

# DIGITAL ATMswitch 900 Management

---

Part Number: EK-DAHSF-MG. B01

March 1998

This document explains how to manage the DIGITAL ATMswitch modules.

**Revision/Update Information:** This is a revised manual.

**Revision/Update Information:** ATMswitch Firmware, Version 2.0

Digital Equipment Corporation makes no representations that the interconnection of its products in the manner described in this document will not infringe existing or future patent rights, nor do the descriptions contained in this document imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Possession, use or copying of the software described in this publication is authorized only pursuant to a valid written license from Digital or an authorized sublicensor.

NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction document, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at the user's own expense.

Copyright © 1998 by Digital Equipment Corporation  
All Rights Reserved

The following are trademarks of Digital Equipment Corporation: DECconnect, DIGITAL, FLOWmaster, GIGAswitch, ThinWire, and the DIGITAL logo.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

All other trademarks and registered trademarks are the property of their respective holders.

# Contents

---

## Preface

## 1 Introduction

ATMswitch 900 Features . . . . .	1-2
SVC Routing . . . . .	1-2
LAN Emulation . . . . .	1-3
Management Interfaces . . . . .	1-3
Console Commands . . . . .	1-3
Menu Interface . . . . .	1-4
SNMP-Compliant Network Management . . . . .	1-4
System Versions . . . . .	1-5

## 2 Managing Startup Parameters

Displaying Information for Configured Interfaces . . . . .	2-2
Configuring Network Interface Parameters . . . . .	2-3
Setting Box Ethernet IP Addresses and Subnet Masks . . . . .	2-3
Configuring Classical IP Parameters . . . . .	2-4
Creating and Saving the CLIP Client's IP Address . . . . .	2-4
Creating and Saving the CLIP Client's IP Address Mask . . . . .	2-5
Creating and Saving the CLIP Server's ATM NSAP Address . . . . .	2-5
Deleting Saved CLIP Parameters . . . . .	2-5
Setting Up PVCs That Use CLIP . . . . .	2-5
Enabling the CLIP Client . . . . .	2-5
Disabling the CLIP Client . . . . .	2-6
Displaying the Status of the CLIP Configuration . . . . .	2-6
Displaying the IP Counters . . . . .	2-6
Displaying the Contents of the ARP Cache . . . . .	2-7
Displaying Cell Counters . . . . .	2-8

Displaying Port Error Counters . . . . .	2-9
Displaying Version Numbers . . . . .	2-9
Setting Up BootP Information . . . . .	2-9
Forcing Image Downloads . . . . .	2-10

### 3 Managing PVCs

Managing PVCs . . . . .	3-2
Displaying PVCs . . . . .	3-3
Displaying Stored PVCs . . . . .	3-3
Displaying Active PVCs . . . . .	3-5
Displaying VCI Type Ranges . . . . .	3-5
Displaying Circuit Mapping Parameters . . . . .	3-5
Displaying VCI Circuit Usage . . . . .	3-5
Displaying VCI States . . . . .	3-5
Creating PVCs . . . . .	3-6
Creating Point-To-Point ABR Circuits . . . . .	3-6
Creating Point-To-Multipoint ABR Circuits . . . . .	3-6
Creating Point-To-Point CBR Circuits . . . . .	3-7
Creating Point-To-Multipoint CBR Circuits . . . . .	3-7
Deleting PVCs . . . . .	3-8
Disabling Virtual Circuits . . . . .	3-8
Enabling Virtual Circuits . . . . .	3-8
Using Scripts . . . . .	3-9
Creating PVCs with Scripts . . . . .	3-10
Deleting PVCs with Scripts . . . . .	3-11

### 4 Managing Switch Parameters

Displaying the Slot Configuration . . . . .	4-2
Disabling a Port . . . . .	4-3
Enabling a Port . . . . .	4-4
Configuring Framing Support . . . . .	4-5
Displaying SDH/SONET Support . . . . .	4-5
Modifying SDH/SONET Support . . . . .	4-5
Rebooting the Switch Software . . . . .	4-6
Configuring the Transmit Clock Timing Source . . . . .	4-7
Displaying the Transmit Timing Source . . . . .	4-7
Changing the Transmit Timing Source . . . . .	4-7

## 5 Managing System Parameters

Displaying the Current Date and Time .....	5-2
Changing Passwords .....	5-3
Displaying Error Logs .....	5-4
Clearing All Error Logs .....	5-5
Managing Parameters Stored In Flash .....	5-6
Transferring Stored Parameters to the Host .....	5-6
Retrieving Stored Parameters from the Host .....	5-6
Erasing Flash Storage.....	5-7

## 6 Managing Signaling Parameters

Configuring UNI Versions.....	6-2
Configuring the UNI Version for Each Port.....	6-2
Displaying the UNI Status .....	6-2
Storing the UNI Configuration .....	6-3
Clearing the UNI Configuration .....	6-3
Displaying the Stored Configuration .....	6-3
Restoring the UNI Configuration .....	6-3
Configuring Q.SAAL PDU Types.....	6-4
Displaying Q.SAAL Link Statistics .....	6-4
Displaying Q.SAAL Link Status .....	6-5
Displaying Q.2931 Link Statistics .....	6-6
Displaying SVC Connection Information .....	6-7
Displaying Connection Summary.....	6-7
Displaying Link Connections .....	6-7

## 7 Configuring Dynamic Routing

Overview of Dynamic Routing .....	7-2
Automatic Topology Discovery .....	7-2
Shortest Path Routing .....	7-2
Parallel Links Between Switches and VC Load Sharing .....	7-3
Static Routing .....	7-3
Exporting Static Routes .....	7-5
IISP Support .....	7-5
IISP Gateway Functionality .....	7-5
Addressing Guidelines for DIGITAL ATM Switch Networks .....	7-6
Address Autoconfiguration.....	7-7
Configurable Default Prefix .....	7-7
Configurable Address Prefixes for Each Port.....	7-8

Managing Routing Parameters . . . . .	7-9
Displaying Routing Statistics . . . . .	7-9
Managing Static Routes . . . . .	7-9
Configuring a Static Route . . . . .	7-9
Altering an Existing Static Route . . . . .	7-10
Deleting a Static Route . . . . .	7-10
Displaying Static Routes . . . . .	7-10
Managing the ATM Address Prefix . . . . .	7-10
Configuring the Default Address Prefix . . . . .	7-11
Deleting the Default Address Prefix . . . . .	7-11
Displaying the Default Address Prefix . . . . .	7-11
Altering the ATM Address Prefix . . . . .	7-11

## 8 LAN Emulation

LANE Configuration Overview . . . . .	8-2
LAN Emulation Configuration Server . . . . .	8-3
Initializing the LECS . . . . .	8-3
Connecting to the LECS . . . . .	8-3
Configuring the LECS . . . . .	8-4
Managing the LECS . . . . .	8-4
Displaying the LECS Configuration . . . . .	8-4
Enabling the LECS . . . . .	8-5
Disabling the LECS . . . . .	8-5
Deleting the LECS Stored in Memory . . . . .	8-5
Displaying the LECS Counters . . . . .	8-5
Displaying Clients Connected to the LECS . . . . .	8-6
Configuring the ELAN . . . . .	8-6
Displaying the ELAN Summary . . . . .	8-7
Displaying the ELAN Configuration . . . . .	8-7
Creating an ELAN . . . . .	8-8
Enabling the ELAN . . . . .	8-8
Disabling the ELAN . . . . .	8-8
Deleting the ELAN Stored in Memory . . . . .	8-8
Creating an ELAN With a LES and BUS . . . . .	8-9
Enabling the ELAN With a LES and BUS . . . . .	8-9
Disabling the ELAN With a LES and BUS . . . . .	8-9
Deleting the ELAN With a LES and BUS Stored in Memory . . . . .	8-9
Displaying the ELAN State . . . . .	8-9
LAN Emulation Server . . . . .	8-11
Initializing the LES . . . . .	8-11
Joining the Emulated LAN . . . . .	8-11
Registering MAC Addresses . . . . .	8-12
Address Resolution . . . . .	8-13

Topology Changes . . . . .	8-13
Flushing Data VCCs . . . . .	8-14
Terminating the Connection . . . . .	8-14
Managing the LES . . . . .	8-14
Creating a LES . . . . .	8-15
Enabling the LES . . . . .	8-15
Disabling the LES . . . . .	8-15
Deleting the LES Stored in Memory . . . . .	8-15
Displaying the LES Summary . . . . .	8-15
Displaying the LES Configuration . . . . .	8-15
Displaying the Control VCC Counters . . . . .	8-16
Displaying the LES Join Counters . . . . .	8-17
Displaying the LES Register Counters . . . . .	8-18
Displaying the LES ARP Counters . . . . .	8-19
Displaying the LES Topology Change Counters . . . . .	8-20
Displaying the LES Flush Counters . . . . .	8-20
Displaying the Clients Connected to the LES . . . . .	8-21
Displaying Client Address Mappings for the LES . . . . .	8-21
Broadcast and Unknown Destination Server . . . . .	8-22
Initializing the BUS . . . . .	8-22
Forwarding Data Frames . . . . .	8-22
Forwarding Control Frames . . . . .	8-22
Managing the BUS . . . . .	8-23
Creating a BUS . . . . .	8-23
Enabling the BUS . . . . .	8-23
Disabling the BUS . . . . .	8-23
Deleting the BUS Stored in Memory . . . . .	8-23
Displaying the BUS Summary . . . . .	8-23
Displaying the BUS Configuration . . . . .	8-24
Displaying BUS Counters . . . . .	8-24
Displaying the Clients Connected to the BUS . . . . .	8-25
Redundant LANE . . . . .	8-27
Redundant LECS ATM Address . . . . .	8-27
Master Selection . . . . .	8-27
Restoration Option . . . . .	8-28
Hello Timer Interval . . . . .	8-28
Timeout Timer Interval . . . . .	8-28
Dead Factor . . . . .	8-28
Connection Attempt . . . . .	8-28
Managing the Redundant LECS . . . . .	8-28
Displaying the RLECS Parameters . . . . .	8-28
Configuring the Remote RLECS Address . . . . .	8-29
Enabling the Restoration Option . . . . .	8-29
Disabling the Restoration Option . . . . .	8-29
Changing the Hello Timer Interval . . . . .	8-29

Changing the Timeout Timer Interval . . . . .	8-29
Changing the Dead Factor Parameter . . . . .	8-29
Changing the Connection Attempt Parameter . . . . .	8-29
Storing the RLECS Parameters . . . . .	8-30
Displaying the Stored RLECS Parameters. . . . .	8-30
Changing the Master Selection Criteria. . . . .	8-30

## 9 Managing the System Using SNMP

SNMP Management Concepts . . . . .	9-2
Initial NMS Commands . . . . .	9-4
Displaying the SNMP Agent Configuration . . . . .	9-4
Configuring the SNMP Agent . . . . .	9-4
Modifying the Read-Write Community Name . . . . .	9-5
Modifying the Read-Only Community Name . . . . .	9-5
Modifying the Trap Community Name . . . . .	9-5
Adding the NMS IP Trap Address. . . . .	9-5
Deleting the NMS IP Trap Address . . . . .	9-5
Displaying Link Information . . . . .	9-5
Displaying the Status of All Ports . . . . .	9-5
Displaying Registered Addresses. . . . .	9-6
ATM SNMP Implementation . . . . .	9-7
Address Registration Unit . . . . .	9-7
SNMP Agent . . . . .	9-8
Implemented MIB Object Groups. . . . .	9-9
ATM SNMP Agent . . . . .	9-11

## A ATM Command Summary

bus . . . . .	A-3
clearflash . . . . .	A-5
clip. . . . .	A-6
date . . . . .	A-7
decnni . . . . .	A-8
diagnose. . . . .	A-10
elan . . . . .	A-12
elanall . . . . .	A-15
errorlog . . . . .	A-17
h . . . . .	A-18
help . . . . .	A-19
host . . . . .	A-20
iam. . . . .	A-21
ifconfig . . . . .	A-22



lecs	A-24
les	A-25
link	A-27
loadconfig	A-28
netstat	A-29
params	A-32
passwd	A-33
password_disable	A-34
password_enable	A-35
plcp	A-36
port	A-37
pvc	A-38
restart_factory_defaults	A-41
rlecs	A-42
script	A-43
service	A-44
set_gateway_IP_address("gateway-ip-addr")	A-46
shellLock(1)	A-47
shellLock(0)	A-48
show_port_status	A-49
sig	A-50
snmp	A-53
status	A-55
switch	A-56
task	A-57
trace	A-58
versions	A-61
whoami	A-62

## Figures

7-1	DIGITAL ATM Network Configuration Example . . . . .	7-3
7-2	Static Route Configuration Example . . . . .	7-5
7-3	DIGITAL ATM IISP Gateway . . . . .	7-6
7-4	ATM NSAP Address Format . . . . .	7-7

## Tables

3-1	PVC Configuration Worksheet . . . . .	3-3
9-1	Additional Information About SNMP . . . . .	9-2
9-2	MIB Object Groups . . . . .	9-9
9-3	Interface Numbering . . . . .	9-12

---

# Preface

This document provides instructions for managing the DIGITAL ATMswitch modules using the ATM command line interface and other network management tools. For information about configuring and testing your particular system (including the use of module LEDs), refer to your Installation and Configuration manual.

## Intended Audience

This document is intended for personnel who are experienced in managing network hardware.

## Document Structure

This document is structured as follows:

- Chapter 1 describes the management information covered by this manual.
- Chapter 2 describes the commands used to manage the startup parameters.
- Chapter 3 describes the commands used to configure the permanent virtual circuits.
- Chapter 4 describes the commands used to manage the switch configuration parameters.
- Chapter 5 describes the commands used to configure the system utility parameters.
- Chapter 6 describes the commands used to configure the signaling parameters.
- Chapter 7 describes how to configure dynamic routing for switched virtual circuits (SVCs).
- Chapter 8 describes the LAN emulation support.
- Chapter 9 introduces SNMP management concepts and describes how to use SNMP to manage your ATM system. It also describes some of the commands used to manage the SNMP configuration parameters.
- Appendix A is the ATM command summary.

