-15 (Stinger default)
<code>atmInterfaceMaxVccs</code>	0 through 65536	N/A (MIB default)
		oc3-8192, ds3-8192, xdsl-95
<code>atmInterfaceConfVpcs</code>	0 through 4096	N/A
<code>atmInterfaceConfVccs</code>	0 through 65536	N/A
<code>atmInterfaceMaxActiveVpiBits</code>	0 through 12 (read-write)	N/A (MIB default)
	Read-only (Stinger)	oc3-8, ds3-8, xdsl-4 (Stinger default)
<code>atmInterfaceMaxActiveVciBits</code>	0 through 16 (read-write)	N/A (MIB default)
	oc3-13, ds3-13, xdsl-7 read only	read-only
<code>atmInterfaceIlmiVpi*</code>	<code>AtmVpIdentifier</code>	0
<code>atmInterfaceIlmiVci*</code>	<code>AtmVcIdentifier</code>	16
<code>atmInterfaceAddressType*</code>	private(1), nsapE164(2), nativeE164(3), other(4)	N/A
<code>atmInterfaceAdminAddress*</code>	Deprecated	Deprecated
<code>atmInterfaceMyNeighborIpAddress*</code>	<code>IpAddress</code>	N/A
<code>atmInterfaceMyNeighborIfName*</code>	<code>DisplayString</code>	N/A
<code>atmInterfaceCurrentMaxVpiBits</code>	0 through 12	N/A
<code>atmInterfaceCurrentMaxVciBits</code>	0 through 16	N/A
<code>atmInterfaceSubscrAddress*</code>	<code>AtmAddr</code>	N/A

* Not currently supported. Will be supported with ILMI.

Table 3-17 contains the DS3 Physical Layer Convergence Protocol (PLCP) configuration and state parameters of those ATM interfaces that use DS3 PLCP for carrying ATM cells over DS3.

Table 3-17. atmInterfaceDs3PlcpTable

MIB object	Value type or range	Default
atmInterfaceDs3PlcpSEFSs	Counter32	N/A
atmInterfaceDs3PlcpAlarmState	noAlarm(1), receivedFarEndAlarm(2), incomingLOF(3)	N/A
atmInterfaceDs3PlcpUASs	Counter32	N/A

Table 3-18 contains transmission convergency (TC) sublayer configuration and state parameters of ATM interfaces that use the TC sublayer for carrying ATM cells over SONET/SDH or DS3.

Table 3-18. atmInterfaceTCtable

MIB object	Value type or range	Default
atmInterfaceOCDEvents	Counter32	N/A
atmInterfaceTCAlarmState	noAlarm(1), lcdFailure(2)	N/A

ATM traffic descriptor configuration

Table 3-19 contains a set of self-consistent ATM traffic parameters. The ATM virtual link tables use this table to assign traffic parameters and a service category to the links' receive and transmit directions. To make sure that no fields in the `AtmTrafficDescrParamTable` can be modified when the row is active, remove the row from service by using the following command:

SET RowStatus = notInService

You can then set objects to define the entry. When the objects have been modified, you can activate the row by using the following command:

SET RowStatus = active

Table 3-19. atmTrafficDescrParamTable

MIB object	Value type or range	Default
atmTrafficDescrParamIndex	AtmTrafficDescrParamIndex	
atmTrafficDescrType	OBJECT IDENTIFIER	atmNoClpNoScr
atmTrafficDescrParam1	Integer32	0
atmTrafficDescrParam2	Integer32	0
atmTrafficDescrParam3	Integer32	0
atmTrafficDescrParam4	Integer32	0
atmTrafficDescrParam5	Integer32	0

Table 3-19. atmTrafficDescrParamTable (continued)

MIB object	Value type or range	Default
atmTrafficQoSClass	Deprecated	Deprecated
atmTrafficDescrRowStatus	RowStatus	active
atmServiceCategory	other(1), cbr(2), rtVbr(3), nrtVbr(4), abr(5), ubr(6)	ubr
atmTrafficFrameDiscard	Truth value	true

ATM interface VPL and VCL configuration

The virtual path link (VPL) and virtual channel link (VCL) configuration MIB tables shown in Table 3-20 and Table 3-21 contain configuration and state information about VPLs and VCLs, respectively. The tables can be used to create, delete, or modify a link that is terminated in the Stinger or cross-connected to another VPL or VCL.

Table 3-20. atmVplTable

MIB object	Value type or range	Default
atmVplVpi	AtmVpIdentifier	N/A
atmVplAdminStatus	AtmVorXAdminStatus	down
atmVplOperStatus	AtmVorXOperStatus	N/A
atmVplLastChange	AtmVorXLastChange	N/A
atmVplReceiveTrafficDescrIndex	0 through 2147483647	0*
atmVplTransmitTrafficDescrIndex	0 through 2147483647	0*
atmVplCrossConnectIdentifier	0 through 2147483647	N/A
atmVplRowStatus	RowStatus	active
atmVplCastType	AtmConnCastType	p2p*
atmVplConnKind	AtmConnKind	pvc*

* You must set atmVplAdminStatus to down to write these fields.

Table 3-21. atmVclTable

MIB object	Value type or range	Default
atmVclVpi	AtmVpIdentifier	N/A
atmVclVci	AtmVcIdentifier	N/A
atmVclAdminStatus	AtmVorXAdminStatus	down
atmVclOperStatus	AtmVorXOperStatus	N/A

Management Information Bases (MIBs)
RFC 2515 ATM management objects

Table 3-21. *atmVclTable (continued)*

MIB object	Value type or range	Default
atmVclLastChange	AtmVorXLastChange	N/A
atmVclReceiveTrafficDescrIndex	0 through 2147483647	0*
atmVclTransmitTrafficDescrIndex	0 through 2147483647	0*
atmVccAalType	aal1(1), aal34(2), aal5(3), other(4), unknown(5), aal2(6)	aal5*
atmVccAal5CpcsTransmitSduSize	1 through 65535	9188*
atmVccAal5CpcsReceiveSduSize	1 through 65535	9188*
atmVccAal5EncapsType	vcMultiplexRoutedProtocol(1), vcMultiplexBridgedProtocol8023(2), vcMultiplexBridgedProtocol8025(3), vcMultiplexBridgedProtocol8026(4), vcMultiplexLANemulation8023(5), vcMultiplexLANemulation8025(6), llcEncapsulation(7), multiprotocolFrameRelaySscs(8), other(9), unknown(10)	llcEncapsulation*
atmVclCrossConnectIdentifier	0 through 2147483647	N/A
atmVclRowStatus	RowStatus	active
atmVclCastType	AtmConnCastType	p2p*
atmVclConnKind	AtmConnKind	pvc*

* You must set atmVclAdminStatus to down to write these fields.

ATM cross-connects for PVCs

The ATM virtual path (VP) and virtual channel (VC) cross-connect MIB tables shown in Table 3-22 and Table 3-23 contain configuration and state information about VP and VC cross-connects for PVCs. The tables can be used to cross-connect VPLs or VCLs together. The atmVpCrossConnectIndex or atmVcCrossConnectIndex value is used to associate the cross-connected links. The terms Low and High in these variables refer to an ATM interface with a lower or higher ifIndex value.

Table 3-22. *atmVpCrossConnectTable*

MIB object	Value type or range	Default
atmVpCrossConnectIndex	1 through 2147483647	N/A
atmVpCrossConnectLowIfIndex	InterfaceIndex	N/A
atmVpCrossConnectLowVpi	AtmVpIdentifier	N/A
atmVpCrossConnectHighIfIndex	InterfaceIndex	N/A

Table 3-22. *atmVpCrossConnectTable* (continued)

MIB object	Value type or range	Default
atmVpCrossConnectHighVpi	AtmVpIdentifier	N/A
atmVpCrossConnectAdminStatus	AtmVorXAdminStatus	down
atmVpCrossConnectL2HOperStatus	AtmVorXOperStatus	N/A
atmVpCrossConnectH2LOperStatus	AtmVorXOperStatus	N/A
atmVpCrossConnectL2HLastChange	AtmVorXLastChange	N/A
atmVpCrossConnectH2LLastChange	AtmVorXLastChange	N/A
atmVpCrossConnectRowStatus	RowStatus	active

Table 3-23. *atmVcCrossConnectTable*

MIB object	Value type or range	Default
atmVcCrossConnectIndex	1 through 2147483647	N/A
atmVcCrossConnectLowIfIndex	InterfaceIndex	N/A
atmVcCrossConnectLowVpi	AtmVpIdentifier	N/A
atmVcCrossConnectLowVci	AtmVcIdentifier	N/A
atmVcCrossConnectHighIfIndex	InterfaceIndex	N/A
atmVcCrossConnectHighVpi	AtmVpIdentifier	N/A
atmVcCrossConnectHighVci	AtmVcIdentifier	N/A
atmVcCrossConnectAdminStatus	AtmVorXAdminStatus	down
atmVcCrossConnectL2HOperStatus	AtmVorXOperStatus	N/A
atmVcCrossConnectH2LOperStatus	AtmVorXOperStatus	N/A
atmVcCrossConnectL2HLastChange	AtmVorXLastChange	N/A
atmVcCrossConnectH2LLastChange	AtmVorXLastChange	N/A
atmVcCrossConnectRowStatus	RowStatus	active

AAL5 VCC performance statistics

Table 3-24 contains the AAL5 performance statistics of a VCC at the interface associated with an AAL5 entity.

Table 3-24. *aal5VccTable*

MIB object	Value type or range	Default
aal5VccVpi	AtmVpIdentifier	N/A
aal5VccVci	AtmVcIdentifier	N/A
aal5VccCrcErrors	Counter32	N/A
aal5VccSarTimeOuts	Counter32	N/A
aal5VccOverSizedSDUs	Counter32	N/A

SNMP M4 network element view MIB

Stinger units support the `atmfM4VpNextVpiTable` and `atmfM4VcNextVciTable` (Table 3-25 and Table 3-26) defined in the *ATM Forum SNMP M4 Network Element View MIB*, `af-nm-0095_001_mib.txt`, July, 1998. These read-only tables facilitate dynamic configuration of the `atmVplTable` and `atmVclTable` by allowing the NMS to obtain information about unused VPI and VCI values for an ATM interface.

Table 3-25. `atmfM4VpNextVpiTable`

MIB object	Value type or range	Default
<code>AtmfM4VpNextVpiValue</code>	-1 through 4095	N/A

Table 3-26. `atmfM4VcNextVciTable`

MIB object	Value type or range	Default
<code>AtmfM4VcNextVciValue</code>	-1 through 65535	N/A

ATM Forum Soft PVC MIB

This MIB is defined in the *ATM Forum Private Network-to-Network Interface Specification Version 1.0 Addendum (Soft PVC MIB)*, `af-pnni-0066.00`, September 1996. The objects shown in table Table 3-27 define counters and intervals related to success or failure of SPVC establishment.