## Additional Resources

The following document provides additional information:

- *ATMswitch 900 Installation and Configuration*, EK-DAHSF-IN
- *GIGAswitch/ATM System Installation and Service*, AA-QCV7\*-TE
- *GIGAswitch/ATM 5-Slot System Installation and Service*, EK-DAGWG-IN
- *DIGITAL ATM Modular PHY Cards Installation*, EK-DAGGM-IN
- *Configuring the SNMP Agent*, AA-PR84A-TE

The following documents provide information about the implemented standards:

- UNI V3.0, ATM User-Network Interface Specification
- UNI V3.1, ATM User-Network Interface Specification
- LAN Emulation Over ATM Version 1.0
- RFC 1157: SNMP Standard
- RFC 1213: MIB-II
- RFC 1406: DS1/E1 MIB
- RFC 1407: DS3/E3 MIB
- RFC 1573: Evolution of the Interfaces Group of MIB-II
- RFC 1595: SONET MIB
- RFC 1695: ATM MIB
- DEC ATM MIB
- DEC ATM Signaling MIB
- DEC ATM Switch Chassis MIB
- DEC LAN Emulation Service MIBs
- Common DEChub 900 Objects MIB

Documentation for your network management station (NMS) should also be available for regular use. RFCs can be obtained from `gatekeeper.dec.com` (the DIGITAL FTP site) in the `/pub/net/info/rfc` directory using anonymous ftp. You can obtain a list of primary RFC repositories from `nic.ddn.mil` using anonymous ftp.

## Online Services

Product-specific information is available from the DIGITAL Network Product Business Web Site.

The Network Product Business Web Site can be reached from different geographic locations using the following URLs:

---

Americas	<a href="http://www.networks.digital.com">http://www.networks.digital.com</a>
Europe	<a href="http://www.networks.europe.digital.com">http://www.networks.europe.digital.com</a>
Australia	<a href="http://www.networks.digital.com.au">http://www.networks.digital.com.au</a>

---

## Conventions

The following conventions are used in this document:

---

<b>Convention</b>	<b>Meaning</b>
<i>italic type</i>	Indicates a variable in examples. In text, indicates that a word or phrase is being emphasized to the reader.
<b>bold type</b>	Indicates a literal example of user input.
monospaced type	Indicates system output. In text, indicates command names, keywords, node names, filenames, directories, and utilities.
<Return>	Indicates that you press the Return key on the keyboard.
<Ctrl/x>	Indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.

---

## How to Order Additional Documentation

To order additional documentation, use the following information:

<b>To Order</b>	<b>Contact</b>
By Telephone	USA (except Alaska, New Hampshire, and Hawaii): 1-800-DIGITAL (1-800-344-4825) Alaska, New Hampshire, and Hawaii: 1-603-884-6660 Canada: 1-800-267-6215
Electronically (USA only)	Dial 1-800-DEC-DEMO (For assistance, call 1-800-DIGITAL)
By Mail (USA and Puerto Rico)	DIGITAL EQUIPMENT CORPORATION P.O. Box CS2008 Nashua, New Hampshire 03061 (Place prepaid orders from Puerto Rico with the local DIGITAL subsidiary: 809-754-7575)
By Mail (Canada)	DIGITAL EQUIPMENT of CANADA LTD. 940 Belfast Road Ottawa, Ontario, Canada K1G 4C2 Attn.: A&SG Business Manager
Internationally	DIGITAL EQUIPMENT CORPORATION Attn.: A&SG Business Manager c/o local DIGITAL subsidiary or approved distributor
Internally	U.S. Software Supply Business (SSB) DIGITAL EQUIPMENT CORPORATION 8 Cotton Road Nashua, New Hampshire 03063

---

# Chapter 1

---

## Introduction

The ATMswitch 900 is a standalone or networked, intelligent, switching system that forwards cells among a set of connected ATM links based on the VCI field within the cell header. The links can be connected to end stations, bridges, routers, another vendor's switches, and to other Digital ATM switches. The core of the ATMswitch 900 is a shared memory switching system that forms point-to-point and point-to-multipoint communication paths by mapping inputs to outputs.

The ATMswitch 900 allows seamless operation of legacy protocols using connections to ATM stations that operate a standard LAN Emulation Client (LEC) protocol. Some ATM stations can act as a proxy for stations connected to a legacy LAN. For example, the VNswitch 900 connects Ethernet LANs to the ATM network. Together, the ATMswitch 900 and the VNswitch 900 provide a high-speed, highly configurable ATM backbone for the network. A single ATMswitch 900 can connect hundreds of Ethernet LANs (each containing many Ethernet stations), several ATM file servers, and two WAN connections into an ATM network. Multiple ATM switches can be connected in a mesh topology that provides increased data capacity and availability through redundant connections. The overall network can be configured into one LAN or many independent LANs using LAN emulation by local and remote network management facilities.

## ATMswitch 900 Features

The ATMswitch 900 provides point-to-point and point-to-multipoint circuits in both single and multiple switch environments. It supports Constant Bit Rate (CBR), Variable Bit Rate (VBR), and Available Bit Rate (ABR) traffic classes, including ABR with credit-based flow control, in the following manner:

- CBR point-to-point circuits
- CBR point-to-multipoint circuits
- VBR point-to-point circuits — DIGITAL's ATM switches accept SVC setup requests for VBR circuits and displays them as CBR circuits with Peak Cell Rate (PCR) equal to Sustained Cell Rate (SCR).
- ABR point-to-point circuits — where credit-based flow control is enabled automatically on compatible devices.
- ABR point-to-point circuits — where rate-based flow control, using Explicit Forward Congestion Indication (EFCI), is enabled automatically.
- ABR point-to-multipoint circuits

The ATMswitch 900 also allows you to set up these circuits in the following manner:

- PVC — using the local console or a remote system with a Telnet connection.
- PVC — using SNMP and the ATM MIB (RFC 1695).
- SVC — using the ATM Forum UNI specification.

The ATMswitch 900 offers management interfaces to support SVC routing, LAN emulation, and other management features.

## SVC Routing

The ATMswitch 900 provides multiswitch routing features that support ATM Forum Interim Interswitch Signaling Protocol (IISP) functions for third-party interoperability. These routing features include:

- Automatic configuration
- Shortest path first (SPF) routing
- Congestion avoidance
- Congestion detection
- Smart multipoint routing
- Rapid recovery



## ATMswitch 900 Features

The network manager does not need to configure the network because the following operations are performed automatically if ILMI is supported.

- The default ATM NSAP addressing information for all switches is configured.
- The ATM NSAP prefix is distributed throughout the network from a single switch.
- The IISP static routes are distributed.
- ATM addresses are registered using ILMI.
- The network topology is determined.

The network automatically determines if the other object is a DIGITAL host, a DIGITAL switch, or another vendor's equipment. If the environment contains only DIGITAL equipment, autoconfiguration takes place. However, if you are dealing with a mixed environment, you might have to perform some configuration. For example, if you must add other vendor's equipment that is not standards-compliant or you have hosts that do not support ILMI, you must perform some configuration procedures for static routing using the menu interface.

## LAN Emulation

LAN emulation allows existing network applications to operate on a native ATM end station without modification. Supported LAN emulation features include:

- Console commands to manage LAN emulation servers.
- A LAN Emulation Configuration Server (LECS) supporting multiple emulated LANs.
- Multiple LAN Emulation Servers (LESSs) allowing for the support of multiple emulated LANs.
- Multiple Broadcast and Unknown Servers (BUSs) to support multiple emulated LANs.

## Management Interfaces

You can use the following management interfaces:

- Command line interface (console commands)
- Menu interface (sometimes referred to as the out-of-band management facility)
- SNMP management (network management station)

## Console Commands

Console commands can be issued from the local console or from a remote Telnet session. These commands can be used to:

## ATMswitch 900 Features

- Configure permanent virtual circuits (PVCs).
- Configure and manage LAN emulation servers.
- Configure and manage UNI versions.
- Display module information such as firmware versions, creation time, error logs, and network information.
- Configure and manage module parameters such as passwords; setup information for Telnet access, SNMP management, and image downloads; and parameters stored in nonvolatile memory.
- Invoke other utilities such as the menu interface or the help utility.

## Menu Interface

The menu interface, sometimes referred to as the out-of-band management (OBM) facility, provides several management functions for the ATMswitch 900. You can use the menu interface by connecting a terminal to the module's setup port or creating an OBM session from a remote system on the network using Telnet, TCP/IP, and Ethernet. Diagnostic services for the ATMswitch 900 are also available from the local console or the network connection.

The menu interface provides the ability to:

- Assign an IP address to the ATMswitch 900 module.
- Show and change the state of module parameters and characteristics.
- Set initial parameters for module operation.
- Set and show important operational parameters when SNMP management is unavailable.

## SNMP-Compliant Network Management

The ATMswitch 900 can be managed from anywhere on the Ethernet network using a network management station (NMS) that implements Simple Network Management Protocol (SNMP). The ATMswitch 900 has an SNMP agent that allows an SNMP-compliant NMS to read and set certain management parameters contained in Management Information Bases (MIBs).

The ATMswitch 900 supports a wide range of standard MIBs as well as vendor-specific and device-specific MIBs. All supported MIBs must be loaded onto the NMS before it can be used to manage an ATMswitch 900. Any SNMP management station on the network can manage the ATMswitch 900 once it knows the ATMswitch 900 module's IP address and community name.

---

## System Versions

The management features described in this manual are supported for the ATMswitch 900 module and its supported ATM modular PHY cards. Refer to the firmware release notes for the latest information about the firmware and its support for ATM modular PHY cards.



## Chapter 2

---

# Managing Startup Parameters

To manage the DIGITAL ATM system when you first start up the system, you might need to configure or display the following information:

- IP addresses
- Classical IP parameters
- Port status
- Port counters
- Firmware images

Displaying Information for Configured Interfaces

---

## Displaying Information for Configured Interfaces

To display all configured interfaces (non-ATM), use the `ifconfig -show` command.

## Configuring Network Interface Parameters

To configure network interface parameters such as Ethernet IP addresses and subnet masks, use the `ifconfig` command.

### Setting Box Ethernet IP Addresses and Subnet Masks

This address is used for Telnet and SNMP access to the switch. Both the IP address and the subnet mask must be specified in standard dotted decimal notation. You must reboot the switch to have changes to the IP address or the netmask take effect.

To set the IP address and subnet mask for the switch Ethernet interface, use the following command:

```
ifconfig -iface snx -address addr -nm mask
```

where *x* is the ID (for example, 0), *addr* is the IP address, and *mask* is the netmask for the interface.

## Configuring Classical IP Parameters

A Classical IP (CLIP) client supports ATM access to the IP stack. This CLIP client allows SNMP management of the switch over ATM, as well as out-of-band management activities using shell interfaces that previously were only available via Ethernet.

Connectivity is supported in a client-client configuration with permanent virtual circuit (PVC) connections or in a client-server configuration over switched virtual circuits (SVCs).

The CLIP agent is created when the switch is powered on or when the `clip -enable` command is executed. This command configures the CLIP client using the available CLIP parameters and writes a powerup CLIP flag to nonvolatile storage (Flash). Subsequent powerups for the ATM system result in CLIP being configured at powerup.

A unique client IP address is required for the CLIP agent to operate over either PVCs or SVCs. At powerup, if the CLIP client IP address has been set up previously, it will be initialized from Flash. The CLIP server ATM NSAP address must be initialized for the switch CLIP client to signal SVC connections over which CLIP traffic is passed. At powerup, if a CLIP server address has been set up previously, it will be initialized from Flash.

Depending on the parameters that have been set when you try to enable CLIP, the following situations are possible:

- If the CLIP client is enabled without the client IP address and the server ATM address being configured, the CLIP client is not enabled and an error message is displayed to the user.
- If the CLIP client is enabled with only a client IP address set up, then CLIP will support only client-client traffic. If you require client-server traffic, disable the CLIP agent, set up the server ATM address, and enable the CLIP agent.
- If the CLIP client is enabled with both the client IP address and the server ATM address set up, then the CLIP agent can support either SVC or PVC connections over CLIP.

You can use the `clip` command and its options to configure the CLIP client.

### Creating and Saving the CLIP Client's IP Address

Use the `clip -ipaddr address` command to create and save the CLIP client's IP address in Flash memory. If the client IP address has been configured already, it is initialized at powerup from Flash.



## Creating and Saving the CLIP Client's IP Address Mask

Use the `clip -ipmask mask` command to create and save the CLIP client's IP address mask in Flash memory. If the client IP address has been configured already, it is initialized at powerup from Flash.

## Creating and Saving the CLIP Server's ATM NSAP Address

Use the `clip -server address` command to create and save the CLIP server's ATM NSAP address in Flash memory. If the server ATM NSAP address has been configured already, it is initialized at powerup from Flash.

## Deleting Saved CLIP Parameters

Use the `clip -clrflash` command to delete the CLIP parameters saved in Flash memory. All CLIP PVCs, the CLIP client IP address, and the CLIP server ATM NSAP address are deleted from Flash. The powerup CLIP flag is also deleted from Flash (see Section ). The following message is displayed:

```
Nonvolatile CLIP Configuration Deleted
```

## Setting Up PVCs That Use CLIP

CLIP is not supported over PVCs set up with the `pvc` command. PVCs using CLIP must be set up with the `clip -pvc` command so that they are registered with the CLIP agent. PVCs set up using this command are saved to Flash automatically.

## Enabling the CLIP Client

Use the `clip -enable` command to enable the CLIP client. In addition to configuring the CLIP client, this command writes a powerup CLIP flag to Flash. If this flag is present when the switch is powered up, the CLIP client is turned on automatically. This situation continues until the powerup CLIP flag is cleared using the `clip -clrflash` command.

After you enter this command, you should see a display that resembles the following example:

```
No CLIP Client IP Mask configured,  
Default of 255.255.255.0 will be used.
```

```
CLIP Client enabled
```

## Configuring Classical IP Parameters

### Disabling the CLIP Client

Use the `clip -disable` command to disable the CLIP client. After you enter this command, you should see a display that resembles the following example:

```
Deregister CLIP Agent
```

```
CLIP Client disabled; use clip -enable to restart
```

### Displaying the Status of the CLIP Configuration

Use the `clip -show` to display the status of the CLIP configuration. You should see a display that resembles the following example:

```
CLIP Agent Configuration:
```

```
CLIP Client IP Addr:      100.1.1.19
CLIP Client IP Mask:     255.255.255.0
CLIP Server (NSAP) Addr: 3999990000000000000f8465b80.88002b223430.00
CLIP Client (NSAP) Addr: 3999990000000000000f8465b80.0000f8465b95.00
```

### Displaying the IP Counters

Use the `clip -counters` to display the ARP and IP counters for the CLIP interface. You should see a display that resembles the following example:

```
CLIP Counters:
xmt_pkts_dropped_no_conn:      0
xmt_mcast_pkts_dropped:       0
xmt_ok:                         0
xmt_congested:                 0
xmt_failed:                    0
arp_naks_sent:                 0
arp_replies_sent:              0
inverse_arp_replies_sent:      1
buffer_access_failed:          0
inverse_atm_arp_reqs:          0
inverse_atm_arp_retries:       0
inverse_atm_arp_refreshes:     0
atm_arp_reqs:                  0
atm_arp_retries:               0
```

## Configuring Classical IP Parameters

```
atm_arp_refreshes:          0
svc_connections_aged:       0
unsolicited_connections_aged: 0
pvc_aged_refreshes:        0
conn_rejects_sap_mismatch:  0
conn_rejects_resources:     0
conn_accepts:               0
network_releases:           0
ip_pkts_rcvd:               0
invalid_encapsulations_rcvd: 0
null_proto_addr_rcvd:       0
arp_naks_rcvd:              0
inverse_arp_replies_rcvd:   0
inverse_arp_reqs_rcvd:      1
arp_reqs_rcvd:              0
arp_replies_rcvd:          0
unrecognized_arps_rcvd:     0
```

## Displaying the Contents of the ARP Cache

Use the `clip -atmarp` to display the current contents of the ARP cache. You should see a display that resembles the following example:

```
ARP Cache Configuration:
```

```
ARP Entry IP Addr:          100.1.1.210
CLIP ARP Table (NSAP) Addr:  39999900000000000000f8465b80.88002b223430.00
```

Displaying Cell Counters

---

## Displaying Cell Counters

To display cell counters for a particular port, use the `port -cell -port 1:port` command, where *port* specifies the port.

---

## Displaying Port Error Counters

Use the `port -suni -port 1:port` command to display port error counters, where *port* specifies the port. You should see listings for the running counters for each port interface. These counters include byte interleave parity (BIP) errors, far-end block errors, and header check sequence (HCS) errors. Definitions for all of these counters conform with the SONET standards.

```
Rcv Section Overhead BIP Errors:          436
Rcv Line Overhead BIP Errors:            370
Rcv Line Overhead Far-End Block Errors:   70
Rcv Path Overhead BIP Errors:            148
Rcv Path Overhead Far-End Block Errors:  2125
Rcv ATM Cell Correctable HCS Errors:      0
Rcv ATM Cell Uncorrectable HCS Errors:    0
```

## Displaying Version Numbers

Use the `versions` command to display the version numbers for the Boot ROM and the firmware image that are running on the switch as shown in the following example:

```
ATMswitch-> versions
RomVersion: Release 1.3[503] : Wed Feb 2 12:01:52 EST 1998
AppVersion: Release 1.3[503] : Wed Frb 2 12:06:15 EST 1998
Host IP Address for flash downloads 16.17.18.19.
Host Boot Filename N900V1-3.bin.
ATMswitch->
```

## Setting Up BootP Information

If you do not wish to use the BootP server to download images, you can set up the information normally obtained from the BootP process using the following command:

```
loadconfig -conf -swip address -netmask mask -hostip address -file
filename
```

where *address* for the `-swip` option specifies the switch IP address, *mask* specifies the switch netmask, *address* for the `-hostip` option specifies the host IP address, and *filename* is the name of the default configuration file that is downloaded when using the `loadconfig -download` command.

Displaying Port Error Counters

## Forcing Image Downloads

After downloading kernel and application images from the network and storing them in Flash, the `loadconfig -download` command can be used to perform a one-time download of a different firmware version.

To download a different version of firmware (for example, when upgrading to a new version or downgrading to an older version), use the `loadconfig -download` command and then reboot the switch. While rebooting, the switch detects that new images have to be loaded and uses the setup information from Flash (set by the `loadconfig` command as described in Setting Up BootP Information) or from the BootP process (if no information is available in Flash) to perform the download.

## **Chapter 3**

---

# **Managing PVCs**

This chapter describes how to manage permanent virtual circuits (PVCs).

## Managing PVCs

The `pvc` command lets you manipulate the permanent virtual circuits (PVCs) on the ATM system. You can display, set up, and delete PVCs using different options. You might want to use the switched virtual circuits (SVCs) described in Chapter 7 instead of managing PVCs. If you choose to set up PVCs, the following worksheet can help you gather information about your hosts for setting up PVCs. You might also want to include information about bandwidth and flow control. You do not need this information for setting up SVCs.





## Managing PVCs

The PVC records provide the following information:

inslot, inport, invci	Indicates the slot, port, and VCI for the inbound link.
outslot, outport, outvci	Indicates the slot, port, and VCI for the outbound link.
flowmaster	Indicates whether FLOWmaster flow control is on or off. ON indicates that flow control is on. Otherwise, nothing is displayed in this column.
bandwidth	Indicates the rate in Mb/s.
bvci	Indicates the branch VCI.

ATMswitch-> **pvc -flash**

Point to Point VCs

inslot	inport	invci	outslot	outport	outvci
1	1	460	1	2	360

Point to MultiPoint VBR VCs

inslot	inport	invci	flowmaster
1	4	410	
1	1	34	ON

Point to MultiPoint CBR VCs

inslot	inport	invci	bandwidth	bvci
1	1	102	14	202
1	1	340	14	370

Point to MultiPoint CBR Branch VCs

inslot	inport	invci	outslot	outport	outvci
1	1	102	1	2	202
1	1	340	1	3	370

## Displaying Active PVCs

To display the active PVCs, use the `pvc -all` command. The syntax for this command is:

```
pvc -all [ -internal ] [ -slot 1 | -port 1:port ]
```

where *port* is the port number. If you specify the `-internal` option, internal PVCs are included in the display.

## Displaying VCI Type Ranges

To display the VCI type ranges for a specific port, use this command:

```
pvc -range -port 1:port
```

where *port* is the port number.

## Displaying Circuit Mapping Parameters

To display the internal circuit mapping parameters for a specific port, use this command:

```
pvc -map -port 1:port
```

where *port* is the port number. The output would resemble the following display:

For linecard: 1 and port: 1 the base and size are:

```
VCI Base:      0
```

```
VCI Size:      8192
```

## Displaying VCI Circuit Usage

To display the VCI circuit usage for a specific VC, use this command:

```
pvc -usage -link 1:port:vci
```

where *port* is the port number and *vci* is the VCI. The output would resemble the following display:

```
VCI 400 on linecard 1 port 1 is in use in both directions
```

## Displaying VCI States

To display the VCI in and out states for a specific VC, use this command:

```
pvc -state -link 1:port:vci
```

where *port* is the port number and *vci* is the VCI.

## Managing PVCs

### Creating PVCs

Using the `pvc` command, you can create Available Bit Rate (ABR), Constant Bit Rate (CBR), and Unspecified Bit Rate (UBR) circuits that are point-to-point or point-to-multipoint.

### Creating Point-To-Point ABR Circuits

Use the following command to set up and save parameters for ABR point-to-point circuits:

```
pvc -abr -link_1 1:port:vci -link_2 1:port:vci [-nodup] [-fcin] [-fcout]
```

You can specify the port and VCI for the links. By default, the circuit is duplex, but you can use the `-nodup` option to change this parameter. You can specify that FLOWmaster flow control be used for your input and output ports by using the `-fcin` and `-fcout` options. Note that the link must support FLOWmaster flow control for flow control to be enabled. Make sure that the virtual circuit and the link to the end station are created with flow control support.

For example, the following command creates a half-duplex, point-to-point ABR circuit that supports flow control from port 3, VCI 300 to port 1, VCI 400.

```
ATMswitch-> pvc -abr -link_1 1:3:300 -link_2 1:1:400 -nodup -fcin
```

```
Circuit established between 3 Invc1:300 and 1 Outvc1: 400
```

### Creating Point-To-Multipoint ABR Circuits

Use the following command to set up and save parameters for an ABR point-to-multipoint root:

```
pvc -inputabr -link_1 1:port:vci [-fcin]
```

You can specify the port and VCI for the link. You can specify that FLOWmaster flow control be used by using the `-fcin` option. Note that the link must support FLOWmaster flow control for flow control to be enabled. Make sure that the virtual circuit and the link to the end station are created with flow control support.

Use the following command to set up and save parameters for an ABR point-to-multipoint branch:

```
pvc -outputabr -link_1 1:port:vci -link_2 1:port:vci [-fcout]
```

You can specify the port and VCI for the links. You can specify that FLOWmaster flow control be used by using the `-fcout` option. Note that the link must support FLOWmaster flow control for flow control to be enabled. Make sure that the virtual circuit and the link to the end station are created with flow control support.

## Creating Point-To-Point CBR Circuits

Use the following command to set up and save parameters for CBR point-to-point circuits:

```
pvc -cbr -link_1 1:port:vci -link_2 1:port:vci -bw mbs -rbw mbs [-nodup]
```

You can specify the port and VCI for the links. You must specify the bandwidth using the `-bw` (for the forward bandwidth) and `-rbw` (for the reverse bandwidth) options. The bandwidth values cannot be 0. The total CBR bandwidth for an OC-3 link is 126 Mb/s. The maximum bandwidth for an OC-12 link is 400 Mb/s. By default, the circuit is duplex, but you can use the `-nodup` option to change this parameter.

For example, the following command creates a full-duplex, point-to-point CBR circuit with 20 Mb/s forward and reverse bandwidth that is stored in Flash. Because the VCIs are specified as 0, the circuit setup makes internal assignments.

```
ATMswitch-> pvc -cbr -link_1 1:2:0 -link_2 1:1:0 -bw 20 -rbw 20
```

## Creating Point-To-Multipoint CBR Circuits

Use the following command to set up and save parameters for a CBR point-to-multipoint root:

```
pvc -inputcbr -link_1 1:port:vci -bw mbs -bvci vci
```

You can specify the port and VCI for the link used as the root. You must specify the bandwidth using the `-bw` option. The bandwidth value cannot be 0. The total CBR bandwidth for an OC-3 link is 126 Mb/s. The maximum bandwidth for an OC-12 link is 400 Mb/s. Use the `-bvci` option to specify the branch VCI. If you specify 0 as the branch VCI, an available VCI is selected for you.

Use the following command to set up and save parameters for a CBR point-to-multipoint branch:

```
pvc -outputcbr -link_1 1:port:vci -link_2 1:port:vci
```

You can specify the port and VCI for the links.

For example, the following command creates a point-to-multipoint CBR circuit on port 1 with input VCI 200 that is stored in Flash. CBR circuits must specify the `-bvci` argument with a valid value. VCI 201 is reserved at all ports even though no branches are created.

```
ATMswitch-> pvc -inputcbr -link_1 1:1:200 -bw 10 -bvci 201
```

```
CBR P-MP Cct rsvd from Port 1 and Invci: 200 for Outvci: 201
```

```
The reserved bandwidth for the circuit is 10 Megabits per sec
```

## Managing PVCs

```
The reserved bandwidth for the circuit is 26292 cells per second
The reserved bandwidth for the circuit is 14 slots per frame
Point-to-Point PVC Record committed to flash successfully.
```

The following command adds a branch to the circuit in the preceding example. Note that whenever a branch is added, the root input VCI must be specified completely to correctly associate the branch.

```
ATMswitch-> pvc -outputcbr -link_1 1:1:200 -link_2 1:2:201
```

```
CBR P-MP Branch added on outvci: 201 from invci: 200
```

```
Point-to-Multipoint PVC Branch Record committed to flash
successfully
```

## Deleting PVCs

Using the `pvc` command, you can delete point-to-point, root, or branch addresses from the VC address mapping and remove them from Flash.

To delete point-to-point or root addresses from the VC address mapping and remove them from Flash, use the following command:

```
pvc -rm -pp | -root -link_1 1:port:vci
```

where *port* is the port number and *vci* is the virtual circuit number for the point-to-point circuit or point-to-multipoint root. If you specify `-pp`, then a point-to-point circuit teardown is attempted. If you specify `-root`, then a point-to-multipoint circuit teardown is attempted. In the case of a point-to-multipoint circuit teardown, the entire circuit (including branches) is deleted.

To delete branch addresses from the VC address mapping and remove them from Flash, use the following command:

```
pvc -rm -branch [ -link_1 1:port:vci ] -link_2 1:port:vci
```

where *port* is the port number and *vci* is the virtual circuit number for the branch.

## Disabling Virtual Circuits

To disable a specific virtual circuit, use the following command:

```
pvc -disable -link 1:port:vci
```

where *port* is the port number and *vci* is the VCI.

## Enabling Virtual Circuits

To enable a specific virtual circuit, use the following command:

```
pvc -enable -link 1:port:vci
```

where *port* is the port number and *vci* is the VCI.