Table 3-27. `atmSoftPvcBaseGroup`

MIB object	Value type or range	Default
<code>atmSoftPvcCallFailuresTrapEnable</code>	TruthValue	false (MIB default)
		true (Stinger default)
<code>atmSoftPvcCallFailures</code>	Counter32	N/A
<code>atmSoftPvcCurrentlyFailingSoft-PVccs</code>	Gauge32	N/A
<code>atmSoftPvcCurrentlyFailingSoft-PVpcs</code>	Gauge32	N/A
<code>atmSoftPvcNotificationInterval</code>	0 through 3600 seconds	30

Managing SPVCs at the originating switch

The MIB tables shown in Table 3-28 and Table 3-29 contain entries for soft permanent VCC (SPVCC) and soft permanent VPC (SPVPC) connections originating at a switch interface. To make sure that no fields in the `atmSoftPVccTable` or `atmSoftPVpcTable` can be modified when the row is active, remove the row from service by using the following command:

```
SET RowStatus = notInService
```

You can then set objects to define the entry. When the objects have been modified, you can activate the row by using the following command:

SET RowStatus = active

Table 3-28. *atmSoftPVccTable*

MIB object	Value type or range	Default
atmSoftPVccLeafReference	1 through 65535	N/A
atmSoftPVccTargetAddress	AtmAddr	N/A
atmSoftPVccTargetSelectType	required(1), any(2)	required
atmSoftPVccTargetVpi	0 through 4095	0
atmSoftPVccTargetVci	0 through 65535	N/A
atmSoftPVccLastReleaseCause	1 through 127	N/A
atmSoftPVccLastReleaseDiagnostic	OCTET STRING(SIZE(0 through 8))	N/A
atmSoftPVccOperStatus	other(1), establishmentInProgress(2), connected(3), retriesExhausted(4)	N/A
atmSoftPVccRestart	restart(1), noop(2)	N/A
atmSoftPVccRetryInterval	0 through 3600 seconds	10
atmSoftPVccRetryTimer	0 through 86400	N/A
atmSoftPVccRetryThreshold	0 through 65535	1
atmSoftPVccRetryFailures	Gauge32	N/A
atmSoftPVccRetryLimit	0 through 65535	0
atmSoftPVccRowStatus	RowStatus	N/A

Table 3-29. *atmSoftPVpcTable*

MIB object	Value type or range	Default
atmSoftPVpcLeafReference	1 through 63535	N/A
atmSoftPVpcTargetAddress	AtmAddr	N/A
atmSoftPVpcTargetSelectType	required(1), any(2)	required
atmSoftPVpcTargetVpi	0 through 4095	N/A
atmSoftPVpcLastReleaseCause	1 through 127	N/A
atmSoftPVpcLastReleaseDiagnostic	OCTET STRING(SIZE(0 through 8))	N/A
atmSoftPVpcOperStatus	other(1), establishmentIn- Progress(2), connected(3), retriesExhausted(4)	N/A
atmSoftPVpcRestart	restart(1), noop(2)	N/A
atmSoftPVpcRetryInterval	0 through 3600 seconds	10
atmSoftPVpcRetryTimer	0 through 86400	N/A
atmSoftPVpcRetryThreshold	0 through 65535	1

Management Information Bases (MIBs)

ATM Forum Soft PVC MIB

Table 3-29. *atmSoftPVpcTable* (continued)

MIB object	Value type or range	Default
atmSoftPVpcRetryFailures	Gauge32	N/A
atmSoftPVpcRetryLimit	0 through 65535	0
atmSoftPVpcRowStatus	RowStatus	N/A

SPVC addresses

The MIB table shown in Table 3-30 is used to configure ATM addresses before setting up SPVCCs or SPVPCs at an ATM interface. In addition, before setting up an SPVC at the source interface, you can consult this table at the destination interface. To make sure that no fields in the `atmInterfaceSoftPvcAddressTable` can be modified when the row is active, remove the row from service by using the following command:

```
SET RowStatus = notInService
```

You can then set objects to define the entry. When the objects have been modified, you can activate the row by using the following command:

```
SET RowStatus = active
```

Table 3-30. *atmInterfaceSoftPvcAddressTable*

MIB object	Value type or range	Default
atmInterfaceSoftPvcAddress	AtmAddr	N/A
atmInterfaceSoftPvcAddressRow-Status	RowStatus	N/A

Monitoring failing SPVCs

The MIB tables shown in Table 3-31 and Table 3-32 indicate all SPVCs for which the row status is active and the operational status is any value other than `connected`.

Table 3-31. *atmCurrentlyFailingSoftPVccTable*

MIB object	Value type or range	Default
atmCurrentlyFailingSoftPVccTimeStamp	TimeStamp	N/A

Table 3-32. *atmCurrentlyFailingSoftPVpcTable*

MIB object	Value type or range	Default
atmCurrentlyFailingSoftPVpcTimeStamp	TimeStamp	N/A

Supplemental managed objects for ATM management

Stinger units support the following tables from an IETF draft entitled *Definitions of Supplemental Managed Objects for ATM Management* (draft-ietf-atommib-atm2-13.txt), September 1999. The following copyright notice applies to these MIB definitions:

Copyright© The Internet Society (1999). All Rights Reserved. This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

ATM interface configuration extensions

The following table contains ATM interface configuration and monitoring information not defined in the atmInterfaceConfTable from the ATM-MIB.

Table 3-33. atmInterfaceExtTable

MIB object	Value type or range	Default
atmIntfConfigType	AtmInterfaceType	autoConfig (MIB default) PVC (Stinger default) Must set to atmFPnni1Dot0 to support PNNI on an interface.
atmIntfActualType	AtmInterfaceType	N/A
atmIntfConfigSide	other(1), user(2), network(3)	N/A (MIB default) other (Stinger default)
atmIntfActualSide	other(1), user(2), network(3), symmetric(4)	N/A
atmIntfIlmiAdminStatus	ilmi(0), ilmiAddressRegistration(1), ilmiConnectivity(2), ilmiPvcPvpMgmt(3), ilmiSigVccParamNegotiation(4)s	N/A (MIB default) down (Stinger default)
atmIntfIlmiOperStatus	ilmi(0), ilmiAddressRegistration(1), ilmiConnectivity(2), ilmiPvcPvpMgmt(3), ilmiSigVccParamNegotiation(4)	N/A

Management Information Bases (MIBs)
Supplemental managed objects for ATM management

Table 3-33. *atmInterfaceExtTable (continued)*

MIB object	Value type or range	Default
atmIntfIlmiFsmState	stopped(1), linkFailing(2), establishing(3), configuring(4), retrievingNetworkPrefixes(5), registeringNetworkPrefixes(6), retrievingAddresses(7), registeringAddresses(8), verifying(9)	N/A
atmIntfIlmiEstablishConPollIntvl	1 through 65535 seconds	1 (MIB default) ILMI, not yet supported (Stinger default)
atmIntfIlmiCheckConPollIntvl	0 through 65535 seconds	5
atmIntfIlmiConPollInactFactor	0 through 65535	4
atmIntfIlmiPublicPrivateIndctr	other(1), public(2), private(3)	private
atmInterfaceConfMaxSvpcVpi	0 through 4095	N/A (MIB default) 255 (Stinger default)
atmInterfaceCurrentMaxSvpcVpi	0 through 4095	N/A
atmInterfaceConfMaxSvccVpi	0 through 4095	N/A (MIB default) 255 (Stinger default)
atmInterfaceCurrentMaxSvccVpi	0 through 4095	N/A
atmInterfaceConfMinSvccVci	0 through 65535	N/A (MIB default) 32 (Stinger default)
atmInterfaceCurrentMinSvccVci	0 through 65535	N/A
atmIntfSigVccRxTrafficDescrIndex	AtmTrafficDescrParamIndex	N/A (MIB default) 2 (Stinger default)
atmIntfSigVccTxTrafficDescrIndex	AtmTrafficDescrParamIndex	N/A (MIB default) 2 (Stinger default)
atmIntfPvcFailures	Counter32	N/A
atmIntfCurrentlyFailingPVpls	Gauge32	N/A
atmIntfCurrentlyFailingPVcls	Gauge32	N/A
atmIntfPvcFailuresTrapEnable	TruthValue	false
atmIntfPvcNotificationInterval	1 through 3600 seconds	30

ATM cross-connects for SVCs and SPVCs

The ATM SVC virtual path (VP) and virtual channel (VC) cross-connect tables, shown in Table 3-34 and Table 3-35 respectively, contain configuration and state information of virtual

path and virtual channel cross-connects for SVCs and SPVCs. The terms Low and High in these variables refer to an ATM interface with a lower or higher `ifIndex` value. Note that `atmSvcVpCrossConnectRowStatus` and `atmSvcVcCrossConnectRowStatus` were changed from read-create to read-only access.

Table 3-34. *atmSvcVpCrossConnectTable*

MIB object	Value type or range	Default
<code>atmSvcVpCrossConnectIndex</code>	1 through 2147483647	N/A
<code>atmSvcVpCrossConnectLowIfIndex</code>	InterfaceIndex	N/A
<code>atmSvcVpCrossConnectLowVpi</code>	AtmVpIdentifier	N/A
<code>atmSvcVpCrossConnectHighIfIndex</code>	InterfaceIndex	N/A
<code>atmSvcVpCrossConnectHighVpi</code>	AtmVpIdentifier	N/A
<code>atmSvcVpCrossConnectCreationTime</code>	TimeStamp	N/A
<code>atmSvcVpCrossConnectRowStatus</code>	Read-create (MIB)	N/A
	Read-only (Stinger)	

Table 3-35. *atmSvcVcCrossConnectTable*

MIB object	Value type or range	Default
<code>atmSvcVcCrossConnectIndex</code>	1 through 2147483647	N/A
<code>atmSvcVcCrossConnectLowIfIndex</code>	InterfaceIndex	N/A
<code>atmSvcVcCrossConnectLowVpi</code>	AtmVpIdentifier	N/A
<code>atmSvcVcCrossConnectLowVci</code>	AtmVcIdentifier	N/A
<code>atmSvcVcCrossConnectHighIfIndex</code>	InterfaceIndex	N/A
<code>atmSvcVcCrossConnectHighVpi</code>	AtmVpIdentifier	N/A
<code>atmSvcVcCrossConnectHighVci</code>	AtmVcIdentifier	N/A
<code>atmSvcVcCrossConnectCreationTime</code>	TimeStamp	N/A
<code>atmSvcVcCrossConnectRowStatus</code>	Read-create (MIB)	N/A
	Read-only (Stinger)	

ATM interface signaling statistics

The MIB table Table 3-36 provides statistical information about signaling on an ATM interface.

Table 3-36. *atmSigStatTable*

MIB object	Value type or range	Default
<code>mSigSSCOPConEvents</code>	Counter32	N/A
<code>atmSigSSCOPErrdPdus</code>	Counter32	N/A
<code>atmSigDetectSetupAttempts</code>	Counter32	N/A

Management Information Bases (MIBs)
Supplemental managed objects for ATM management

Table 3-36. *atmSigStatTable* (continued)

MIB object	Value type or range	Default
atmSigEmitSetupAttempts	Counter32	N/A
atmSigDetectUnavailRoutes	Counter32	N/A
atmSigEmitUnavailRoutes	Counter32	N/A
atmSigDetectUnavailResrcs	Counter32	N/A
atmSigEmitUnavailResrcs	Counter32	N/A
atmSigDetectCldPtyEvents	Counter32	N/A
atmSigEmitCldPtyEvents	Counter32	N/A
atmSigDetectMsgErrors	Counter32	N/A
atmSigEmitMsgErrors	Counter32	N/A
atmSigDetectClgPtyEvents	Counter32	N/A
atmSigEmitClgPtyEvents	Counter32	N/A
atmSigDetectTimerExpires	Counter32	N/A
atmSigEmitTimerExpires	Counter32	N/A
atmSigDetectRestarts	Counter32	N/A
atmSigEmitRestarts	Counter32	N/A
atmSigInEstabls	Counter32	N/A
atmSigOutEstabls	Counter32	N/A

Monitoring active ATM circuits

The MIB, LUCENT-ATM-VC-STATS-MIB (or *atmvcstats.mib*) is an enterprise-specific MIB that enables an SNMP manager to monitor a Stinger unit's ATM network utilization. The unit reports information regarding active virtual path connections (VPCs) and active virtual channel connections (VCCs) to an SNMP manager.