## Using Scripts

Scripts are used to facilitate VC definition during startup. The syntax for the `script` command is as follows:

```
script ("username", "user-password", "host-address", "host-dir", "file-name")
```

Each argument is enclosed in quotes. The host IP address is optional. If 0 is specified, then the file is loaded from the bootp server.

You might use a command resembling the following command to execute your script file.

```
ATMswitch-> script ("smith", "smithpswd", "16.17.18.19", "/users/  
smith", "pvcset")
```

## Managing PVCs

### Creating PVCs with Scripts

This example `script` command creates point-to-multipoint ABR circuits and commits them to Flash memory. The script file called `pvc1.scr` is shown first with the command and the log of the script running on the console following it.

```
pvc -inputabr -link_1 1:1:100
pvc -outputabr -link_1 1:1:100 -link_2 1:2:101
pvc -outputabr -link_1 1:1:100 -link_2 1:3:102
pvc -outputabr -link_1 1:1:100 -link_2 1:4:103
```

```
ATMswitch-> script ("smith","smithpswd","16.17.18.19","/users/
smith","pvc1.scr")
```

```
ATMswitch-> pvc -inputabr -link_1 1:1:100
```

```
ABR PMP circuit reserved on port: 1 InvcI: 100
```

```
Point-to-Point PVC Record committed to flash successfully.
```

```
1
```

```
ATMswitch-> pvc -outputabr -link_1 1:1:100 -link_2 1:2:101
```

```
ABR P-MP Branch added to root at port: 1 from invci: 100
```

```
ABR P-MP Branch added on port: 2 with outvci: 0
```

```
Point-to-Multipoint PVC Branch Record committed to flash
successfully.
```

```
ATMswitch-> pvc -outputabr -link_1 1:1:100 -link_2 1:3:102
```

```
ABR P-MP Branch added to root at port: 1 from invci: 100
```

```
ABR P-MP Branch added on port: 3 with outvci: 0
```

```
Point-to-Multipoint PVC Branch Record committed to flash
successfully.
```



## Managing PVCs

```
ATMswitch-> pvc -outputabr -link_1 1:1:100 -link_2 1:4:103
```

```
ABR P-MP Branch added to root at port: 1 from invci: 100
```

```
ABR P-MP Branch added on port: 4 with outvci: 0
```

```
Point-to-Multipoint PVC Branch Record committed to flash  
successfully.
```

Use the `pvc -all` command to display the created PVCs.

## Deleting PVCs with Scripts

The following script called `dell.scr` removes the PVCs. By specifying 0 as the third argument, the log of this script shows the bootp server as the load source of the script.

```
pvc -rm -branch -link_1 1:1:100 -link_2 1:2:101
```

```
pvc -rm -branch -link_1 1:1:100 -link_2 1:3:102
```

```
pvc -rm -branch -link_1 1:1:100 -link_2 1:4:103
```

```
pvc -rm -root -link_1 1:1:100
```

```
ATMswitch-> script ("smith","smithpswd","0","/users/  
smith","dell.scr")
```

```
pvc -rm -branch -link_1 1:1:100 -link_2 1:2:101
```

```
1 Circuits were unbound on Port 2
```

```
ABR Branch Record deleted from flash successfully.
```

```
value = 0 = 0x0
```

```
pvc -rm -branch -link_1 1:1:100 -link_2 1:3:102
```

```
1 Circuits were unbound on Port 3
```

```
ABR Branch Record deleted from flash successfully.
```

```
value = 0 = 0x0
```

```
pvc -rm -branch -link_1 1:1:100 -link_2 1:4:103
```

## Managing PVCs

```
1 Circuits were unbound on Port 4
ABR Branch Record deleted from flash successfully.
value = 0 = 0x0
```

```
pvc -rm -root -link_1 1:1:100
```

```
1 Circuits were unbound on Port 1
ABR PMP Record deleted from flash successfully.
value = 0 = 0x0
value = 0 = 0x0
```

Use the `pvc -all` command to display the status of the deleted PVCs as demonstrated by the following example:

```
ATMswitch-> pvc -all
```

```
VCIs on Port 1
```

```
VCIs on Port 2
```

```
VCIs on Port 3
```

```
VCIs on Port 4
```

```
value = 0 = 0x0
```

## Chapter 4

---

# Managing Switch Parameters

This chapter describes how to display the switch configuration and manage the following options on the switch:

- Enabled or disabled ports
- Framing support
- Firmware reboot
- Timing source

## Displaying the Slot Configuration

---

### Displaying the Slot Configuration

Use the `switch -slot` command to display the slot configuration. You might see a display that resembles the following:

```
Enet Addr:      FW Rev:      FW String:  
08:00:2b:a7:0e:c0  000750  Release 2.0FT  DAHSF-MA
```

---

## Disabling a Port

Use the `port -dis -port 1:port` command to disable a port, where *port* specifies the port.

Enabling a Port

---

## Enabling a Port

Use the `port -en -port 1:port` command to enable a port, where *port* specifies the port.

---

## Configuring Framing Support

Each port in the module can be individually configured for a particular Physical layer framing structure. You can display or modify the framing support.

### Displaying SDH/SONET Support

To display the framing support for each port in a particular module, use the `port -plf -slot 1` command. You might see a display that resembles the following:

```
Framing Configuration
      Port:      Framing:
          1      SONET
          2      SONET
          3      SONET
          4      SONET
          5      SONET
          6      SONET
          7      SONET
          8      SONET
```

### Modifying SDH/SONET Support

Each port in the module can be individually configured for either SONET or SDH compatibility in the framing structure. Use the following command to configure this parameter:

```
port -plf -sdh|-sonet -port 1:port
```

where *port* specifies the port.

Rebooting the Switch Software

---

## Rebooting the Switch Software

Use the `switch -reboot` command to reboot the switch software. This command disconnects this OBM session and all circuits. The switch reboots automatically.



## Configuring the Transmit Clock Timing Source

Transmit timing sources are established for each port. Each port can be configured to derive its transmit timing from either a local oscillator or the recovered receive clock. Local timing specifies that the transmit timing is derived from the local oscillator. Loop timing specifies that the transmit timing is derived from the recovered receive clock.

When the switch is rebooted, the nonvolatile records for each populated port are examined. If a record is found for the PHY type detected, the stored setting is used. If a record is not found, local timing is used. This implies that if a PHY type is changed while the switch is powered down, switch initialization will not necessarily yield the default setting.

The same sequence of actions occurs if a port is unpopulated at switch powerup, but is populated later.

When modPHY hotswap results in a PHY type change, the new port is initialized with the default setting (local timing). When modPHY hotswap does *not* result in a PHY type change, the link is reinitialized with its last known setting (unless it is overridden by the port conditions described in the next paragraph). Note that hotswap does *not* trigger a nonvolatile record update. So, a subsequent switch reboot might, by virtue of a previously stored record for that link and type, change the timing setting.

Two conditions, loss of signal and loss of frame, will force a loop-timed port to revert to local timing. Subsequent clearing of the condition will force such a port back to loop timing. Neither of these forced transitions will cause a nonvolatile record update.

### Displaying the Transmit Timing Source

Use the port `-tct -port 1:port` command to display the transmit timing source, where *port* specifies the port.

### Changing the Transmit Timing Source

Use the port `-tct -port 1:port` command to change the transmit timing source, where *port* specifies the port.

## Configuring the Transmit Clock Timing Source

## Chapter 5

---

# Managing System Parameters

This chapter describes how to manage the following parameters for your system:

- Time
- Passwords
- Error logs

It describes how to save and retrieve parameters stored in nonvolatile memory (Flash). It also describes how to erase the Flash.

Displaying the Current Date and Time

---

## Displaying the Current Date and Time

The `date` command displays the current date and time as shown in the following example:

```
ATMswitch-> date
```

```
Time since last initialization: 118 hrs, 30 mins, 50 sec
```

```
ATMswitch->
```

---

## Changing Passwords

Use the `passwd -u` command to change the user password and the `passwd -m` command to change the manager password. This command changes the access password, so you must inform other authorized switch users or managers about the new password. Once set, the new password is preserved through power cycles. The default password for the user account is `username` and the default password for the manager account is `managename`.

You are prompted for the old password and the new password when you use this command as shown in the following example:

```
ATMswitch-> passwd -u
```

```
Changing password for user
```

```
Old password: old_password
```

```
This operation changes the user access password, inform  
other authorized switch users of the new password
```

```
Do you want to continue this operation, y or n? (n): y
```

```
Enter new password:
```

```
Verify:
```

```
User account password updated and committed to flash storage.
```

```
ATMswitch->
```

## Displaying Error Logs

You can display or create a file of log information contained in the internal module logs. The error log is a nonvolatile log that is retained through power cycles and keeps only entries that result from the detection of errors. The log is a ring buffer that contains the most recent entries.

To display a specified number of the latest error log entries, use the following command:

```
errorlog -slot 1 -last number
```

where *number* is the number of entries that you want to display.

For example, if you wanted to display the three latest entries for slot 1, you would enter the following command and you might see a display similar to the following:

```
ATMswitch-> errorlog -slot 1 -last 3
[001] start#1, Aug 1 0:00:27 1997, module#23, code#1
      src/app/util/alog.c #370: Master: slot#1, start num 1 Local: slot#1,
start num 1
[002] start#2, Aug 1 0:00:27 1997, module#23, code#1
      src/app/util/alog.c #370: Master: slot#1, start num 2 Local: slot#1,
start num 2
[001] start#3, Aug 1 0:00:27 1997, module#23, code#1
      src/app/util/alog.c #385: Master: slot#1, start num 3 Local: slot#1,
start num 3
Reached the end of error log.
```

A file of the log contents can also be made using the shell scripting capability on the UNIX client workstation. Before establishing the Telnet connection to the module, type the following command in the client shell.

```
% script atmErrorLog.txt
```

This command causes the commands printed to the screen by the switch to also be printed into the `atmErrorLog.txt` file. The procedure can be used for both error and status logs. Turn off the scripting by typing `exit` in the client shell after logging out of the Telnet session to the module.

---

## Clearing All Error Logs

To clear all the error log entries, use the `errorlog -slot 1 -clear` command.

## Managing Parameters Stored In Flash

To manage the parameters stored in Flash, use the `params` command. To use this command, the target host must be running `tftp`.

### Transferring Stored Parameters to the Host

To transfer all the parameters currently stored in Flash as a file to a specified host, use the following command:

```
params -save -ip addr -file name -port y
```

where *addr* is the host's IP address, *name* is the file name to which you want to transfer the stored parameters, and *y* is the port number (usually 0).

### Retrieving Stored Parameters from the Host

To retrieve the parameter file that was saved on the host system using the `params` command and program the parameters into Flash if they are set up manually, use the following command:

```
params -restore -ip addr -file name -port y -overwrite|-nooverwrite
```

where *addr* is the host's IP address, *name* is the file name from which you want to retrieve the stored parameters, and *y* is the port number (usually 0). The `-overwrite` option specifies whether this set of parameters should overwrite the current ones. The `-nooverwrite` option specifies that these parameters should not overwrite existing parameters.



---

## Erasing Flash Storage

Use the `clearflash` command to erase management parameters that have been saved in Flash such as PVC records, LAN Emulation settings, Classical IP settings, and MIB values. Note that the Flash is cleared immediately. The switch does not reboot when clearing Flash.

Erasing Flash Storage

## Chapter 6

---

# Managing Signaling Parameters

This chapter describes the commands used to configure signaling parameters, including:

- UNI versions
- Q.SAAL information
- Q.2931 information

## Configuring UNI Versions

The `sig -uni` command lets users configure the UNI version for an individual link on their ATM switch. It also lets users view the UNI status of all links, store the configuration into Flash for permanent storage, and restore UNI operation mode as stored in Flash.

The physical link is toggled and the new UNI version is in effect 10 seconds after this command is executed.

### Configuring the UNI Version for Each Port

Use the following command to configure the UNI version for a link:

```
sig -uni -ver version -port 1:port
```

where *port* indicates port number and *version* must be one of the following:

0	autoconfigure mode
30	UNI version 3.0
31	UNI version 3.1
40	UNI version 4.0

If the port was in autosensing mode before the change, the following message appears:

```
slot 1 port 2 changed from AUTOSENSING to UNI30
```

If the port is already running UNI 3.0, the following message appears:

```
slot 1 port 2 is already running UNI30
```

### Displaying the UNI Status

To view the protocol status for each port, use the `sig -uni -status` command. This command causes a display similar to the following to appear:

```
UNI Configuration in memory
port 1          <UNI31>
port 2          down
port 3          down
port 4          down
port 5          down
port 6          down
port 7          <UNI31>
port 8          down
```

## Configuring UNI Versions

note: <UNlxx> uni version detected via ILMI

The UNI version indicated by <UNlxx> in the display is detected using ILMI.

### Storing the UNI Configuration

To store the configuration in Flash, use the `sig -uni -wf` command. There is no output message for this command unless an error occurs when writing to Flash.

### Clearing the UNI Configuration

To clear the configuration in Flash, use the `sig -uni -cf` command. There is no output message for this command unless an error occurs when clearing Flash.

### Displaying the Stored Configuration

To display the database stored in Flash, use the `sig -uni -df` command.

### Restoring the UNI Configuration

To restore the UNI configuration stored in Flash, use the `sig -uni -rf` command. There is no output message for this command unless an error occurs when reading Flash.

## Configuring Q.SAAL PDU Types

Use the `sig -stack` command and its options to configure the Q.SAAL PDU types. You can display the statistics or set certain parameters using this command.

You can display the Q.SAAL statistics for a particular port and the Q.SAAL status for all ports.

## Displaying Q.SAAL Link Statistics

Use the `sig -stack -qs -stats -port 1:port` command to display the Q.SAAL statistics for a particular port, where *port* indicates port number. Your output might resemble the following display.

```
QSAAL Statistics Information for Slot 1   Port 1

BGN PDUs Received           0           BGN PDUs Transmitted       0
BGNAK PDUs Received         0           BGNAK PDUs Transmitted     0
BGREJ PDUs Received         0           BGREJ PDUs Transmitted     0
END PDUs Received           0           END PDUs Transmitted       0
ENDAK PDUs Received         0           ENDAK PDUs Transmitted     0

POLL PDUs Received          0           POLL PDUs Transmitted      0
STAT PDUs Received          0           STAT PDUs Transmitted      0
USTAT PDUs Received         0           USTAT PDUs Transmitted    0

RS PDUs Received            0           RS PDUs Transmitted        0
RSAK PDUs Received          0           RSAK PDUs Transmitted      0
SD PDUs Received            0           SD PDUs Transmitted        0
SDP PDUs Received           0           SDP PDUs Transmitted       0
UD PDUs Received            0           UD PDUs Transmitted        0
MD PDUs Received            0           MD PDUs Transmitted        0

Receive Error PDUs          0           Transmit Error PDUs        0
Receive Discarded PDUs      0           Transmit Discarded PDUs    0
Transmit Discarded PDUs     0
```

## Displaying Q.SAAL Link Status

Use the `sig -stack -qs -state -slot 1` command to display the Q.SAAL status for each port on the specified slot. Your output might resemble the following display.

```
QSAAL protocol is DOWN on Slot 1 Port 1
QSAAL protocol is DOWN on Slot 1 Port 2
QSAAL protocol is DOWN on Slot 1 Port 3
QSAAL protocol is DOWN on Slot 1 Port 4
```

## Displaying Q.2931 Link Statistics

---

### Displaying Q.2931 Link Statistics

Use the `sig -stack -q2 -stats -port 1:port` command to display the Q.2931 statistics for a particular port, where *port* indicates port number. Your output might resemble the following display.

```
Q.93B Statistics Information for Slot 1   Port 2

Setup           Msgs Rcvd 0           Setup           Msgs Xmtd 0
Call  Proc      Msgs Rcvd 0           Call  Proc      Msgs Xmtd 0
Connect         Msgs Rcvd 0           Connect         Msgs Xmtd 0
Connect  Ack    Msgs Rcvd 0           Connect  Ack    Msgs Xmtd 0
Release         Msgs Rcvd 0           Release         Msgs Xmtd 0
Release  Comp   Msgs Rcvd 0           Release  Comp   Msgs Xmtd 0
Restart        Msgs Rcvd 0           Restart        Msgs Xmtd 0
Restart  Ack    Msgs Rcvd 0           Restart  Ack    Msgs Xmtd 0
Status         Msgs Rcvd 0           Status         Msgs Xmtd 0
Status  Enq    Msgs Rcvd 0           Status  Enq    Msgs Xmtd 0
Add Party      Msgs Rcvd 0           Add Party      Msgs Xmtd 0
Add Party  Ack  Msgs Rcvd 0           Add Party  Ack  Msgs Xmtd 0
Add Party  Rej  Msgs Rcvd 0           Add Party  Rej  Msgs Xmtd 0
Drop Party     Msgs Rcvd 0           Drop Party     Msgs Xmtd 0
Drop Party  Ack  Msgs Rcvd 0           Drop Party  Ack  Msgs Xmtd 0
Total Conns Established 0           Active Conns 0
Last Rcvd Cause Code    0           Last Xmtd Cause Code    0
Last Rcvd Diagnostic Code 0           Last Xmtd Diagnostic Code 0
```



## Displaying SVC Connection Information

You can display the SVC connection information, such as the number of active SVCs or the SVC connections for each port.

### Displaying Connection Summary

Use the `sig -ncc -cons` command to display the active SVCs for each port. Your output might resemble the following:

```
SVC Connection Matrix
Number Of Active SVCs Per Port
LCP          1
Port1        0
Port2        2
Port3        0
Port4        3
Port5        0
Port6        0
Port7        0
Port8        0
Total Number Of Active SVCs: 6
```

### Displaying Link Connections

Use the `sig -ncc -cpp -port 1:port` command to display the SVC connections for each port, where *port* indicates port number. Your output might resemble the following display.

```
Incoming SVC Connections on Slot 1 Port 2

Callid      Ingress VCI      Egress Slot      Egress Port      Egress VCI

Outgoing SVC Connections on Slot 1 Port 2

Callid      Egress VCI      Ingress Slot      Ingress Port      Ingress VCI
```

Displaying SVC Connection Information

## Chapter 7

---

# Configuring Dynamic Routing

This chapter describes configuration options for dynamic switched virtual circuit (SVC) call routing in the ATM system. It describes the features and functions of ATM routing and the routing configuration options available with the ATM system.

## Overview of Dynamic Routing

The DIGITAL ATM system has automatic routing features that allow the routing of User-to-Network Interface (UNI) SVC calls in a multiswitch ATM network. The ATM system also supports the ATM Forum Interim Interswitch Signaling Protocol (IISP) for signaling SVC calls between switches in a multivendor ATM switch network.

The ATM system has the following automatic routing and addressing features:

- Automatic Topology Discovery
- Shortest Path Routing
- Parallel Links Between Switches and VC Load Sharing
- Static Routing
- Exporting Static Routes
- IISP Support
- IISP Gateway Functionality
- Address Autoconfiguration
- Configurable Switch Prefix
- Configurable Address Prefixes Per Line

### Automatic Topology Discovery

The DIGITAL ATM system automatically discovers the topology of the ATM network by exchanging Link State Packets between switches. When a link initializes, the ATM system determines if the link is to another switch or to an end system. By exchanging Link State Packets on all links between switches, each switch in the ATM network has a complete and identical topological view of the network. Included in the distributed Link State Packets is resource information, such as available CBR bandwidth, for making intelligent routing decisions based on current resource utilization. If a link fails, then all switches in the network are informed of the link failure automatically. As available CBR bandwidth on a link changes, these changes are advertised to other switches in the network so that appropriate call routing decisions are made.

### Shortest Path Routing

SVC call setup requests received at a switch are routed to the destination along the shortest available path. In most cases, the shortest path is based on hop count. For CBR calls, if no bandwidth is available along the shortest path, the ATM system attempts to

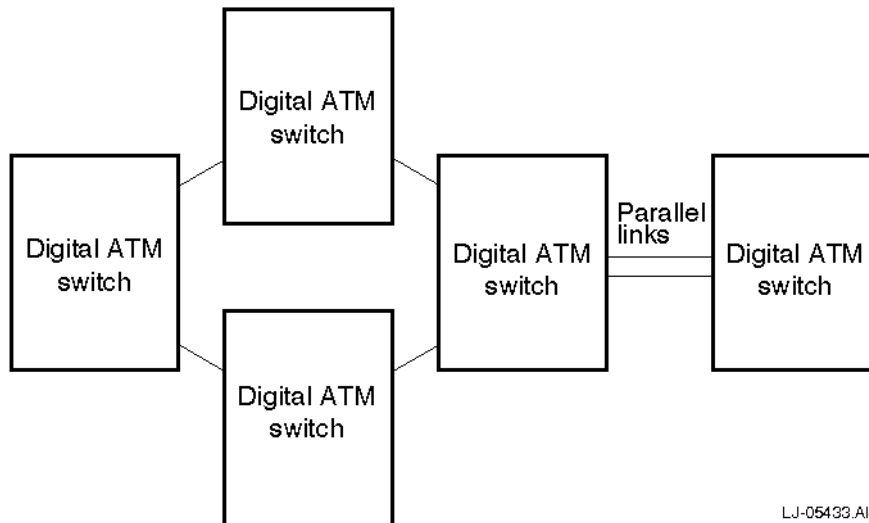
find an alternate path that can support the requested bandwidth. If multiple shortest paths are possible, then the ATM system balances calls over the shortest paths available.

### Parallel Links Between Switches and VC Load Sharing

For increasing available bandwidth between switches, multiple parallel links can be used between switches. Figure 7-1 shows a DIGITAL ATM switch network configuration.

The ATM system balances calls routed between the switches across the parallel links. For ABR calls, the link with the least ABR calls active is selected. For CBR or VBR calls, the link with the most bandwidth available is selected.

**Figure 7-1: DIGITAL ATM Network Configuration Example**



### Static Routing

To provide more flexibility and for use with the ATM Forum IISP, DIGITAL ATM switches provide static routing features. Static routing overrides automatic routing features and allows the network administrator to specify the route that a particular call follows. Static routes are manually configured by the network administrator.

Static routes are based on the Called Party Address received in a UNI SVC Call Setup message. The Called Party Address is the 20-byte ATM Forum NSAP address of the destination ATM end system that a call is destined for. Static routes specify particular

## Overview of Dynamic Routing

patterns that are compared with all calls routed by the switch. If a pattern match is found during call setup, then the call is routed to the particular link specified when the static route is configured.

---

### NOTE

The last byte of the ATM NSAP address is the selector field (SEL Field) and this field is *not* used for static or automatic routing.

---

The ATM system supports static routes based on the first 19 bytes of the ATM NSAP address or on any portion of the ATM NSAP address prefix (bytes 1 to 13). If a 19-byte pattern is specified for a static route, then the Called Party Address in the signaled setup message must match the pattern *exactly* for the static route to be used. A 19-byte static route is a route to a specific end system.

A prefix route can be configured when the network administrator specifies between 1 and 13 bytes as a pattern. In this case, a partial prefix match is performed with all calls received at the switch. When performing a partial prefix match, the switch determines a successful match if all the bits of the configured pattern match the corresponding bits of a called party address prefix.

When configuring such static routes, it is possible that a particular called party address matches many configured static routes. This is possible because multiple partial prefix routes can be configured and full 19-byte address routes can also be configured. If an address matches multiple static routes, the ATM system selects the route with the *best match*. The best match route is the route where most preceding bits of the configured pattern match the called party address being considered. If one such route is a complete 19-byte pattern, then that route is always selected because there can be no better match than all the bits of the pattern.

Static routing works with other ATM routing features. If an address fails to match any configured static route, then the normal ATM automatic call routing features are used to determine the best route to the destination.

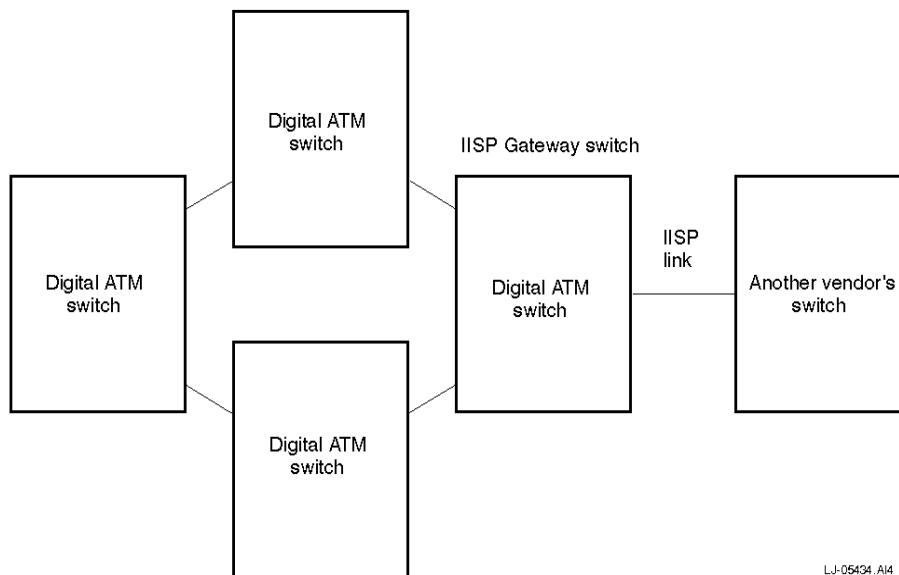
In general, great care must be taken when constructing static routes. The network administrator must ensure that no loops are introduced by improper configuration of static routes. Static routing should only be used when absolutely necessary as automatic routing features ensure optimal routes and adapt to changing network topologies.



## Overview of Dynamic Routing

ATM switches attached to it. The gateway function is achieved by "exporting" all static routes configured for the IISP link. Other DIGITAL ATM switches do not need to be configured. Automatic routing features are used to forward calls to the correct *gateway* switch.

**Figure 7-3: DIGITAL ATM IISP Gateway**



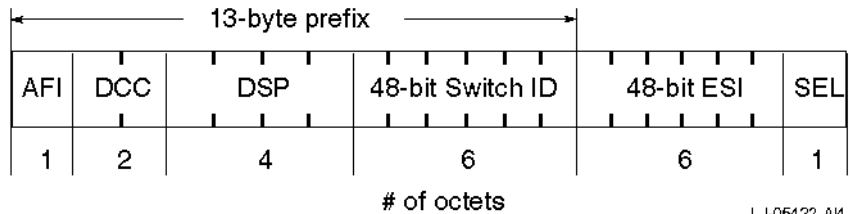
## Addressing Guidelines for DIGITAL ATM Switch Networks

The ATM system conforms to all ATM Forum standards relating to addressing and UNI signaling. This section provides guidelines for address configuration in ATM networks.

The prefix portion is assigned by the network administrator. Each switch in the ATM network should have a unique 13-byte ATM prefix. The ATM prefix is assigned by the network administrator and this prefix is communicated to all end systems attached to the switch during ILMI address registration. The 13-byte ATM address prefix is used to route calls through the ATM network. The ATM system requires that the last 6 bytes of the address prefix be unique for each switch. Therefore, each switch in the DIGITAL ATM switch network has a unique 6-byte switch ID and this switch ID is used as the last portion of the 13-byte address prefix. This addressing convention allows the ATM system to autoconfigure addresses, route calls efficiently, and reduce the amount of routing traffic distributed between switches. For the DIGITAL ATM switch, the address structure used is shown in Figure 7-4.