This SNMP feature is applicable only if all the following conditions are met:

- The unit's configuration includes optional (or specially licensed) Network Management services.
- SNMP access is enabled on the unit.
- You must possess a MIB Browser.

Overview of the ATM status MIB

The MIB, LUCENT-ATM-VC-STATS-MIB includes four tables that accumulate and report active ATM circuit information:

- *atmVccStatsTable*—VCCs statistics indexed by slot, port, virtual path identifier (VPI), and virtual channel identifier (VCI)
- *atmIfVccStatsTable*—VCCs statistics indexed by the interface index (*ifIndex*) values of ATM interfaces, VPIs, and VCI

- `atmVpcStatsTable`—VPCs statistics indexed by slot, port, and VPI
- `atmIfVpcStatsTable`—VPCs statistics indexed by `ifIndex` values of the unit's ATM interfaces and VPIs

Two of the tables index the information using the interface index values of the unit's ATM interfaces. The `ifIndex` is a fixed value that the unit assigns for each interface at the time it performs a power-on self-test (POST). The tables contain 21 variables.

Overview of ATM status variables

With one exception, each of the variables has four possible variations within the tables listed above. For example, `atm[if][VCC|VPC]StatsCircuitName` describes a variable that can appear as follows:

- `atmVCCStatsCircuitName`—The name of an active VCC, in the `atmVccStatsTable`
- `atmifVCCStatsCircuitName`—The name of an active VCC, in the `atmifVccStatsTable`
- `atmVPCStatsCircuitName`—The name of an active VPC in the `atmVpcStatsTable`
- `atmifVPCStatsCircuitName`—The name of an active VPC in the `atmifVpcStatsTable`

The one exception is the `atm[if]VCCStatsOtherSideVci` variable, which applies only to active VCCs and has only two variations:

- `atmVCCStatsOtherSideVci`
- `atmifVCCStatsOtherSideVci`

AAL5 circuit information

Four of the variables include a zero (0) to indicate that ATM Adaptation Layer 5 (AAL5) information is used by the active ATM circuit, either on the local unit or on a remote unit. AAL5 is a protocol that translates higher-layer data from its native size and format to the size and format of an ATM cell, enabling engineers to adapt the ATM layer to particular services. The protocol's AAL5 provides guidelines by which a sending unit segments and a receiving unit reassembles data packets.

ATM cell information

Six of the variables report information regarding ATM cells. An ATM cell is a 53-byte fixed-length data packet consisting of a 5-byte header and a 48-byte payload. Each ATM cell that passes through the unit's connections includes a virtual path identifier (VPI) and a virtual channel identifier (VCI).

ATM QoS information

Two of the variables report quality of service (QoS) information. The QoS process measures, improves, and predicts data rates, error rates, and other facets of network transmission. QoS is particularly important for the transmission of high-bandwidth video and multimedia data. When you use the Resource Reservation Protocol (RSVP), you can use criteria prepared in

advance to expedite packets going through a gateway. ATM enables you to specify QoS properties by setting the values for cell loss ratio, cell transfer delay, and cell delay variation.

A QoS contract defines an ATM service category and related traffic characteristics. An ATM permanent virtual circuit (PVC) is typically assigned a QoS contract for each direction of the connection. The ATM switches that reside between the source and destination each meet a requested QoS as long as the end nodes comply with the negotiated QoS contract. The ATM network can make use of traffic-management capabilities, such as altering the characteristics of a cell stream, to meet the QoS objectives and ensure that the contract is enforced.

ATM status variables

Following are the definitions of the variables included with the LUCENT-ATM-VC-STATS-MIB:

Variable	Reports
<code>atm[if][VCC VPC]StatsCircuitName</code>	Name of the active ATM circuit. The unit uses the value you specify for the <code>station</code> parameter, located in a local <code>Connection</code> profile, or the <code>User-Name</code> attribute in a <code>RADIUS</code> connection profile.
<code>atm[if]VCC VPCStatsConnectionType</code>	Type of active ATM circuit: switched or terminating.
<code>atm[if]VCC VPCStatsVcState</code>	Whether the active circuit is currently transferring data.
<code>atm[if]VCC VPCStatsOtherSideSlotIndex</code>	Slot identifier associated with the remote end of the active ATM circuit. If the circuit is AAL5, this variable contains a zero value.
<code>atm[if]VCC VPCStatsOtherSidePort</code>	Port identifier associated with the remote end of the active ATM circuit. If the circuit is AAL5, this variable contains a zero value.
<code>atm[if]VCC VPCStatsOtherSideVpi</code>	VPI of the remote end of the active ATM circuit. If the circuit is AAL5, this variable contains a zero value.
<code>atm[if]VCCStatsOtherSideVci</code>	VCI of the remote end of the active ATM circuit. (Applicable to VCCs only.) If the circuit is AAL5, this variable contains a zero value.
<code>atm[if]VCC VPCStatsTrafficUpStreamContract</code>	QOS contract name for data transmitted from the CPE to the Stinger unit (or <i>upstream traffic</i>).
<code>atm[if]VCC VPCStatsTrafficDnStreamContract</code>	QOS contract name for data transmitted from the Stinger to the CPE (or <i>downstream traffic</i>).
<code>atm[if]VCC VPCStatsHCInCells</code>	Number of cells received.
<code>atm[if]VCC VPCStatsHCDiscardedCells</code>	Number of inbound cells received and discarded.
<code>atm[if]VCC VPCStatsHCInPackets</code>	Number of inbound AAL5 packets received.
<code>atm[if]VCC VPCStatsHCInDiscardedPackets</code>	Number of inbound AAL5 packets received and discarded.

Variable	Reports
atm[if]VCC VPCStatsHCInErrorPackets	Number of inbound packets received that have errors.
atm[if]VCC VPCStatsInCellsInLastPacket	Number of cells in the last inbound packet.
atm[if]VCC VPCStatsHCOutCells	Number of cells transmitted on this ATM circuit.
atm[if]VCC VPCStatsHCOutDiscardedCells	Number of outbound cells discarded on transmission.
atm[if]VCC VPCStatsHCOutPackets	Number of outbound AAL5 packets transmitted on the active ATM circuit.
atm[if]VCC VPCStatsHCOutDiscardedPackets	Number of outbound AAL5 packets discarded on transmission.
atm[if]VCC VPCStatsHCOutErrorPackets	Number of outbound AAL5 packets transmitted that have errors.
atm[if]VCC VPCStatsOutCellsInLastPacket	Number of cells in the last outbound packet.

ATM ILMI tables

The MIB tables shown in Table 3-37 and Table 3-38 specify the ATM network services known by this device, and per-interface network prefixes.

Table 3-37. *atmIlmiSrvcRegTable*

MIB object	Value type or range	Default
atmIlmiSrvcRegIndex	InterfaceIndexOrZero	N/A
atmIlmiSrvcRegServiceID	OBJECT IDENTIFIER	N/A
atmIlmiSrvcRegAddressIndex	1 through 2147483647	N/A
atmIlmiSrvcRegATMAddress	AtmAddr	N/A
atmIlmiSrvcRegParm1	OCTET STRING (SIZE(1 through 255))	N/A
atmIlmiSrvcRegRowStatus	RowStatus	N/A

Table 3-38. *atmIlmiNetworkPrefixTable*

MIB object	Value type or range	Default
atmIlmiNetPrefixPrefix	AtmIlmiNetworkPrefix	N/A
atmIlmiNetPrefixRowStatus	RowStatus	N/A

Currently failing permanent VPLs and VCLs

The *atmCurrentlyFailingPVplTable* and *atmCurrentlyFailingPVclTable* (shown in Table 3-39 and Table 3-40 respectively) report all VPLs and VCLs of type PVC that do not have an operator status of *up*, and the time at which the link failed.

Management Information Bases (MIBs)

ATM Forum PNNI MIB

Table 3-39. atmCurrentlyFailingPVplTable

MIB object	Value type or range	Default
atmCurrentlyFailingPVplTimeStamp	TimeStamp	N/A

Table 3-40. atmCurrentlyFailingPVclTable

MIB object	Value type or range	Default
atmCurrentlyFailingPVclTimeStamp	TimeStamp	N/A

ATM Forum PNNI MIB

The MIB for managing ATM Forum Private Network-to-Network Interface (PNNI) routing is defined in *Private Network-Network Interface Specification Version 1.0 (PNNI 1.0)*, af-pnni-0055.000, March 1996, and *PNNI v1.0 Errata and PICs*, af-pnni-0081.000, af-pnni-0081.000, *PNNI v1.0 Errata and PICs*, May 1997.

Table 3-41 defines version numbers, counters, and intervals related to PNNI operations.

Table 3-41. pnniBaseGroup

MIB object	Value type or range	Default
pnniHighestVersion	PnniVersion	N/A
pnniLowestVersion	PnniVersion	N/A
pnniDtlCountOriginator	Counter32	N/A
pnniDtlCountBorder	Counter32	N/A
pnniCrankbackCountOriginator	Counter32	N/A
pnniCrankbackCountBorder	Counter32	N/A
pnniAltRouteCountOriginator	Counter32	N/A
pnniAltRouteCountBorder	Counter32	N/A
pnniRouteFailCountOriginator	Counter32	N/A
pnniRouteFailCountBorder	Counter32	N/A
pnniRouteFailUnreachableOriginator	Counter32	N/A
pnniRouteFailUnreachableBorder	Counter32	N/A

PNNI node configuration

Table 3-42 contains values that affect the operation of a PNNI logical node.

Table 3-42. *pnniNodeTable*

MIB object	Value type or range	Default
pnniNodeIndex	0 through 65535	N/A (MIB default)
		1 (Stinger default)*
pnniNodeLevel	PnniLevel	96*
pnniNodeId	PnniNodeId	N/A*
pnniNodeLowest	TruthValue	true*
pnniNodeAdminStatus	up(1), down(2)	up
pnniNodeOperStatus	up(1), down(2)	N/A
pnniNodeDomainName	DisplayString	" "
pnniNodeAtmAddress	PnniAtmAddr	N/A*
pnniNodePeerGroupId	PnniPeerGroupId	N/A (MIB default)
		0 (Stinger default)*
pnniNodeRestrictedTransit	TruthValue	false
pnniNodeComplexRep	TruthValue	N/A (MIB default)
		false (Stinger default)
pnniNodeRestrictedBranching	TruthValue	N/A (MIB default)
		false (Stinger default)
pnniNodeDatabaseOverload	TruthValue	N/A
pnniNodePtses	Gauge32	N/A
pnniNodeRowStatus	RowStatus	N/A

* You must set pnniNodeAdminStatus to down to write these fields.

Table 3-43 contains peer group leader (PGL) election information for a PNNI node in this switching system.

Table 3-43. *pnniNodePglTable*

MIB object	Value type or range	Default
pnniNodePglLeadershipPriority	0 through 205	0
pnniNodeCfgParentNodeIndex	PnniNodeIndex	0
pnniNodePglInitTime	Integer32 (seconds)	15
pnniNodePglOverrideDelay	Integer32 (seconds)	30
pnniNodePglReelectTime	Integer32 (seconds)	15

Management Information Bases (MIBs)
ATM Forum PNNI MIB

Table 3-43. *pnniNodePglTable* (continued)

MIB object	Value type or range	Default
pnniNodePglState	starting(1), awaiting(2), awaitingFull(3), initialDelay(4), calculating(5), awaitUnanimity(6), operPgl(7), operNotPgl(8), hungElection(9), awaitReElection(10)	N/A (MIB default)
		starting (Stinger default)
pnniNodePreferredPgl	PnniNodeId	0
pnniNodePeerGroupLeader	PnniNodeId	0
pnniNodePglTimeStamp	TimeStamp	N/A (MIB default)
		0 (Stinger default)
pnniNodeActiveParentNodeId	PnniNodeId	0

Table 3-44 contains initial PNNI timer values and significant change thresholds.

Table 3-44. *pnniNodeTimerTable*

MIB object	Value type or range	Default
pnniNodePtseHolddown	Integer32 (100 milliseconds)	10
pnniNodeHelloHolddown	Integer32 (100 milliseconds)	10
pnniNodeHelloInterval	Integer32 (seconds)	15
pnniNodeHelloInactivityFactor	Integer32	5
pnniNodeHlinkInact	Integer32 (seconds)	120
pnniNodePtseRefreshInterval	Integer32 (seconds)	1800
pnniNodePtseLifetimeFactor	101 through 1000 (percent)	200
pnniNodeRxmtInterval	Integer32 (seconds)	5
pnniNodePeerDelayedAckInterval	Integer32 (100 milliseconds)	10
pnniNodeAvcrPm	1 through 99 (percent)	50
pnniNodeAvcrMt	1 through 99 (percent)	3
pnniNodeCdvPm	1 through 99 (percent)	25
pnniNodeCtdPm	1 through 99 (percent)	50

Table 3-45 contains variables related to SVCC-based routing control channels (RCCs).

Table 3-45. *pnniNodeSvccTable*

MIB object	Value type or range	Default
pnniNodeSvccInitTime	Integer32 (seconds)	4

Table 3-45. *pnniNodeSvccTable* (continued)

MIB object	Value type or range	Default
pnniNodeSvccRetryTime	Integer32 (seconds)	30
pnniNodeSvccCallingIntegrityTime	Integer32 (seconds)	35
pnniNodeSvccCalledIntegrityTime	Integer32 (seconds)	50
pnniNodeSvccTrafficDescriptorIndex	AtmTrafficDescrParamIndex	N/A (MIB default)
		0 (Stinger default)

Table 3-46 contains the mappings of membership and connection scope from organizational scope values used at UNI interfaces to PNNI routing level indicators.

Table 3-46. *pnniScopeMappingTable*

MIB object	Value type or range	Default
pnniScopeLocalNetwork	PnniLevel	96
pnniScopeLocalNetworkPlusOne	PnniLevel	96
pnniScopeLocalNetworkPlusTwo	PnniLevel	96
pnniScopeSiteMinusOne	PnniLevel	80
pnniScopeIntraSite	PnniLevel	80
pnniScopeSitePlusOne	PnniLevel	72
pnniScopeOrganizationMinusOne	PnniLevel	72
pnniScopeIntraOrganization	PnniLevel	64
pnniScopeOrganizationPlusOne	PnniLevel	64
pnniScopeCommunityMinusOne	PnniLevel	64
pnniScopeIntraCommunity	PnniLevel	48
pnniScopeCommunityPlusOne	PnniLevel	48
pnniScopeRegional	PnniLevel	32
pnniScopeInterRegional	PnniLevel	32
pnniScopeGlobal	PnniLevel	0

The summary address table, shown in Table 3-47, contains the summary address prefixes that can be advertised by the specified logical PNNI entity. This table's predecessor, the summary advertising table, has been deprecated (see "pnniSummaryTable (deprecated)" on page 3-39).

Table 3-47. *pnniSummaryAddressTable*

MIB object	Value type or range	Default
pnniSummaryAddressType	internal(1), exterior(2)	N/A
pnniSummaryAddressAddress	AtmAddrPrefix	N/A
pnniSummaryAddressPrefixLength	PnniPrefixLength	N/A
pnniSummaryAddressSuppress	TruthValue	false

Management Information Bases (MIBs)

ATM Forum PNNI MIB

Table 3-47. *pnniSummaryAddressTable* (continued)

MIB object	Value type or range	Default
<code>pnniSummaryAddressState</code>	<code>advertising(1)</code> , <code>suppressing(2)</code> , <code>inactive(3)</code>	N/A
<code>pnniSummaryAddressRowStatus</code>	<code>RowStatus</code>	N/A

PNNI interface configuration

The PNNI interface table, shown in Table 3-48, contains the attributes necessary to configure a physical interface that is capable of being used for PNNI routing. A row in this table is created for each interface that has an `ifType` of `atm(37)` or `atmLogical(80)`.

Table 3-48. *pnniIfTable*

MIB object	Value type or range	Default
<code>pnniIfNodeIndex</code>	1 through 65535	1
<code>pnniIfPortId</code>	<code>PnniPortId</code>	N/A
<code>pnniIfAggrToken</code>	<code>PnniAggrToken</code>	0
<code>pnniIfVPCapability</code>	<code>TruthValue</code>	true for <code>IfType atm(37)</code>
<code>pnniIfAdmWeightCbr</code>	1 through 16777215	5040
<code>pnniIfAdmWeightRtVbr</code>	1 through 16777215	5040
<code>pnniIfAdmWeightNrtVbr</code>	1 through 16777215	5040
<code>pnniIfAdmWeightAbr</code>	1 through 16777215	5040
<code>pnniIfAdmWeightUbr</code>	1 through 16777215	5040
<code>pnniIfRccServiceCategory</code>	<code>ServiceCategory</code>	N/A (MIB default) nrt-vbr (Stinger default)
<code>pnniIfRccTrafficDescrIndex</code>	<code>AtmTrafficDescrParamIndex</code>	N/A (MIB default) 2 (Stinger default)

PNNI metrics configuration

The PNNI metrics table, shown in Table 3-49, specifies a set of metrics that can be associated with the connectivity from a certain node and port to another node or port or to one or more reachable address prefixes and/or transit networks, for one (or more) particular service category(s). Note that multiple sets of parameters can have the same tag, in which case all sets apply to the specified connectivity.

Table 3-49. *pnniMetricsTable*

MIB object	Value type or range	Default
pnniMetricsTag	1 through 2147483647	N/A
pnniMetricsDirection	incoming(1), outgoing(2)	N/A
pnniMetricsIndex	1 through 2147483647	N/A
pnniMetricsClasses	0 through 31	N/A
pnniMetricsGcacClp	ClpType	N/A
pnniMetricsAdminWeight	1 through 16777215	5040
pnniMetrics1	Unsigned32	'FFFFFFFF'h
pnniMetrics2	Unsigned32	'FFFFFFFF'h
pnniMetrics3	Unsigned32	'FFFFFFFF'h
pnniMetrics4	Unsigned32	'FFFFFFFF'h
pnniMetrics5	Unsigned32	'FFFFFFFF'h
pnniMetrics6	Unsigned32	'FFFFFFFF'h
pnniMetrics7	Unsigned32	'FFFFFFFF'h
pnniMetrics8	Unsigned32	'FFFFFFFF'h
pnniMetricsRowStatus	RowStatus	N/A

PNNI route configuration

The routing base group, shown in Table 3-50, defines the number of current precalculated PNNI routes to valid nodes, addresses, and transit networks.

Table 3-50. *pnniRouteBaseGroup*

MIB object	Value type or range	Default
pnniRouteNodeNumber	Gauge32	N/A
pnniRouteAddrNumber	Gauge32	N/A

Table 3-51 contains the attributes that enable the PNNI entity to identify reachable ATM end system addresses (AESAs) and to determine which nodes are advertising this reachability. This table is also used to configure static routes to reachable address prefixes. Note that PnniRouteAddrInfo was changed from read-create to read-only access.

Table 3-51. *pnniRouteAddrTable*

MIB object	Value type or range	Default
pnniRouteAddrAddress	AtmAddrPrefix	N/A
pnniRouteAddrPrefixLength	PnniPrefixLength	N/A
pnniRouteAddrIndex	1 through 65535	N/A

Management Information Bases (MIBs)
ATM Forum PNNI MIB

Table 3-51. *pnniRouteAddrTable* (continued)

MIB object	Value type or range	Default
pnniRouteAddrIfIndex	InterfaceIndex	N/A (MIB default)
		0 (Stinger default)
pnniRouteAddrAdvertisingNodeId	PnniNodeId	N/A
pnniRouteAddrAdvertisedPortId	PnniPortId	0
pnniRouteAddrType	other(1), reject(2), internal(3), exterior(4)	exterior (4)
pnniRouteAddrProto	other(1), local(2), mgmt(3), pnni(4)	N/A
pnniRouteAddrPnniScope	PnniLevel	N/A (MIB default)
		Advertise at level of lowest level node (Stinger default)
pnniRouteAddrVPCapability	TruthValue	N/A
pnniRouteAddrMetricsTag	PnniMetricsTag	0
pnniRouteAddrPtseId	Unsigned32	N/A
pnniRouteAddrOriginateAdvertisement	TruthValue	true
pnniRouteAddrInfo	OBJECT IDENTIFIER (read-create: MIB)	zeroDotZero
	(Read-only: Stinger)	
pnniRouteAddrOperStatus	inactive(1), active(2), advertised(3)	N/A
pnniRouteAddrTimeStamp	TimeStamp	N/A
pnniRouteAddrRowStatus	RowStatus	N/A

Table 3-52 contains the attributes that enable the PNNI entity to identify reachable transit networks and to determine which nodes are advertising this reachability. This table is also used to add static routes to reachable transit networks. Note that PnniRouteTnsInfo was changed from read-create to read-only access.

Table 3-52. *pnniRouteTnsTable*

MIB object	Value type or range	Default
pnniRouteTnsType	TnsType	N/A
pnniRouteTnsPlan	TnsPlan	N/A
pnniRouteTnsId	DisplayString	N/A
pnniRouteTnsIndex	1 through 65535	N/A

Table 3-52. *pnniRouteTnsTable* (continued)

MIB object	Value type or range	Default
pnniRouteTnsIfIndex	InterfaceIndex	N/A
pnniRouteTnsAdvertisingNodeId	PnniNodeId	N/A
pnniRouteTnsAdvertisedPortId	PnniPortId	0
pnniRouteTnsRouteType	other(1), exterior(4)	exterior
pnniRouteTnsProto	other(1), local(2), mgmt(3), pnni(4)	N/A
pnniRouteTnsPnniScope	PnniLevel	N/A
pnniRouteTnsVPCapability	TruthValue	N/A
pnniRouteTnsMetricsTag	PnniMetricsTag	0
pnniRouteTnsPtseId	Unsigned32	N/A
pnniRouteTnsOriginateAdvertisement	TruthValue	true
pnniRouteTnsInfo	OBJECT IDENTIFIER (read-create: MIB) (Read-only: Stinger)	zeroDotZero
pnniRouteTnsOperStatus	inactive(1), active(2), advertised(3)	N/A
pnniRouteTnsTimeStamp	TimeStamp	N/A
pnniRouteTnsRowStatus	RowStatus	N/A

Read-only tables of dynamic link and node information

Because the information in these tables is discovered dynamically rather than configured, the objects are read-only.

Table 3-53 describes the operation of logical links and the relationship with the neighbor nodes on the other end of the links. Links can represent horizontal links between lowest-level neighboring peers, outside links, uplinks, or horizontal links to or from logical group nodes (LGNs).