**Figure 7-4: ATM NSAP Address Format**



LJ-05432.A4

AFI	Address format identifier
DCC	Data country code
DSP	Domain-specific part (first 4 bytes)
Switch ID	Unique ID for this switch
ESI	End system identifier
SEL	Selector field (not used by routing)

**NOTE:**

All portions of the 13-byte prefix are configurable. If the prefix is not configured,

**Address Autoconfiguration**

The ATM system exploits some of the addressing properties described previously to allow for autoconfiguration of all addresses in a DIGITAL ATM switch network. Each ATM system generates a unique 13-byte address prefix and communicates this prefix to all ATM end systems using ILMI address registration to ensure that each addressable entity has a unique address. By default, the prefix is generated and communicated to all end systems supporting ILMI.

The first 7 bytes of the 13-byte prefix is identical for all ATM systems and the remaining 6 bytes are unique for each switch. The 48-bit unique Ethernet address of the switch is used as the unique switch ID when constructing the default prefix.

**Configurable Default Prefix**

The address prefix that the ATM system autoconfigures is called the default prefix. To ensure maximum flexibility, the values used to generate the default prefix can be altered through manual configuration. The values used to generate the default prefix can be controlled by altering the value of a ATM switch ID and by specifying a value to be used for the preceding 7 bytes of the default prefix. If the switch ID is altered, the network administrator must ensure that each switch in the network has a unique ID. The default prefix is registered on all lines. If this is undesirable, then the generation of the default prefix can be disabled altogether.

## Overview of Dynamic Routing

### **Configurable Address Prefixes for Each Port**

Multiple 13-byte prefixes can be registered on each ATM interface individually. When configuring such prefixes, the network administrator must ensure that the lower 6 bytes of the prefix contain the switch ID for that switch. If this is not done, calls are not routed to those addresses from remote switches in the network. Configuring address prefixes for each port is not necessary in most networks, and controlling the content of the default prefix is recommended.

---

## Managing Routing Parameters

You can manage routing parameters using the `decnni` command. This section provides information about managing static routes, the default DEC address prefix, and the ATM address prefix.

### Displaying Routing Statistics

The `decnni` command shows all the addresses learned through ILMI address registration procedures. It also saves the routing parameters in nonvolatile memory (Flash) or clears them from Flash.

The `decnni -sr` command and its options display all the static routes that were configured locally and the routes that were learned from another switch; that is, those routes that were configured on another switch and exported. If a route is learned from another switch, then the switch ID of the relevant switch and the learned route is displayed.

The `decnni -prefix` command displays information about the address prefixes.

### Managing Static Routes

This section describes how to configure, alter, delete, or display static routes using the `decnni -sr` command.

### Configuring a Static Route

To configure a static route, use the following command:

```
decnni -sr -conf atm-address -full | -partial -port 1:port -exp | -noexp
```

where *atm-address* is the ATM prefix in hexadecimal digits and *port* is the port where SVCs are routed if the Called Party Address matches this static route. Enter 0 as your forwarding port if you want to use the local service (for example, LAN emulation).

Use the `-full` option if the address previously entered is the full 20-byte NSAP address of an end system. Use the `-partial` option if a more general address prefix was entered so that a partial match on the address is a successful match.

Use the `-noexp` option if it is a single switch configuration or if other switches do *not* need to route using this static route. Using the `-exp` option causes the ATM system to broadcast this route to all other DIGITAL ATM switches. The network administrator should ensure that only a small number of export routes are configured in the ATM network.

## Managing Routing Parameters

### Altering an Existing Static Route

To alter a static route that was already configured, use the following command:

```
decnri -sr -mod atm-address -full | -partial -port 1:port -exp | -noexp
```

where *atm-address* is the ATM prefix in hexadecimal digits and *port* is the port where SVCs are routed if the Called Party Address matches this static route. Enter 0 as your forwarding port if you want to use the local service (for example, LAN emulation).

Use the `-full` option if the address previously entered is the full 20-byte NSAP address of an end system. Use the `-partial` option if a more general address prefix was entered so that a partial match on the address is a successful match.

Use the `-noexp` option if it is a single switch configuration or if other switches do *not* need to route using this static route. Using the `-exp` option causes the ATM system to broadcast this route to all other DIGITAL ATM switches. The network administrator should ensure that only a small number of export routes are configured in the ATM network.

If you are unsure about previous entries, use the `decnri -sr -showall` command to display all static routes.

If you want to alter the address to be matched or the match characteristics of the route, delete an unneeded route and add another with the desired address prefix and match characteristics.

### Deleting a Static Route

To delete static routes that were already configured, use the following command:

```
decnri -sr -rm atm-address -full | -partial
```

where *atm-address* is the ATM prefix in hexadecimal digits.

Use the `-full` option if the address previously entered is the full 20-byte NSAP address of an end system. Use the `-partial` option if a more general address prefix was entered so that a partial match on the address is a successful match.

### Displaying Static Routes

To display locally configured static routes, use the `decnri -sr -show` command.

To display all the static routes, including those that were configured and learned from other switches, use the `decnri -sr -showall` command.

### Managing the ATM Address Prefix

This section describes how to configure, delete, display, or alter the default ATM address prefix using the `decnri -prefix` command.

## Configuring the Default Address Prefix

Use this configuration option only if the default address prefix used by the DIGITAL ATM switch is unsuitable and you must have a specific 13 bytes for the default prefix.

To configure the default address prefix, use the following command:

```
decnri -prefix -conf atm-prefix -enable | -disable
```

where *atm-prefix* is the 13 bytes of the default address prefix.

Use the `-disable` option if you want to disable the default address prefix feature. If you disable this feature, then the ATM system does not automatically register a prefix on all lines during line initialization. You should use the `-enable` option unless you have specific addressing needs that require manual address configuration.

Configuration and alteration of the default address prefix takes effect upon the next switch reboot.

## Deleting the Default Address Prefix

To disable the default address prefix feature (where a unique ATM address is automatically registered on all lines), configure a default prefix with the `-disable` option.

To delete the configured default address prefix, use the following command:

```
decnri -prefix -rm
```

This command causes the ATM system to return to using the preconfigured prefix for the default prefix upon the next switch reboot.

## Displaying the Default Address Prefix

To display the default address prefix, use the `decnri -prefix -show` command.

## Altering the ATM Address Prefix

To alter a configured prefix, delete the unneeded prefix first and then add the new one.

## Managing Routing Parameters

## Chapter 8

---

# LAN Emulation

This chapter describes the LAN Emulation Server (LES), the Broadcast and Unknown Server (BUS), and LAN Emulation Configuration Server (LECS) functions in the ATM system. It also describes the redundant LAN Emulation functions.

DIGITAL's LAN Emulation (LANE) implementation greatly reduces the amount of effort needed to configure and operate a typical ATM network. In many cases, the network administrator should configure and start this service during the initial installation of the DIGITAL ATM system. LANE should be configured and started if any of the attached stations can operate as a LAN Emulation Client (LEC) and LANE is not configured already. When LANE is configured already, you can configure the ATM system to support existing or additional emulated LANs.

LANE lets existing network applications operate on a native ATM end station without modification by preserving supported packet formats during transmission to the ATM LAN. The ATM system supports LANE features that include:

- A command line interface to instantiate and configure LANE servers in the switch.
- A LECS supporting multiple emulated LANs (ELANs).
- Support for the ILMI Service Registry MIB.
- Multiple LESs to support multiple ELANs.
- Multiple BUSs to support multiple ELANs.
- SNMP MIBs to create, configure, and monitor LANE servers.
- Redundant LANE servers.

## LANE Configuration Overview

The switch administrator must determine the network nodes that will provide the LECS function. Often, switch administrators choose to have one instance of the LECS operating on an ATM network. If the ATM system is chosen to provide the LECS, then it must be enabled because the LECS is disabled in the default configuration. If another DIGITAL ATM switch is providing the LECS, the ATM system uses the established server. Manual configuration is required if more than one LECS is desired or another vendor's switch is acting as the LECS.

The switch administrator must also configure and enable at least one instance of the LES and BUS. Multiple instances of these servers can be created. Each instance would form an emulated LAN. This capability allows the network manager to partition the network into multiple independent LANs for security, traffic control, or other purposes. During initialization, the ATM system automatically configures a LES and a BUS for an Ethernet (IEEE 802.3) LAN and places the servers in a disabled state. For many simple networks, these servers need only be enabled to provide seamless operation of legacy protocols for ATM network stations (connected to the ATM system) that operate as a standard LAN Emulation Client.

The switch administrator uses the command line interface to create, configure, enable, monitor, disable, and delete LANE servers. Server configurations can be saved for restoring after reboot.

The LECS always exists and cannot be deleted. It can only be enabled and disabled. By default, it is disabled. Associated with the LECS is a group of ELAN entries describing the various emulated LANs that the LECS can tell clients to join. Each of these ELAN entries can be independently enabled and disabled.

To create an emulated LAN and advertise it to clients through the LECS, the following actions must be taken using either SNMP or the commands described in Managing Parameters Stored In Flash:

- A BUS must be created.
- A LES must be created and configured with the BUS ATM address.
- An ELAN entry must be created and configured with the LES ATM address.
- The LECS must be configured to direct clients to the ELAN.
- All of the components must be configured to agree on the values for such crucial parameters as LAN type and maximum frame size.
- All of the components must be enabled.

Most LANE server characteristics cannot be modified once the server is running. Exceptions to this rule are noted within the sections for each type of server.



## LAN Emulation Configuration Server

The ATM system allows the operation of one resident LECS that lets the switch provide complete ELAN support without the need for other external LANE service components.

The LECS supports multiple ELANs. Clients can be distributed among multiple emulated LANs by:

- ELAN name, where clients that specify an ELAN name in the Configure request will be directed to the appropriate LESs (if the requested ELANs exist).
- Default ELANs, where clients that specify a blank or invalid ELAN name will be directed to default ELANs that are compatible with their requirements. The network manager can denote several ELANs as default ELANs. For example, there might be a default IEEE 802.3 ELAN and a default IEEE 802.5 ELAN. When two or more default ELANs match a client request, the client is assigned to the lowest-numbered ELAN. Note that while several ELANs with the same LAN type and maximum frame size can be denoted as default ELANs, there is no advantage in doing so.

### Initializing the LECS

At boot time, a check is made for configured LECS information in Flash. This information is used to configure and enable the LECS. If no LECS configuration exists, a default LECS is created, but it is left in the disabled state.

When the LECS is enabled, it starts listening for incoming Configuration Direct VCCs. The LECS listens on the well-known LECS address and its own address for Configure request frames from LAN Emulation Clients (LECs).

The LECS can be enabled as a redundant LECS using the `lecs -redundant` command. Refer to Section for more information about LANE redundancy.

### Connecting to the LECS

A LEC can access the LECS by attempting to connect to the well-known LECS ATM address. The ATM system routes the connection to the LECS well-known address because the LECS well-known address is registered with routing whenever the LECS is enabled. Note that you should configure the route using the `decnri -sr -exp` command so that the route is exported.

The LECS does not support connection timeout. The LEC must terminate the Configure Direct VCC.

## LAN Emulation Configuration Server

### Configuring the LECS

Once the LECS has accepted a Configuration Direct VCC, it discards any frame that is not a Configure request.

Configure requests are processed as follows:

- 1) If the Configure request specifies an ELAN name, and there is an enabled ELAN with that name that is compatible with the client, the LECS assigns the client to that ELAN. Note that if several ELANs have the same name, the LECS considers the one with the lowest-numbered ID only.
- 2) Otherwise, if one or more enabled default ELANs are compatible with the client, the LECS tells the client to use one of them. If possible, it will pick a default ELAN whose maximum frame size exactly matches that of the client.
- 3) If no fit is found, the LECS sends a failure response to the client.

### Managing the LECS

The `lecs` command is used to display all information configured at the LECS. It is used to enable and disable the LECS. You would use the `-disable` option if you have another LECS in the network.

#### Set Attributes

The LECS can only be enabled or disabled.

#### Viewable Attributes

All of the set attributes are viewable attributes. Other attributes (including the LECS address and LECS statistics) can also be viewed for monitoring.

### Displaying the LECS Configuration

Use the `lecs` command to provide configuration information about the LECS, including the state. The following example also shows that the specified LECS is in the disabled state. The LECS can be in the enabled, disabled, disabling, or deleting state.

```
ATMswitch-> lecs -show
```

```
LAN Emulation Configuration Server Parameters
```

```
State: Disabled
```

```
LECS address: 399999000000008002ba5aa80.aa000302ff15.00
```

## Enabling the LECS

Use the `lecs -enable` command to enable the LECS.

Use the `lecs -redundant` command to enable the LECS as a redundant LECS. Refer to Section for more information about LANE redundancy.

## Disabling the LECS

Use the `lecs -disable` command to disable the LECS.

## Deleting the LECS Stored in Memory

Use the `lecs -clrflash` command to delete the LECS stored in Flash.

## Displaying the LECS Counters

Use the `lecs -counters` command to display the LECS counters. Note that counters are never reset to zero. Also, note that the receive queue full field is used to indicate the number of frames that are discarded after the server has passed the queue limit for processing frames as specified by this server. Your display might resemble the following example.

```
ATMswitch-> lecs -counters
```

### LAN Emulation Configuration Server Counters

```
Configuration Direct VCCs
  active:                                0
  releasing:                              0
  attempts:                               0
  failures
    invalid setup:                       0
    no resources:                         0
Frames
  received:                               0
  discarded
    invalid frame:                       0
    receive queue full:                   0
    frames too short:                     0
    other:                                0
```



## LAN Emulation Configuration Server

- Control timeout
- Maximum unknown frame count
- Maximum unknown frame time
- VCC timeout period
- Maximum retry count
- Aging time
- Forward delay time
- Expected LE ARP response time
- Flush timeout
- Multicast Send VCC type
- Multicast Send VCC average rate
- Multicast Send VCC peak rate
- LES ATM address

### Displaying the ELAN Summary

Use the `elan` command to display a summary of the emulated LANs.

```
ATMswitch-> elan -show
```

#### LECS Emulated LAN Summary

Number	Status	Name	Default	Type	Size
1	disabled			802.3	1516

### Displaying the ELAN Configuration

Use the `elan id` command to display information about the specified ELAN, where *id* is the index number associated with the specified ELAN. This example shows that the specified ELAN is in the disabled state. ELANs can be in the enabled or disabled state.

```
ATMswitch-> elan 1
```

#### LAN Emulation Configuration Server ELAN Parameters

## LAN Emulation Configuration Server

```
name:                               control time-out:
type:                               Ethernet/IEEE 802.3    inactive VC time-out:
max frame size:                     1516                ARP entry time-out:
max unknown frame count:            forward delay time-out:
max unknown frame time:             expected ARP response:
max retry count:                   flush time-out:
path switch delay:                 conn. complete time:
local segment ID:

State:                               Disabled

LES address:  00000000000000000000000000000000.000000000000.00

Multicast Send VCC:
type:
average rate:
peak rate:
```

## Creating an ELAN

Use the `elan id -create` command to create an emulated LAN, where *id* is the index number associated with the created ELAN. Although you can create more than 32 ELANs with this command, 32 ELANs is the most that you can use.