Table 3-53. *pnniLinkTable*

MIB object	Value type or range	Default
pnniLinkPortId	PnniPortId	N/A
pnniLinkType	unknown(1), lowestLevelHorizontalLink(2), horizontalLinkToFromLgn(3), lowestLevelOutsideLink(4), uplink(5), outsideLink-AndUplink(6)	N/A

Management Information Bases (MIBs)

ATM Forum PNNI MIB

Table 3-53. *pnniLinkTable* (continued)

MIB object	Value type or range	Default
pnniLinkVersion	PnniVersion	N/A
pnniLinkHelloState	PnniHelloState	N/A
pnniLinkRemoteNodeId	PnniNodeId	N/A
pnniLinkRemotePortId	PnniPortId	N/A
pnniLinkDerivedAggrToken	PnniAggrToken	N/A
pnniLinkUpnodeId	PnniNodeId	N/A
pnniLinkUpnodeAtmAddress	PnniAtmAddress	N/A
pnniLinkCommonPeerGroupId	PnniPeerGroupId	N/A
pnniLinkIfIndex	InterfaceIndex	N/A
pnniLinkSvccRccIndex	PnniSvccRccIndex	N/A
pnniLinkRcvHellos	Counter32	N/A
pnniLinkXmtHellos	Counter32	N/A

Table 3-54 describes the relationship of a node with a neighboring node within the same peer group.

Table 3-54. *pnniNbrPeerTable*

MIB object	Value type or range	Default
pnniNbrPeerRemoteNodeId	PnniNodeId	N/A
pnniNbrPeerState	INTEGER	N/A
pnniNbrPeerSvccRccIndex	PnniSvccRccIndex	N/A
pnniNbrPeerPortCount	Gauge32	N/A
pnniNbrPeerRcvDbSums	Counter32	N/A
pnniNbrPeerXmtDbSums	Counter32	N/A
pnniNbrPeerRcvPtspS	Counter32	N/A
pnniNbrPeerXmtPtspS	Counter32	N/A
pnniNbrPeerRcvPtseReqs	Counter32	N/A
pnniNbrPeerXmtPtseReqs	Counter32	N/A
pnniNbrPeerRcvPtseAcks	Counter32	N/A
pnniNbrPeerXmtPtseAcks	Counter32	N/A

Table 3-55 describes all ports in Hello state 2-Way Inside to a given neighboring peer node. It is used only for lowest-level nodes.

Table 3-55. *pnniNbrPeerPortTable*

MIB object	Value type or range	Default
pnniNbrPeerPortId	PnniPortId	N/A
pnniNbrPeerPortFloodStatus	TruthValue	N/A

Table 3-56 contains the attributes used to analyze the operation of the PNNI protocol on SVCC-based routing control channels.

Table 3-56. *pnniSvccRccTable*

MIB object	Value type or range	Default
pnniSvccRccIndex	PnniSvccRccIndex	N/A
pnniSvccRccVersion	PnniVersion	N/A
pnniSvccRccHelloState	PnniHelloState	N/A
pnniSvccRccRemoteNodeId	PnniNodeId	N/A
pnniSvccRccRemoteAtmAddress	PnniAtmAddr	N/A
pnniSvccRccRcvHellos	Counter32	N/A
pnniSvccRccXmtHellos	Counter32	N/A
pnniSvccRccIfIndex	InterfaceIndex	N/A
pnniSvccRccVpi	0 through 4095	N/A
pnniSvccRccVci	0 through 65535	N/A

Table 3-57 describes the most recent instances of PNNI topology state elements (PTSEs) in a node's topology database.

Table 3-57. *pnniPtseTable*

MIB object	Value type or range	Default
pnniPtseOriginatingNodeId	PnniNodeId	N/A
pnniPtseId	Unsigned32	N/A
pnniPtseType	other(1), nodalStateParameters(96), nodalInformation(97), internalReachableAddresses(224), exteriorReachableAddresses(256), horizontalLinks(288), uplinks(289)	N/A
pnniPtseSequenceNum	Unsigned32	N/A
pnniPtseChecksum	Unsigned32 (0 through 65535)	N/A
pnniPtseLifeTime	Unsigned32 (0 through 65535)	N/A
pnniPtseInfo	OCTET STRING (SIZE(0 through 65535))	N/A

Management Information Bases (MIBs)

ATM Forum PNNI MIB

Table 3-58 contains attributes used to find and analyze the operation of all links and nodes within the PNNI hierarchy from the perspective of a local node.

Table 3-58. *pnniMapTable*

MIB object	Value type or range	Default
pnniMapOriginatingNodeId	PnniNodeId	N/A
pnniMapOriginatingPortId	PnniPortId	N/A
pnniMapIndex	0 through 65535	N/A
pnniMapType	horizontalLink(1), uplink(2), node(3)	N/A
pnniMapPeerGroupId	PnniPeerGroupId	N/A
pnniMapAggrToken	PnniAggrToken	N/A
pnniMapRemoteNodeId	PnniNodeId	N/A
pnniMapRemotePortId	PnniPortId	N/A
pnniMapVPCapability	TruthValue	N/A
pnniMapPtseId	Unsigned32	N/A
pnniMapMetricsTag	PnniMetricsTag	N/A

Table 3-59 contains a list of nodes from the perspective of a local node.

Table 3-59. *pnniMapNodeTable*

MIB object	Value type or range	Default
pnniMapNodeId	PnniNodeId	N/A
pnniMapNodePeerGroupId	PnniPeerGroupId	N/A
pnniMapNodeAtmAddress	PnniAtmAddr	N/A
pnniMapNodeRestrictedTransit	TruthValue	N/A
pnniMapNodeComplexRep	TruthValue	N/A
pnniMapNodeRestrictedBranching	TruthValue	N/A
pnniMapNodeDatabaseOverload	TruthValue	N/A
pnniMapNodeIAmLeader	TruthValue	N/A
pnniMapNodeLeadershipPriority	0 through 255	N/A
pnniMapNodePreferredPgl	PnniNodeId	N/A
pnniMapNodeParentNodeId	PnniNodeId	N/A
pnniMapNodeParentAtmAddress	PnniAtmAddr	N/A
pnniMapNodeParentPeerGroupId	PnniPeerGroupId	N/A
pnniMapNodeParentPglNodeId	PnniNodeId	N/A

Table 3-60 contains a list of all reachable addresses from each node visible to the local node.

Table 3-60. *pnniMapAddrTable*

MIB object	Value type or range	Default
pnniMapAddrAdvertisingNodeId	PnniNodeId	N/A
pnniMapAddrAdvertisedPortId	PnniPortId	N/A
pnniMapAddrIndex	1 through 2147483647	N/A
pnniMapAddrAddress	AtmAddrPrefix	N/A
pnniMapAddrPrefixLength	PnniPrefixLength	N/A

Table 3-61 contains a list of all reachable transit networks from each node visible to the local node.

Table 3-61. *pnniMapTnsTable*

MIB object	Value type or range	Default
pnniMapTnsAdvertisingNodeId	PnniNodeId	N/A
pnniMapTnsAdvertisedPortId	PnniPortId	N/A
pnniMapTnsType	TnsType	N/A
pnniMapTnsPlan	TnsPlan	N/A
pnniMapTnsId	DisplayString	N/A

Deprecated and unsupported PNNI tables

The summary table, shown in Table 3-62, has been deprecated. See “pnniSummaryAddressTable” on page 3-31 for related information.

Table 3-62. *pnniSummaryTable (deprecated)*

MIB object	Value type or range	Default
pnniSummaryAddress	AtmAddrPrefix	N/A
pnniSummaryPrefixLength	PnniPrefixLength	N/A
pnniSummaryType	internal(1), exterior(2)	N/A
pnniSummarySuppress	TruthValue	false
pnniSummaryState	advertising(1), suppressing(2), inactive(3)	N/A
pnniSummaryRowStatus	RowStatus	N/A

Management Information Bases (MIBs)

ATM Forum PNNI MIB

Table 3-63 defines the managed system's PNNI routing table of routes to other nodes. This table is not currently supported. All read-create objects have been changed to read-only.

Table 3-63. *pnniRouteNodeTable* (currently not supported)

MIB object	Value type or range	Default
pnniRouteNodeClass	ServiceCategory	N/A
pnniRouteNodeDestNodeId	PnniNodeId	N/A
pnniRouteNodeDTL	1 through 2147483647	N/A
pnniRouteNodeDestPortId	PnniPortId	0
pnniRouteNodeProto	other(1), local(2), mgmt(3), pnni(4)	N/A
pnniRouteNodeTimeStamps	TimeStamp	N/A
pnniRouteNodeInfo	OBJECT IDENTIFIER	zeroDotZero
pnniRouteNodeGcacClp	ClpType	N/A
pnniRouteNodeFwdMetricAW	Unsigned32	'FFFFFFFF'h
pnniRouteNodeFwdMetric1	Unsigned32	'FFFFFFFF'h
pnniRouteNodeFwdMetric2	Unsigned32	'FFFFFFFF'h
pnniRouteNodeFwdMetric3	Unsigned32	'FFFFFFFF'h
pnniRouteNodeFwdMetric4	Unsigned32	'FFFFFFFF'h
pnniRouteNodeFwdMetric5	Unsigned32	'FFFFFFFF'h
pnniRouteNodeFwdMetric6	Unsigned32	'FFFFFFFF'h
pnniRouteNodeFwdMetric7	Unsigned32	'FFFFFFFF'h
pnniRouteNodeFwdMetric8	Unsigned32	'FFFFFFFF'h
pnniRouteNodeBwdMetricAW	Unsigned32	'FFFFFFFF'h
pnniRouteNodeBwdMetric1	Unsigned32	'FFFFFFFF'h
pnniRouteNodeBwdMetric2	Unsigned32	'FFFFFFFF'h
pnniRouteNodeBwdMetric3	Unsigned32	'FFFFFFFF'h
pnniRouteNodeBwdMetric4	Unsigned32	'FFFFFFFF'h
pnniRouteNodeBwdMetric5	Unsigned32	'FFFFFFFF'h
pnniRouteNodeBwdMetric6	Unsigned32	'FFFFFFFF'h
pnniRouteNodeBwdMetric7	Unsigned32	'FFFFFFFF'h
pnniRouteNodeBwdMetric8	Unsigned32	'FFFFFFFF'h
pnniRouteNodeVPCapability	TruthValue	N/A
pnniRouteNodeStatus	RowStatus	N/A

Table 3-64 describes the set of all designated transit list (DTL) stacks used for the precomputed routes maintained by the managed system. This table is not currently supported. All read-create objects have been changed to read-only.

Table 3-64. *pnniDTLTable* (currently not supported)

MIB object	Value type or range	Default
pnniDTLIndex	1 through 2147483647	N/A
pnniDTLEntryIndex	1 through 200	N/A
pnniDTLNodeId	PnniNodeId	N/A
pnniDTLPortId	PnniPortId	N/A
pnniDTLLinkType	invalid(1), horizontal(2), uplink(3), last(4)	N/A
pnniDTLStatus	RowStatus	N/A

ATM interface status MIB

The ATM interface status MIB tables provide information about active ATM interfaces. Because the information in these tables is discovered dynamically rather than configured, the objects are read-only.

Table 3-65 contains parameters for the atmIfStatus profile. All objects in the table are preceded by the following string:

atmifstatus

Table 3-65. *mibatmIfStatus* profile table

MIB object	Value type or range	Default
Shelf-o	Integer	N/A
Item-o	Integer	N/A
LogicalItem-o	Integer	N/A
Address-PhysicalAddress-Shelf	Integer	0 0 0
PhysicalAddress-Slot	Integer	N/A
Address-PhysicalAddress-ItemNumber	Integer	N/A
Address-LogicalItem	Integer	N/A
IfNumber	Integer	N/A
NailedGroup	Integer	0
PortState	notConfigured (1), up (2), down (3)	N/A
SignallingState	notConfigured (1), up (2), down (3)	N/A
PnniLinkState	notConfigured (1), up (2), down (3)	N/A

Management Information Bases (MIBs)

ATM interface status MIB

Table 3-65. *mibatmIfStatus* profile table (continued)

MIB object	Value type or range	Default
IlmiLinkState	notConfigured (1), up (2), down (3)	N/A
Action-o	noAction (1), createProfile (2), deleteProfile (3)	N/A

DS3-ATM line status

Table 3-66 contains parameters for the *ds3AtmLineStatus* profile. All objects in the table are preceded by the following string:

ds3AtmLineStatus

Table 3-66. *ds3AtmLineStatus* table

MIB object	Value type or range	Default
Shelf-o	Integer	N/A
Slot-o	Integer	N/A
Item-o	Integer	N/A
PhysicalAddress-Shelf	Integer	N/A
PhysicalAddress-Slot	Integer	N/A
PhysicalAddress-ItemNumber	Integer	N/A
LineState	doesNotExist (1), disabled (2), lossOfSignal (3) lossOfFrame (4), yellowReceive (5), aisReceive (6), active (7)	N/A
SparePhysicalAddress-Shelf	Integer	N/A
SparePhysicalAddress-Slot	Integer	N/A
SparePhysicalAddress-ItemNumber	sparingNone (1), primaryActive (2), primaryInactive (3), secondaryActive (4), secondaryInactive (5)	0
SparingChangeReason	unknown (1), manual (2), automatic (3), test (4)	N/A
SparingChangeTime	Integer	N/A
SparingChangeCounter	Integer	N/A
VpiVciRange	Integer	N/A
VcSwitchingVpi	Display string	0
FBitErrorCount	Integer	N/A
PBitErrorCount	Integer	N/A
CpBitErrorCount	Integer	N/A
FebErrorCount	Integer	N/A
BpvErrorCount	Integer	N/A

Table 3-66. *ds3AtmLineStatus* table (continued)

MIB object	Value type or range	Default
LossOfSignal	false (1), true (2)	N/A
LossOfFrame	false (1), true (2)	N/A
YellowReceive	false (1), true (2)	N/A
AisReceive	false (1), true (2)	N/A
Action-o	noAction (1), createProfile (2), deleteProfile (3)	N/A

E3-ATM line status

Table 3-67 contains parameters for the *e3AtmLineStatus* profile. All objects in the table are preceded by the following string:

e3AtmLineStatus

Table 3-67. *e3AtmLineStatus* profile table

MIB object	Value type or range	Default
Shelf-o	Integer	N/A
Slot-o	Integer	N/A
Item-o	Integer	N/A
Shelf	Integer	N/A
Slot	Integer	N/A
ItemNumber	Integer	N/A
LineState	doesNotExist (1), disabled (2), lossOfSignal (3) lossOfFrame (4), yellowReceive (5), aisReceive (6), active (7)	N/A
SparePhysicalAddress-Shelf	Integer	N/A
SparePhysicalAddress-Slot	Integer	N/A
SparePhysicalAddress-ItemNumber	Integer	N/A
SparingState	sparingNone (1), primaryActive (2), primaryInactive (3), secondaryActive (4), secondaryInactive (5)	sparingNone (1)
SparingChangeReason	unknown (1), manual (2), automatic (3), test (4)	N/A
SparingChangeTime	Integer	N/A
SparingChangeCounter	Integer	N/A
VpiVciRange	Integer	N/A
VcSwitchingVpi	Integer	0

Management Information Bases (MIBs)

ATM interface status MIB

Table 3-67. e3AtmLineStatus profile table (continued)

MIB object	Value type or range	Default
FBitErrorCount	Integer	N/A
PBitErrorCount	Integer	N/A
CpBitErrorCount	Integer	N/A
FebErrorCount	Integer	N/A
BpvErrorCount	Integer	N/A
LossOfSignal	false (1), true (2)	N/A
LossOfFrame	false (1), true (2)	N/A
YellowReceive	false (1), true (2)	N/A
AisReceive	false (1), true (2)	N/A
Action-o	noAction (1), createProfile (2), deleteProfile (3)	N/A

OC3-ATM line status

Table 3-68 through Table 3-70 contain parameters for the oc3AtmLineStatus profile. All objects in Table 3-68 are preceded by the following string:

oc3AtmLineStatus

Table 3-68. oc3AtmLineStatus profile table

MIB object	Value type or range	Default
Shelf-o	Integer	N/A
Slot-o	Integer	N/A
Item-o	Integer	N/A
PhysicalAddress-Shelf	Integer	N/A
PhysicalAddress-Slot	Integer	N/A
PhysicalAddress-ItemNumber	Integer	N/A
LineState	doesNotExist (1), disabled (2), lossOfSignal (3) lossOfFrame (4), yellowReceive (5), aisReceive (6), active (7)	N/A
SparePhysicalAddress-Shelf	Integer	N/A
SparePhysicalAddress-Slot	Integer	N/A
SparePhysicalAddress-ItemNumber	Integer	N/A
SparingState	sparingNone (1), primaryActive (2), primaryInactive (3), secondaryActive (4), secondaryInactive (5)	N/A

Table 3-68. *oc3AtmLineStatus* profile table (continued)

MIB object	Value type or range	Default
SparingChangeReason	unknown (1), manual (2), automatic (3), test (4)	N/A
SparingChangeTime	Integer	N/A
SparingChangeCounter	Integer	N/A
VpiVciRange	Integer	N/A
VcSwitchingVpi	Integer	N/A
ApsConfigName	Display string	N/A
ApsChannelStatus	unknown (1), onProtection (2), onWorking (3), onWorking (3), lockedOutWorking (4)	
ApsChannelSdCondition	false (1), true (2)	
ApsChannelSfCondition	false (1), true (2)	
ApsChannelLowDirection	lowNone (1), lowRecv (2), lowSend (3), lowBoth (4)	
ApsChannelRecvSdCount	Integer	N/A
ApsChannelRecvSfCount	Integer	N/A
ApsChannelRecvAisCount	Integer	N/A
ApsChannelRecvRdiCoun	Integer	N/A
LossOfSignal	Integer	N/A
LossOfFrame	false (1), true (2)	
OutOfFrame	false (1), true (2)	
SectionState	sonetDisabled (1), sonetSectionActiveNoDefect (2), sonetSectionLossOfSignal (3), sonetSectionLossOfFrame (4)	
PathState	sonetDisabled (1), sonetPathActiveNoDefect (2), sonetPathLossOfPointer (3), sonetPathAlarmIndicationSignal (4), sonetPathRdi (5), sonetPathUnquipped (6), sonetPathSignalLabelMismatch (7)	
AisReceive	false (1), true (2)	
YellowReceive	false (1), true (2)	
OutOfCellDelineation	false (1), true (2)	
LossOfCellDelineation	false (1), true (2)	
ApsReceive	false (1), true (2)	
RsopBipErrorCoun	Integer	N/A
RlopBipErrorCount	Integer	N/A

Management Information Bases (MIBs)

ATM interface status MIB

Table 3-68. *oc3AtmLineStatus* profile table (continued)

MIB object	Value type or range	Default
RlopFebeErrorCount	Integer	N/A
RpopBipErrorCount	Integer	N/A
RpopFebeErrorCount	Integer	N/A
RacpChcsErrorCount	Integer	N/A
RacpUchcsErrorCount	Integer	N/A
RacpRxCellCount	Integer	N/A
TacpTxCellCount	Integer	N/A
FrequencyJustificationCount	Integer	N/A
oHECCellDropCounter	Integer	N/A
oFIFOOverflowCounter	Integer	N/A
IdleCellCounter	Integer	N/A
ValidCellCounter	Integer	N/A
TimeElapsed	Integer	N/A

All objects in Table 3-69 are preceded by the following string:

oc3AtmLineStatus-PerformanceMonitoring

Table 3-69. *oc3AtmLineStatus* profile table

MIB object	Value type or range	Default
SonetSectionErroredSeconds	Integer	N/A
SonetSectionSeverelyErroredSeconds		N/A
SonetSectionSeverelyErroredFramingSeconds	Integer	N/A
SonetSectionCodingViolations	Integer	N/A
SonetLineErroredSeconds	Integer	N/A
SonetLineCodingViolations	Integer	N/A
SonetLineUnavailableSeconds	Integer	N/A
SonetFarEndLineErroredSeconds	Integer	N/A
SonetFarEndLineSeverelyErroredSeconds	Integer	N/A
SonetFarEndLineCodingViolations	Integer	N/A
SonetFarEndLineUnavailableSeconds	Integer	N/A
SonetPathErroredSeconds	Integer	N/A
SonetPathSeverelyErroredSeconds	Integer	N/A
SonetPathCodingViolations	Integer	N/A
SonetPathUnavailableSeconds	Integer	N/A
SonetFarEndPathErroredSeconds	Integer	N/A

Table 3-69. *oc3AtmLineStatus* profile table (continued)

MIB object	Value type or range	Default
SonetFarEndPathSeverelyErroredSeconds	Integer	N/A
SonetFarEndPathCodingViolations	Integer	N/A
SonetFarEndPathUnavailableSeconds	Integer	N/A
Action-o	noAction (1), createProfile (2), deleteProfile (3)	

All objects in Table 3-70 are preceded by the following string:

oc3AtmLineStatus__interval_performance_monitoring

Table 3-70. *oc3AtmLineStatus* profile table

MIB object	Value type or range	Default
Shelf-o	Integer	N/A
Slot-o	Integer	N/A
Item-o	Integer	N/A
Index-o	Integer	N/A
SonetSectionErroredSeconds	Integer	N/A
SonetSectionSeverelyErroredSeconds	Integer	N/A
SonetSectionSeverelyErroredFramingSeconds	Integer	N/A
SonetSectionCodingViolations	Integer	N/A
SonetLineErroredSeconds	Integer	N/A
SonetLineSeverelyErroredSeconds	Integer	N/A
SonetLineCodingViolations	Integer	N/A
SonetLineUnavailableSeconds	Integer	N/A
SonetFarEndLineErroredSeconds	Integer	N/A
SonetFarEndLineSeverelyErroredSeconds	Integer	N/A
SonetFarEndLineCodingViolations	Integer	N/A
SonetFarEndLineUnavailableSeconds	Integer	N/A
SonetPathErroredSecond	Integer	N/A
SonetPathSeverelyErroredSeconds	Integer	N/A
SonetPathCodingViolation	Integer	N/A
SonetPathUnavailableSeconds	Integer	N/A
SonetFarEndPathErroredSeconds	Integer	N/A
SonetFarEndPathSeverelyErroredSeconds	Integer	N/A
SonetFarEndPathCodingViolations	Integer	N/A
SonetFarEndPathUnavailableSeconds	Integer	N/A

SONET extension MIB

This MIB provides extensions to the SONET MIB (RFC 2558) and describes SONET/SDH interface objects. The objects in the MIB contain various statistics that are collected at 24-hour intervals.

The SONET extension MIB tables provide information about active ATM interfaces. Because the information in these tables is discovered dynamically rather than configured, the objects are read-only.

The SONET extension MIB includes the following groups:

- sonetExtSection
- sonetExtLine
- sonetExtFarEndLine
- sonetExtPath
- sonetExtFarEndPath

sonetExtSection group

The sonetExtSection group handles SONET/SDH section statistics and consists of the following two tables:

- sonetExtSectionCurr1DayTable
- sonetExtSection1DayIntervalTable

The sonetExtSectionCurr1DayTable, shown in Table 3-71, contains current day performance information for each section interface.

Table 3-71. sonetExtSectionCurr1DayTable

MIB object	Value type or range	Default
sonetExtSectionCurr1DayTimeElapsed	Integer32 (0..86400)	N/A
sonetExtSectionCurr1DayESS	PerfCurrentCount	N/A
sonetExtSectionCurr1DaySESS	PerfCurrentCount	N/A
sonetExtSectionCurr1DaySEFSS	PerfCurrentCount	N/A
sonetExtSectionCurr1DayCVs	PerfCurrentCount	N/A

The sonetExtSection1DayIntervalTable shown in Table 3-72 provides one row for each performance interval.