## Enabling the ELAN

Use the `elan id -enable` command to enable the ELAN, where *id* is the index number associated with the ELAN.

## Disabling the ELAN

Use the `elan id -disable` command to disable the ELAN, where *id* is the index number associated with the ELAN.

## Deleting the ELAN Stored in Memory

Use the `elan id -rm` command to delete the ELAN stored in Flash, where *id* is the index number associated with the ELAN.

## Creating an ELAN With a LES and BUS

Use the `elanall id -create` command to create an emulated LAN with an associated BUS and LES, where *id* is the index number associated with the created ELAN. Although you can create more than 32 ELANs with this command, 32 ELANs is the most that you can use. The following example creates an ELAN with 7 as its index number and the corresponding LES and BUS.

```
ATMswitch-> elanall 7 -create
```

```
ELAN 7 created.
```

```
LES 7 created.
```

```
BUS 7 created.
```

## Enabling the ELAN With a LES and BUS

Use the `elanall id -enable` command to enable the ELAN, where *id* is the index number associated with the ELAN.

## Disabling the ELAN With a LES and BUS

Use the `elanall id -disable` command to disable the ELAN, where *id* is the index number associated with the ELAN.

## Deleting the ELAN With a LES and BUS Stored in Memory

Use the `elanall id -rm` command to delete the ELAN stored in Flash, where *id* is the index number associated with the ELAN.

## Displaying the ELAN State

Use the `elanall id -show` command (where *id* is the index number associated with the specified ELAN) to display the states of the ELAN, BUS, and LES and the addresses of the LES and BUS.

```
ATMswitch-> elanall 7
```

```
Enabled
```

```
LES 7      LES address:39999900000000008002b223b40.08002b223b53.00  
enabled
```

## LAN Emulation Configuration Server

```
BUS 7   BUS address:399999000000008002b223b40.08002b223b54.00  
enabled
```



---

## LAN Emulation Server

The ATM system can provide multiple LESs. Each LES is configured and enabled through the command line.

### Initializing the LES

At boot time, a check is made for configured LES information in Flash. This information is used to configure and enable the LES. If no LES configuration exists, a default LES is created, but it is left in the disabled state. The default LES is created with default characteristics that are set to the LANE specification with the following exceptions:

- Ethernet/IEEE 802.3 LAN type
- 1516 maximum data frame size
- No ELAN name

When the LES is enabled, it starts listening on its own address for incoming Control Direct VCCs.

### Joining the Emulated LAN

The LEC joins an emulated LAN by creating a Control Direct VCC to the LES ATM address. The B-LLI codepoint in the SETUP message must be properly encoded. The LES accepts the connection if it has resources to support another LEC. The LES then starts the control timeout timer and waits for a Join request. If the request does not come, the LES tears down the connection.

The LES ATM address is unique to a specific emulated LAN.

The Join request is tested to ensure the following:

- 1) The LEC ATM address matches the calling party in the Control Direct VCC.
- 2) The LEC ATM address is not in use by any other LEC on the emulated LAN.
- 3) The LAN type is compatible with the emulated LAN.
- 4) The maximum frame size is either unspecified or greater than or equal to the emulated LAN maximum frame size.
- 5) If the optional MAC address is included, it does not duplicate one that is already registered by another LEC that is a member of this emulated LAN.
- 6) There is enough memory to support minimal LEC database entries.

## LAN Emulation Server

If any test fails, the LES returns a Join response with failure status. If the Control Direct VCC has not been released when the control timeout timer fires, the LES releases the VCC.

If the Join request is valid, the LES creates a Control Distribute VCC to the LEC ATM address by creating a point-to-multipoint VCC if the LEC is the first one to join the emulated LAN or by adding a party to the existing VCC if the LEC is not the first to join. If this connection fails, the LES sends a Join response with failure status to the LEC. If the Control Direct VCC has not been released when the control timeout timer fires, the LES releases the VCC.

If the Control Distribute VCC is created, the LES stops the control timeout timer and returns a Join response with success status. The LAN type and maximum frame size fields reflect the emulated LAN settings.

The response also contains a LEC ID assigned by the LES for use by the LEC while joined to the emulated LAN. The LEC ID may be used by the LEC to identify and filter its own data traffic forwarded by the BUS. It also uniquely identifies the sender of all control and data frames within the emulated LAN.

For each emulated LAN, the LES maintains a database of LECs that have successfully completed the Join phase. The database includes information about each LEC.

For each emulated LAN, the LES maintains a database of LEC ATM addresses for use in verifying that no two LECs share an ATM address.

All Join responses are sent over the Control Direct VCC.

The LES supports all types of emulated LANs.

## Registering MAC Addresses

The LEC can register a MAC address in the Join request. It can also register additional MAC addresses by sending Register requests to the LES after completing the Join phase.

For each emulated LAN, the LES maintains a database that maps MAC addresses to ATM addresses.

The LES accepts Register requests from a LEC that has successfully completed the Join phase. The LES tests the following items before registering the new MAC address:

- 1) The ATM address is not registered by another LEC.
- 2) The MAC address is not registered by another LEC.
- 3) The MAC address is not a multicast address.

## LAN Emulation Server

If the tests succeed, the LES creates an entry in the MAC Address to ATM Address Database and returns a Register response with successful status. If any test fails, the LES sends a Register response with failure status.

The LEC can unregister a MAC address at any time by sending an Unregister request to the LES. The LES validates the request by checking the following items:

- 1) The LEC ID to ATM address pairing is valid.
- 2) The MAC address being unregistered is registered to the LEC.

If the Unregister request is valid, the LES deletes the entry in the database. The LES always returns an Unregister response with successful status.

Registered MAC addresses are not aged from the LES database.

The LEC can also register route descriptors in the same way as MAC addresses.

## Address Resolution

The LES responds to LE ARP requests for registered destinations without forwarding the request to all LECs. The LES sends the LE ARP response using the requesting LECs' Control Direct VCC.

The LES forwards any LE ARP requests it cannot resolve and all LE NARP requests to all the LECs using the Control Distribute VCC.

The Join request identifies whether the LEC is a proxy for unregistered destinations. If it is a proxy, it must receive all LE ARP requests for unknown destinations and respond when appropriate.

Whenever the LES receives an LE ARP response, it forwards it to the requesting LEC using the appropriate Control Direct VCC.

The BUS ATM address is returned by the LES whenever an LE ARP request is received for the broadcast MAC address.

The LES filters all invalid LE ARP requests, including those requesting a multicast address resolution.

## Topology Changes

Topology Change requests are used to help the LECs maintain valid destination to ATM address mappings. The LES receives Topology Change requests from LECs and uses the Control Distribute VCC to forward them to all other LECs. These frames contain information allowing the LECs to adjust their address resolution table aging times to improve address resolution during LAN topology changes.

## LAN Emulation Server

The LES starts a timer when it sends a Topology Change request with the topology change flag set. If the timer expires before the LES receives another Topology request, the LES sends a Topology Change request with the topology change flag cleared.

Topology Change requests have no effect on the LES databases.

## Flushing Data VCCs

When switching from an existing data VCC to a new data VCC, LECs must send a Flush request to flush all pending data in the existing VCC pipeline. This reduces the chance of data being delivered out of order.

The Flush response is sent by the LEC to the LES for forwarding to the requesting LEC. The LES validates the requesting LEC ID to ATM address pairing. If invalid, the LES drops the frame. If valid, the LES forwards the response to the Control Direct VCC for the requesting LEC.

The LES does not respond to Flush requests.

## Terminating the Connection

When a LEC loses a control connection, multicast forward, or distribute connection, it should terminate its membership in the emulated LAN and no longer send traffic until it has rejoined the LAN and received a new LEC ID.

Whenever the LES sees that one control connection is released, it terminates the other control connection and unregisters all destination mappings for the LEC associated with the Control Direct VCC. The LEC ID can be reused by the LES.

## Managing the LES

The `les` command is used to display all information configured at the LES, enable and disable the LES, delete a LES, and set the LES configuration. Any parameters not set by the switch administrator take on the defaults listed in the LAN Emulation Over ATM Specification. The BUS address must be configured.

### Set Attributes

The following attributes can be set by the system manager for each LES:

- Control timeout
- Topology change timeout
- Emulated LAN name
- Emulated LAN type
- Maximum frame size
- BUS ATM address

**Viewable Attributes**

All of the set attributes are viewable attributes. Other attributes (including the LES address and LES statistics) can also be viewed for monitoring.

**Creating a LES**

Use the `les id -create` command to create a LES, where *id* is the index number associated with the created LES.

**Enabling the LES**

Use the `les id -enable` command to enable the LES, where *id* is the index number for the specified LES.

**Disabling the LES**

Use the `les id -disable` command to disable the LES, where *id* is the index number for the specified LES.

**Deleting the LES Stored in Memory**

Use the `les id -rm` command to delete the LES stored in Flash, where *id* is the index number for the specified LES.

**Displaying the LES Summary**

Use the `les -show` command to display a summary of the LESs.

```
ATMswitch-> les -show
```

```
LAN Emulation Servers
```

LES Number	Status	Description
1	enabled	Accounting LAN
2	disabling	Marketing LAN
3	disabled	Under construction
4	enabled	IP VLAN

**Displaying the LES Configuration**

Use the `les id` command to display information about the specified LES, where *id* is the index number for the specified LES. This example shows that the specified LES is in the enabled state. LESs can be in the enabled, disabled, disabling, or deleting state.

## LAN Emulation Server

```
ATMswitch-> les 1
```

### LAN Emulation Server 1 Parameters

```
Description:          Accounting LAN
LAN Name:             Accounting
Type:                 Ethernet/IEEE 802.3
Max frame size:       1516
Control time-out:     120 sec
Topology Change time-out: 300 sec
State:                Enabled

LES address:  31901000000100123456789abc.08002b035817.00

BUS address:  31901000000100123456789abc.08002b035816.00
```

## Displaying the Control VCC Counters

Use the `les id -vcccount` command to display the Control VCC counters, where *id* is the index number for the specified LES. Note that counters are never reset to zero. The client rejected field is used for troubleshooting and indicates to the server that the connect attempt failed. The receive queue full field is used to indicate the number of frames that are discarded after the server has passed the queue limit for processing frames as specified by this server.

```
ATMswitch-> les 1 -vcccount
```

### LAN Emulation Server 1 Control VCC Counters

```
Control Direct
  attempts:                0
  failures
    invalid setup info:    0
    no resources:          0
Control Distribute
  failures
    no resources:          0
```

## LAN Emulation Server

```
client rejected:          0
Frames received:         0
Frames discarded:
  invalid frame:         0
  receive queue full:    0
  LEC not joined:        0
  frame too short:       0
  other:                  0
```

### Displaying the LES Join Counters

Use the `les id -joincount` command to display the LES join counters, where *id* is the index number for the specified LES. Note that counters are never reset to zero. The awaiting request field indicates that the LEC is connected and the LES is waiting for a Join request. The control distribute in process field indicates that the Join request was received and the LES is in the process of creating a Control Distribute to the LEC. The joined field indicates that the LEC has completed joining the LES. The releasing field indicates that the LES is waiting for the LEC to release all connections (Join failed). The receive queue full field is used to indicate the number of frames that are discarded after the server has passed the queue limit for processing frames as specified by this server.

```
ATMswitch-> les 1 -joincount
```

#### LAN Emulation Server 1 Join Counters

```
LEC member states
  awaiting request:      0
  control distribute in process: 0
  joined:                0
  releasing:             0
Join requests:
  received:              0
  duplicates:           0
  discarded
    receive queue full: 0
Join failures
  duplicate ATM address: 0
```

## LAN Emulation Server

```
duplicate destination:          0
invalid ATM address:           0
invalid destination:           0
invalid LAN type:              0
invalid max frame size:        0
invalid LEC ID:                0
invalid parameter:             0
non-duplicate request:         0
control time-out:              0
no resources:                   0
rejected Control Distribute:    0
Join responses sent:           0

Active proxies:                 0
```

## Displaying the LES Register Counters

Use the `les id -regcount` command to display the LES register counters, where *id* is the index number for the specified LES. Note that counters are never reset to zero. The receive queue full field is used to indicate the number of frames that are discarded after the server has passed the queue limit for processing frames as specified by this server.

```
ATMswitch-> les 1 -regcount
```

```
LAN Emulation Server 1 Register Counters

Registered destinations:          0
Register requests
  received:                       0
  discarded
    receive queue full:           0
    no resources:                  0
Invalid register requests
  duplicate ATM address:          0
  duplicate destination:          0
  invalid destination:            0
```



## LAN Emulation Server

```
invalid LEC ID: 0
Register responses sent: 0
Unregister requests: 0
  received: 0
  discarded
  receive queue full: 0
  no resources: 0
Invalid Unregister requests
  invalid address pair: 0
  invalid destination: 0
  invalid LEC ID: 0
  unregistered dest: 0
```