Table 3-72. sonetExtSection1DayIntervalTable

MIB object	Value type or range	Default
sonetExtSection1DayIntervalNumber	Integer	N/A
sonetExtSection1DayIntervalESS	PerfCurrentCount	N/A
sonetExtSection1DayIntervalSESS	PerfCurrentCount	N/A
sonetExtSection1DayIntervalSEFSS	PerfCurrentCount	N/A
sonetExtSection1DayIntervalCVs	PerfCurrentCount	N/A

sonetExtLine group

The sonetExtLine group handles SONET/SDH line statistics and includes the following two tables:

- sonetExtLineCurr1DayTable
- sonetExtLine1DayIntervalTable

The sonetExtLineCurr1DayTable, shown in Table 3-73, contains current day performance information for each section interface.

Table 3-73. sonetExtLineCurr1DayTable

MIB object	Value type or range	Default
sonetExtLineCurr1DayTimeElapsed	Integer32 (0..86400)	N/A
sonetExtLineCurr1DayESS	PerfCurrentCount	N/A
sonetExtLineCurr1DaySESS	PerfCurrentCount	N/A
sonetExtLineCurr1DayCVs	PerfCurrentCount	N/A
sonetExtLineCurr1DayUASS	PerfCurrentCount	N/A

The sonetExtLine1DayIntervalTable, shown in Table 3-74, provides one row for each performance interval.

Table 3-74. sonetExtLine1DayIntervalTable

MIB object	Value type or range	Default
sonetExtLin1DayIntervalNumber	Integer32 (1..30)	N/A
sonetExtLin1DayIntervaleSS	PerfCurrentCount	N/A
sonetExtLin1DayIntervalSESS	PerfCurrentCount	N/A
sonetExtLin1DayIntervalCVs	PerfCurrentCount	N/A
sonetExtLin1DayIntervalUASS	PerfCurrentCount	N/A

sonetExtFarEndLine group

The sonetExtFarEndLine group handles SONET/SDH FarEndLine statistics and includes the following two tables:

- sonetExtFarEndLineCurr1DayTable
- sonetExtFarEndLine1DayIntervalTable

The sonetExtFarEndLineCurr1DayTable, shown in Table 3-75, contains current day performance information for each section interface.

Table 3-75. sonetExtFarEndLineCurr1DayTable

MIB object	Value type or range	Default
sonetExtFarEndLineCurr1DayTimeElapsed	Integer32 (0..86400)	N/A
sonetExtFarEndLineCurr1DayESS	PerfCurrentCount	N/A
sonetExtFarEndLineCurr1DaySESS	PerfCurrentCount	N/A

Management Information Bases (MIBs)

SONET extension MIB

Table 3-75. *sonetExtFarEndLineCurr1DayTable* (continued)

MIB object	Value type or range	Default
sonetExtFarEndLineCurr1DayCVs	PerfCurrentCount	N/A
sonetExtFarEndLineCurr1DayUASs	PerfCurrentCount	N/A

The *sonetExtFarEndLine1DayIntervalTable*, shown in Table 3-76, provides one row for each performance interval.

Table 3-76. *sonetExtFarEndLine1DayIntervalTable*

MIB object	Value type or range	Default
sonetExtFarEndLine1DayIntervalNumber	Integer32 (1..30)	N/A
sonetExtFarEndLine1DayIntervalESs	PerfCurrentCount	N/A
sonetExtFarEndLine1DayIntervalSESSs	PerfCurrentCount	N/A
sonetExtFarEndLine1DayIntervalCVs	PerfCurrentCount	N/A
sonetExtFarEndLine1DayIntervalUASs	PerfCurrentCount	N/A

sonetExtPath group

The *sonetExtPath* group handles the SONET/SDH path statistics and includes the following two tables:

- *sonetExtPathCurr1DayTable*
- *sonetExtPath1DayIntervalTable*

The *sonetExtPathCurr1DayTable*, shown in Table 3-77, contains current day performance information for each section interface.

Table 3-77. *sonetExtPathCurr1DayTable*

MIB object	Value type or range	Default
sonetExtPathCurr1DayTimeElapsed	Integer32 (0..86400)	N/A
sonetExtPathCurr1DayESs	PerfCurrentCount	N/A
sonetExtPathCurr1DaySESSs	PerfCurrentCount	N/A
sonetExtPathCurr1DayCVs	PerfCurrentCount	N/A
sonetExtPathCurr1DayUASs	PerfCurrentCount	N/A

The *sonetExtPath1DayIntervalTable*, shown in Table 3-78, provides one row for each performance interval.

Table 3-78. *sonetExtPath1DayIntervalTable*

MIB object	Value type or range	Default
sonetExtPath1DayIntervalNumber	Integer32 (1..30)	N/A
sonetExtPath1DayIntervalESs	PerfCurrentCount	N/A
sonetExtPath1DayIntervalSESSs	PerfCurrentCount	N/A

Table 3-78. *sonetExtPath1DayIntervalTable* (continued)

MIB object	Value type or range	Default
sonetExtPath1DayIntervalCVs	PerfCurrentCount	N/A
sonetExtPath1DayIntervalUASs	PerfCurrentCount	N/A

sonetExtFarEndPath group

The `sonetExtFarEndPath` group handles SONET/SDH `FarEndPath` statistics and includes the following two tables:

- `sonetExtFarEndPathCurr1DayTable`
- `sonetExtFarEndPath1DayIntervalTable`

The `sonetExtFarEndPathCurr1DayTable`, shown in Table 3-79, contains current day performance information for each section interface.

Table 3-79. *sonetExtFarEndPathCurr1DayTable*

MIB object	Value type or range	Default
sonetExtFarEndPathCurr1DayTimeElapsed	Integer32 (0..86400)	N/A
sonetExtFarEndPathCurr1DayESs	PerfCurrentCount	N/A
sonetExtFarEndPathCurr1DaySESs	PerfCurrentCount	N/A
sonetExtFarEndPathCurr1DayCVs	PerfCurrentCount	N/A
sonetExtFarEndPathCurr1DayUASs	PerfCurrentCount	N/A

The `sonetExtFarEndPath1DayIntervalTable`, shown in Table 3-80, provides one row for each performance interval.

Table 3-80. *sonetExtFarEndPath1DayIntervalTable*

MIB object	Value type or range	Default
sonetExtFarEndPath1DayIntervalNumber	Integer32 (1..30)	N/A
sonetExtFarEndPath1DayIntervalESs	PerfCurrentCount	N/A
sonetExtFarEndPath1DayIntervalSESs	PerfCurrentCount	N/A
sonetExtFarEndPath1DayIntervalCVs	PerfCurrentCount	N/A
sonetExtFarEndPath1DayIntervalUASs	PerfCurrentCount	N/A

Stinger MIB Load Sequence

The Stinger SNMP MIBs are located in the same directory as the software release. To download the MIBs, log in to Lucent OnLine Customer Support at <http://www.lucent.com/support>.

The following list shows the Lucent Access Enterprise MIBs and industry-standard MIBs distributed with TAOS. The MIBs are listed in the recommended load sequence, which satisfies internal MIB dependencies.

Load sequence	MIB	Description
1	rfc2578.mib	SNMPv2-SMI
2	rfc2579.mib	SNMPv2-TC
3	rfc1213.mib	MIB-II
4	ianaiftype.mib	IANA ifType
5	rfc2233.mib	Interface Group MIB using SMIV2
6	rfc1317.mib	RS-232 MIB
7	rfc2495.mib	DS1 MIB
8	rfc2496.mib	DS3 MIB
9	rfc1315.mib	Frame Relay MIB
10	rfc1696.mib	Modem MIB
11	rfc1850.mib	OSPF MIB
12	rfc1850_2.mib	OSPF TRAP MIB
13	rfc2558.mib	SONET MIB
14	rfc2514.mib	ATM-TC MIB
15	rfc2515.mib	ATM MIB
16	rfc1398.mib	Ethernet-like MIB (dot3)
17	rfc1724.mib	RIP 2 MIB
18	rfc1215.trp	SNMP generic traps
19	rfc2571.mib	SNMPV3 Framework MIB
20	rfc2572.mib	SNMPV3 Message Processing And Dispatching MIB
21	rfc2574.mib	SNMPV3 User Based Security Model MIB
22	rfc2573_1.mib	SNMPv3 Management Target MIB
23	rfc2573_2.mib	SNMPv3 Notification MIB

Stinger MIB Load Sequence

Load sequence	MIB	Description
24	rfc2493.mib	Performance History TC MIB
25	rfc2662_1.mib	ADSL TC MIB
26	rfc2662_2.mib	ADSL Line MIB
27	atmftc.mib	ATM FORUM Definitions
28	atmfspvc.mib	ATM FORUM Soft PVC MIB
29	atmfm4.mib	ATM FORUM M4 MIB
30	atmfpnni.mib	ATM FORUM PNNI MIB
31	atm2supp.mib	ATM Supplemental MIB
32	atmfima.mib	IMA MIB
33	ascend.mib	Lucent Access MIB
34	event.mib	Lucent Access Event MIB
35	call.mib	Lucent Access Call MIB
36	session.mib	Lucent Access Session MIB
37	lmodem.mib	Lucent Access LAN Modem MIB
38	radius.mib	Lucent Access RADIUS MIB
39	mcast.mib	Lucent Access Multicast MIB
40	wan.mib	Lucent Access Wan MIB
41	firewall.mib	Lucent Access Firewall Control Protocol MIB
42	sdsl.mib	Lucent Access SDSL interface MIB
43	adslcap.mib	Lucent Access ADSL CAP interface MIB
44	adslgmt.mib	Lucent Access ADSL DMT interface MIB
45	wandialout.mib	Lucent Access Wan Dialout Pkt MIB
46	ps.mib	Lucent Access Power Supply MIB
47	ms.mib	Lucent Access Multi-Shelf MIB
48	advanced.mib	Lucent Access Advanced Agent MIB
49	flash.mib	Lucent Access Flash MIB
50	misc.mib	Lucent Access Miscellaneous MIB
51	atmp.mib	Lucent Access ATMP MIB
52	asndl2tp.mib	Lucent Access L2TP MIB
53	srvcgmt.mib	Lucent Access Service Management MIB
54	call_log.mib	Lucent Access Call Logging MIB
55	mibfrml.mib	Lucent Access Frame Relay profile MIB
56	mibvdslnet.mib	Lucent Access VDSL profile MIB
57	mibds3net.mib	Lucent Access DS3 profile MIB
58	mibuds3net.mib	Lucent Access UDS3 profile MIB

Load sequence	MIB	Description
59	mibsdslnet.mib	Lucent Access SDSL profile MIB
60	mibcadslnet.mib	Lucent Access ADSL-CAP profile MIB
61	mibdadslnet.mib	Lucent Access ADSL-DMT profile MIB
62	mibdmtalnet.mib	Lucent Access CELL-DMT-ADSL profile MIB
63	mibanswer.mib	Lucent Access Answer profile MIB
64	mibinet.mib	Lucent Access Internet profile MIB
65	miboc3atmnet.mib	Lucent Access OC3 ATM profile MIB
66	mibds3atmnet.mib	Lucent Access DS3 ATM profile MIB
67	mibe3atmnet.mib	Lucent Access E3 ATM profile MIB
68	mibredundancy.mib	Lucent Access Redundancy profile MIB
69	mibredundancystats.mib	Lucent Access Redundancy Stats profile MIB
70	mibhdsl2net.mib	Lucent Access HDSL2 profile MIB
71	mibpctfi.mib	Lucent Access PCTFI profile MIB
72	mibpctfistat.mib	Lucent Access PCTFI STAT profile MIB
73	mibxdslslotc.mib	Lucent Access XDSL-SLOT-CONFIG profile MIB
74	mibatmqos.mib	Lucent Access ATM QOS profile MIB
75	cltm.mib	Lucent Access Copper Loop Test profile MIB
76	chassis.mib	Lucent Access Chassis MIB
77	resource.mib	Lucent Access Resource Management MIB
78	rfc1657.mib	BGP MIB
79	rfc1493.mib	BRIDGE MIB
80	watchdog.mib	Lucent Access Watchdog MIB
81	voip.mib	Lucent Access VOIP MIB
82	voipstats.mib	VoIP MIB statistics
83	sparing.mib	Lucent Access LIM and Trunk Sparing Status MIB
84	mibsparing.mib	Lucent Access LIM Sparing Configuration MIB
85	mgstat.mib	Media Gateway Status MIB
86	hdl2.mib	Lucent Access HDSL2 interface MIB
87	debug.mib	Lucent Access Debug MIB
88	eocstat.mib	Lucent Access xDSL EOC DB MIB
89	ping.mib	Remote PING MIB
90	cardcode.mib	Lucent Access card code MIB
91	hashcode.mib	Lucent Access hash code MIB
92	mibatmprefix.mib	Lucent Access ATM Address Generation Prefix
93	mibimagroup.mib	Lucent Access mibimaGroupConfig MIB

Stinger MIB Load Sequence

Load sequence	MIB	Description
94	mibimahw.mib	Lucent Access mibimaHwConfig MIB
95	mibds1atmnet.mib	Lucent Access mibds1AtmNetworkprofile MIB
96	mibatmintnet.mib	Lucent Access ATM Internal profile MIB
97	mibbrilt.mib	Lucent Access IDSL profile MIB
98	mibstat.mib	Lucent Access IDSL Statistic MIB
99	mibbwallocc.mib	Lucent Access Stinger ATM Config profile MIB
100	mibstaticcontroller.mib	Lucent Access Stinger Static Controller profile MIB
101	mibstatichsslot.mib	Lucent Access Stinger Static High Speed Slot profile MIB
102	mibstaticslot.mib	Lucent Access Stinger Static Slot profile MIB
103	ascend.trp	Lucent Access traps
104	ascendv3.trp	Lucent Access traps

Index

A

addresses

- configuration of ATM addresses, 2-11
- default PNNI node address, 2-18
- destination IP for terminating PVC, 2-16
- PNNI node ID, 2-18
- PNNI peer group ID, 2-18
- reachable, 3-33
- SPVC source and target, 2-11
- summary, 3-31
- system-generated for ATM interfaces, 2-11

ANSI standards, xv

asymmetric DSL (ADSL) ifType 94, 1-1

ATM

- address configuration, 2-11
- configuration overview, 1-1
- interface types, 1-1
- interfaces, ifIndex value, 1-2
- stack diagram, 1-1

ATM adaptation layer 5 (AAL5) ifType 49, 1-2

ATM ifType 37, 1-2

ATM interface status MIB

- MIB ATM interface status, 3-41

ATM interfaces

- atmInterfaceConfTable, 3-13
- atmInterfaceDs3PlcpTable, 3-14
- atmInterfaceExtTable, 3-21
- atmInterfaceTCTable, 3-14
- configuration, 1-2, 3-13
- configuring VPI-VCI ranges, 2-3
- configuring VPLs and VCLs, 3-15
- default addresses, 2-11
- finding hte ifIndex, 2-3
- ILMI, 3-27
- signaling statistics, 3-23
- stacked interfaces, 1-3
- system-generated AESAs, 2-11

ATM specifications, xvi

- atmCurrentlyFailingSoftPVccTable, 3-20
- atmCurrentlyFailingSoftPVpcTable, 3-20
- atmfM4VpNextVciTable, 3-18
- atmfM4VpNextVpiTable, 3-18
- atmIlmiNetworkPrefixTable, 3-27
- atmIlmiSrcvRegTable, 3-27

atmInterfaceConfTable, 3-13

atmInterfaceDs3PlcpTable, 3-14

atmInterfaceExtTable, 3-21

atmInterfaceSoftPvcAddressTable, 3-20

atmInterfaceTCTable, 3-14

atmSigStatTable, 3-23

atmSoftPvcBaseGroup, 3-18

atmSoftPVccTable, 3-19

atmSoftPVpcTable, 3-19

atmSvcVcCrossConnectTable, 3-23

atmSvcVpCrossConnectTable, 3-23

atmTrafficDescrParamTable, 3-14

atmVcCrossConnectAdminStatus, 2-9

atmVcCrossConnectTable, 3-17

atmVpCrossConnectTable, 3-16

atmVplTable, 3-15

AToM MIB, 3-21

C

circuits

- defined, 2-6
- DSLAM example, 2-10

configuration

- ATM addresses, 2-11
- creating traffic descriptors, 2-4
- overview, 2-1
- physical interfaces, 2-2
- PNNI routing example, 2-18
- PVC cross-connect, 2-6
- SPVC example, 2-10

connection scope selection, 1-6

crankback, 3-28

createAndGo, 2-1

cross-connects

- activating, 2-9
- atmSvcVcCrossConnectTable, 3-23
- atmSvcVpCrossConnectTable, 3-23
- atmVcCrossConnectTab, 3-17
- atmVpCrossConnectTable, 3-16
- configuration, 3-16
- creating and activating, 2-9
- defined, 1-4

Index

D

PVC example, 2-6

D

defaults

physical interfaces, 3-4, 3-7

Stinger-specific MIB defaults, 3-1

traffic descriptors, 2-5

designated transit list (DTL), 3-40

documentation conventions, xiii

DS1-ATM MIB, 3-4

DS3-ATM ifType 30, 1-1

DS3-ATM line status MIB, 3-42

DS3-ATM MIB, 3-6

DSLAM, configuring a circuit, example, 2-10

E

E3-ATM line status MIB, 3-43

H

Hello packet, 1-7

I

IETF defined MIBs, xvi

ifStackTable, 1-3

ifType values, 1-1

ILMI, 3-27

ITU-T recommendations, xv

L

license, verifying, 2-1

load sequence for MIBs, A-1

M

M4 network element view, 3-18

management objects for ATM, 3-13

mapping UNI scope to PNNI, 1-6

membership scope, 1-6

metrics

configuration, 3-32

described, 1-7

MIB

DS3-ATM line status, 3-42

E3-ATM line status, 3-43

OC3-ATM line status, 3-44

SONET extension, 3-48

MIB tables, 3-1

MIBs

ATM, RFC 2515, 3-18

AToM, 3-21

creating new entries, 2-1

defined by the IETF, xvi

deprecated tables (PNNI), 3-39

DS1-ATM, 3-4

DS3-ATM, 3-6

load sequence, A-1

M4 Network Element View, 3-18

mibinet.mib, 2-15

OC3-ATM, 3-9

PNNI, 3-28

proprietary for physical interfaces, 3-3

read-only tables, 3-35

Soft PVC (SPVC), 3-18

Stinger modifications to standard settings, 3-1

Switch-Config, 3-11

unsupported tables (PNNI), 3-39

N

network configuration guidelines, 2-1

node ID for PNNI, 2-18

O

OC3-ATM interface ifType 39, 1-1

OC3-ATM line status MIB, 3-44

OC3-ATM MIB, 3-9

P

peer group leader election, 1-6

permanent virtual circuits. See PVCs

PGL configuration, 3-29

physical interface

proprietary MIBs, 3-3

physical interfaces

configuration example, 2-2

slotIfTable entry, 2-2

stacked interfaces, 1-3

physical interfaces in the Stinger unit, 1-2

PNNI

conditions for accepting set requests, 2-20

configuration example, 2-18
configuration overview, 1-5
deprecated tables, 3-39
Hello, 1-7, 3-37
interface table, 3-32
introduction, 1-5
license enabled, 2-1
link and node information, 3-35
mapping UNI scope, 1-6
metric, 3-32
metrics, 1-7
node configuration, 3-28
node ID, 2-18
peer group ID, 2-18
PGL election, 1-6
pnniBaseGroup, 3-28
pnniIfTable, 3-32
pnniLinkTable, 3-35
pnniMapAddrTable, 3-39
pnniMapTable, 3-38
pnniMapTnsTable, 3-39
pnniMetricsTable, 3-33
pnniNbrPeerPortTable, 3-37
pnniNbrPeerTable, 3-36
pnniNodePglTable, 3-29
pnniNodeSvccTable, 3-30
pnniNodeTable, 3-29
pnniNodeTimerTable, 3-30
pnniPtseTable, 3-37
pnniRouteBaseGroup, 3-33
pnniRouteTnsTable, 3-34
pnniScopeMappingTable, 3-31
pnniSummaryAddressTable, 3-31
pnniSvccRccTable, 3-37
RCC, 1-7
static routes, 1-7, 3-33
timers, 3-30
topology state elements, 1-7
topology state packets, 1-7
unsupported tables, 3-39

PTSEs, 1-7
PTSPs, 1-7
PVCs
 cross-connect configuration, 3-16
 cross-connect configuration example, 2-6
 defined, 1-4
 example of terminating connection, 2-15
 maximum number of, 2-1

R

reachable ATM end system addresses (AESAs), 3-33
reachable transit networks, 3-34
Requests for Comment (RFCs), xv

routing control channel. See RCC

S

safety information, xiv
semantics of UNI scope hierarchy, 1-6
set requests, conditions for acceptance, 2-20
slotIfTable entry, 2-2
Soft PVC (SPVC) MIB, 3-18
Soft PVC (SPVC) MIB tables overview, 2-11
SONET extension MIB, 3-48
SPVCs
 addresses, 3-20
 atmCurrentlyFailingSoftPVccTable, 3-20
 atmCurrentlyFailingSoftPVpcTable, 3-20
 atmInterfaceSoftPvcAddressTable, 3-20
 atmSoftPvcBaseGroup, 3-18
 atmSoftPVccTable, 3-19
 atmSoftPVpcTable, 3-19
 configuration, 3-18
 configuration example, 2-10
 cross-connect configuration, 3-22
 defined, 1-4
 initiating VCL, creating entry, 2-12
 maximum number of, 2-1
 operational status of, 3-20
 system-generated entries in initiating switch, 2-13
 system-generated entries in the target switch, 2-14
stack, ATM components, 1-1
stacked interfaces, 1-3
static route configuration, 3-33
static routes, 1-7
Stinger documentation set, xiv
Structure of management information definitions, xv
summary addresses, 3-31
SVCC
 RCC configuration, 3-30
 used for RCC, 1-7
Switch-Config MIB, 3-11
switched virtual channel connection. See SVCC
symmetric DSL (SDSL) ifType 96, 1-1

T

timer values, PNNI, 3-30
topology state packets, 1-7
traffic contracts. See ATM
traffic descriptor
 defaults, 2-5
 sample entry, 2-5

Index

V

traffic descriptors
 configuring, 2-4
 MIB table overview, 2-5
transit calls, 2-1
transit networks, 3-34

V

VCL

 atmCurrentlyFailingPVclTable, 3-28
 atmVclTable, 3-15
 configuration, 3-15
 creating an entry, 2-8
 cross-connects, 1-4
 defined, 1-3
 initiating SPVC, 2-12
virtual channel link. See VCL
virtual path link. See VPL.
VPI-VCI ranges on ATM interfaces, 2-3

VPL

 atmCurrentlyFailingPVplTable, 3-28
 atmVplTable, 3-15
 configuration, 3-15
 cross-connects, 1-4
 defined, 1-3