## Displaying the LES ARP Counters

Use the `les id -arpcount` command to display the LES ARP counters, where *id* is the index number for the specified LES. Note that counters are never reset to zero. The receive queue full field is used to indicate the number of frames that are discarded after the server has passed the queue limit for processing frames as specified by this server.

```
ATMswitch-> les 1 -arpcount
```

### LAN Emulation Server 1 ARP Counters

```
LE ARP requests answered: 0
LE ARP requests forwarded: 0
LE ARP requests discarded
  invalid destination: 0
  receive queue full: 0
  no resources: 0
LE NARP requests forwarded: 0
LE NARP requests discarded
  receive queue full: 0
  no resources: 0
LE ARP responses forwarded: 0
LE ARP responses discarded
```

## LAN Emulation Server

```
receive queue full:          0
invalid LEC ID:             0
no resources:                0
```

## Displaying the LES Topology Change Counters

Use the `les id -topocount` command to display the LES topology change counters, where *id* is the index number for the specified LES. Note that counters are never reset to zero. The Topology Change State field is TRUE or FALSE; if it is TRUE, the ELAN is in the topology change state.

```
ATMswitch-> les 1 -topocount
```

```
LAN Emulation Server 1 Topology Change Counters
```

```
Topology Change State:          TRUE
Time in state:                  nnnnnnnnnn seconds
Topology Change requests
  received:                     nnnnnnnnnn
  forwarded:                    nnnnnnnnnn
  discarded
    receive queue full:         nnnnnnnnnn
    no resources:               nnnnnnnnnn
Topology Change requests generated: nnnnnnnnnn
Topology Change request generation failures: nnnnnnnnnn
```

## Displaying the LES Flush Counters

Use the `les id -flushcount` command to display the LES flush counters, where *id* is the index number for the specified LES. Note that counters are never reset to zero. The receive queue full field is used to indicate the number of frames that are discarded after the server has passed the queue limit for processing frames as specified by this server.

```
ATMswitch-> les 1 -flushcount
```

```
LAN Emulation Server 1 Flush Counters
```

```
Flush responses
  received:                      0
```

## LAN Emulation Server

```
discarded
receive queue full:      0
other:                   0
invalid LEC ID:         0
no resources:
Flush responses forwarded: 0
```

### Displaying the Clients Connected to the LES

Use the `les id -clients` command (where *id* is the index number for the specified LES) to display the clients connected to the specified LES, including information about their state and ATM address.

```
ATMswitch-> les 1 -clients
```

```
Connected Clients for LAN Emulation Server 1
```

```
LEC ID    1  Joined          399999000000008002bb51200.08002b223430.00
```

### Displaying Client Address Mappings for the LES

Use the `les id -map` command to display the client address mappings for the specified LES, where *id* is the index number for the specified LES. These mappings can be for the primary address, the MAC address, or the route descriptor (RD). The primary address is the ATM address used by the LEC to connect to the LES. The MAC address or the route descriptor do not have to match the primary address.

```
ATMswitch-> les 1 -map
```

```
Registered Addresses for LAN Emulation Server 1
```

```
LEC ID    1  Primary Addr  399999000000008002bb51200.08002b223430.00
MAC      08002b223430  399999000000008002bb51200.08002b223430.00
```

## Broadcast and Unknown Destination Server

The ATM system can provide multiple Broadcast and Unknown Destination Server (BUSs). Each BUS is configured and enabled through the command line.

### Initializing the BUS

At boot time, a check is made for configured BUS information in Flash. This information is used to configure and enable the BUS. If no BUS configuration exists, a default BUS is created, but it is left in the disabled state. The default BUS is created with default characteristics that are set to the LANE specification with the following exceptions:

- Ethernet/IEEE 802.3 LAN type
- 1516 maximum data frame size

When the BUS is enabled, it starts listening on its own address for incoming Multicast Send VCCs.

### Forwarding Data Frames

Each emulated LAN has a BUS that is uniquely identified by its ATM address.

Every LEC must set up a Multicast Send VCC to the BUS to distribute data frames with a multicast or destination address for which no ATM address is known.

When the BUS accepts an incoming Multicast Send VCC, it creates a Multicast Forward VCC to the ATM address of the accepted connection by creating a point-to-multipoint VCC if the LEC is the first one to connect to the BUS or by adding a party to the existing VCC if the LEC is not the first. If the LEC does not accept the Multicast Forward VCC, the BUS terminates the Multicast Send VCC.

The BUS receives data and control frames on the Multicast Send VCCs. It verifies that the LE header is valid and the length does not exceed the maximum frame size. If the frame checks out, it is sent on the Multicast Forward VCC.

The BUS must be configured with the emulated LAN type only to accept the correct type of incoming Multicast Send VCCs. The BUS does not manipulate the data frames in any way.

### Forwarding Control Frames

The only control frame the BUS can receive is a Flush request. Other control frames are discarded.

## Broadcast and Unknown Destination Server

The BUS forwards all Flush request frames it receives even if its ATM address is the target. The BUS takes no specific action on Flush requests. Flush requests are always queued behind previously received data frames to maintain frame synchronicity.

### Managing the BUS

The `bus` command is used to display all information configured at the BUS, enable and disable the BUS, delete a BUS, and set the BUS configuration. Any parameters not set by the switch administrator take on the defaults listed in the LAN Emulation Over ATM Specification.

#### Set Attributes

The following attributes can be set by the system manager:

- Emulated LAN type
- Maximum frame age

#### Viewable Attributes

All of the set attributes are viewable attributes. Other attributes (including the BUS address and BUS statistics) can also be viewed for monitoring.

### Creating a BUS

Use the `bus id -create` command to create a LES, where *id* is the index number associated with the created BUS.

### Enabling the BUS

Use the `bus id -enable` command to enable the LES, where *id* is the index number for the specified BUS.

### Disabling the BUS

Use the `bus id -disable` command to disable the LES, where *id* is the index number for the specified BUS.

### Deleting the BUS Stored in Memory

Use the `bus id -rm` command to delete the BUS stored in Flash, where *id* is the index number for the specified BUS.

### Displaying the BUS Summary

Use the `bus` command to display a summary of the BUSs.

```
ATMswitch-> bus
```

## Broadcast and Unknown Destination Server

```
LE Broadcast and Unknown Servers
```

BUS Number	Status	Description
1	enabled	Accounting LAN
2	disabling	Marketing LAN
3	disabled	Under construction
4	enabled	IP VLAN

## Displaying the BUS Configuration

Use the `bus id` command to display information about the specified BUS, where *id* is the index number for the specified BUS. This example shows that the specified LES is in the enabled state. BUSES can be in the enabled, disabled, disabling, or deleting state.

```
ATMswitch-> bus 1
```

```
LE Broadcast and Unknown Server 1 Parameters
```

```
Description:           Accounting LAN
LAN Name:              Accounting
Type:                  Ethernet/IEEE 802.3
Max frame size:       1516
State:                 Enabled

BUS address:  31901000000100123456789abc.08002b035816.00
```

## Displaying BUS Counters

Use the `bus id -counters` command to display the BUS counters, where *id* is the index number for the specified BUS. Note that counters are never reset to zero. The Multicast Forward in process field indicates that the LEC is connected to the BUS and the BUS is trying to create a Multicast Forward back to the LEC. The forwarding field indicates that the BUS is forwarding frames to and from the LEC. The releasing field indicates that the LES is waiting for the LEC to release all connections (Join failed). The receive queue full field is used to indicate the number of frames that are discarded after the server has passed the queue limit for processing frames as specified by this server.

```
ATMswitch-> bus 1 -counters
```

## Broadcast and Unknown Destination Server

### Broadcast and Unknown Server 1 Status

```
LEC member states
  Multicast Forward in process:      0
  forwarding:                        0
  releasing:                          0
Multicast Send
  attempts:                           0
  failures
    no resources:                     0
    invalid setup info:               0
    wrong LAN type:                   0
    wrong frame size:                 0
Multicast Forward failures
  no resources:                       0
  client reject:                      0
Frames received:                       0
Frames forwarded:                      0
Frames discarded
  no resources:                       0
  receive queue full:                 0
  invalid length:                     0
  invalid lec state:                  0
  multicast fwd not ready:            0
  rate limited:                       0
  other:                              0
```

## Displaying the Clients Connected to the BUS

Use the `bus id -clients` command (where `id` is the index number for the specified BUS) to display the clients connected to the specified BUS, including information about their state and ATM address.

```
ATMswitch-> bus 1 -clients
```

```
Connected Clients for Broadcast and Unknown Server 1
```

## Broadcast and Unknown Destination Server

Mult Forw in Process	399999000000008002bb51200.08002b123456.00
Forwarding	399999000000008002bb51200.08002b789123.00
Releasing	399999000000008002bb51200.08002b456789.00



---

## Redundant LANE

LANE redundancy is based on a master/standby LECS model where there are at most two LECS per LANE configuration. One of the LECSs is chosen as the master and the other LECS will be in standby mode until the master fails. At this point, the standby will provide all the LANE services. Also, the LECS and the LES/BUS pair work as a group to provide the LANE services. The LECS and LES must be co-located on the same switch. However, it is possible to have a separate BUS entity not co-resident with the LECS and LES. The LES/BUS pair are active for both the master and the standby LECS, but only the LES/BUS pair of the active LECS is used to provide the LANE services. A Hello message is used to keep the state of operation of both the standby and the master LECS active. If the standby LECS determines that the master is no longer active, after a user-configurable time-out value, the standby assumes the master LECS's responsibilities. Upon recovery of the original master LECS, the service can be switched back to the original master depending on a user-configurable option.

The connection between the master and the standby is a bidirectional point-to-point SVC initiated by the selected master. The standby simply listens for the connection request from the selected master.

All the manageable parameters for LANE redundancy is described in the following section. Also, all the aforementioned LECS, LES, and BUS features, requirements and commands are still applicable in a redundant LANE environment. The LECS on both the master and the standby switches must be enabled with the `lecs -redundant` command to use the redundant LANE features.

### Redundant LECS ATM Address

The redundant LECS (RLECS) ATM address is used to set up the SVC between the local and remote LECS. The remote RLECS address must be configured in the local database before enabling the redundancy capability.

### Master Selection

Between the local RLECS and the remote RLECS ATM address, the ATM address that has the highest numerical value is chosen as the master. The selection process is done automatically once the redundancy feature is enabled. You can override the selection process by using the `-master`, `-standby`, and `-default` options.

Redundant LANE

## Restoration Option

The restoration option is enabled by default. That is, if the original master recovers from a failure, the current active LECS will instruct the LES to drop the Control Direct VCC to the LECs, and LECs are forced to rejoin with the original master. However, if disruption of service is undesirable, the restoration option can be disabled and the original master is put in standby mode until the current active LECS fails.

## Hello Timer Interval

The Hello Timer interval is the time between each Hello message. It is set to 2 seconds by default. The allowable range is between 1 and 10 seconds. However, the interval has to be less than the Timeout Timer interval.

## Timeout Timer Interval

The Timeout Timer interval is set to 4 seconds by default. Along with the Dead factor parameter, you can finetune the time to take action for a missing event. The allowable range for the Timeout Timer interval is between 1 and 20 seconds, and this interval has to be greater than the Hello Timer interval.

## Dead Factor

The Dead factor is defined as the missing event threshold. For example, if some action should occur after 3 timeout events have occurred without any hello messages being received, the Dead factor should be set to 3.

## Connection Attempt

This is defined as the number of attempts by the selected master to initiate the SVC connection between the master and the standby upon enabling redundancy. The selected master assumes the standby is not active after the number of attempts has exceeded the predefined value and no response has been received from the standby.

## Managing the Redundant LECS

The `rlecs` command is used to configure the redundant LECS (RLECS).

## Displaying the RLECS Parameters

Use the `rlecs -params` command to display the RLECS parameters. Your display might resemble the following:

```
Redundant LECS Parameters

restoration option:  ENABLED           master           :  LOCAL
hello interval      :           2      timeout interval  :           4
```

## Redundant LANE

```
dead factor      :          3          connection attempt :          2
state            :    ACTIVE          administrative state:  ENABLED
override mode    :    MASTER

Local lecs address : 3999990000000000000f8465380.0000f8465390.00
Local rlecs address : 3999990000000000000f8465380.0000f8465393.00
Remote rlecs address: 0000000000000000000000000000.000000000000.00
Remote lecs address : 0000000000000000000000000000.000000000000.00
```

### Configuring the Remote RLECS Address

Use the `rlecs -remote address` command to configure the remote RLECS address.

### Enabling the Restoration Option

Use the `rlecs -restore` command to enable the restoration option. This option lets a master LECS that has failed and been replaced by a slave LECS to be restored as the master LECS again.

### Disabling the Restoration Option

Use the `rlecs -norestore` command to disable the restoration option.

### Changing the Hello Timer Interval

Use the `rlecs -hello time` command to change the Hello Timer interval.

### Changing the Timeout Timer Interval

Use the `rlecs -timeout time` command to change the Timeout Timer interval, where *time* is the interval in seconds.

### Changing the Dead Factor Parameter

Use the `rlecs -deadfactor value` command to change the dead factor parameter, where *value* is the number of timeouts to wait for a response from the RLECS before deciding that it is down.

### Changing the Connection Attempt Parameter

Use the `rlecs -cons value` command to change the connection attempt parameter, where *value* is the number of attempts to try connecting to the RLECS before deciding that it is down.

Redundant LANE

## Storing the RLECS Parameters

Use the `rlecs -wrflash` command to store the RLECS parameters in nonvolatile memory. This action is performed automatically when the LECS is enabled.

## Displaying the Stored RLECS Parameters

Use the `rlecs -rflash` command to display the RLECS parameters stored in nonvolatile memory. Your display might resemble the following:

```
Redundant LECS NV info

hello interval      :          2          timeout interval      :          4
dead factor         :          3          connection attempt    :          2
restoration option:  ENABLED          override option       :  DEFAULT

Remote rlecs address: 3999990000000008002ba5ac00.08002ba5ac15.00
```

## Changing the Master Selection Criteria

Use the `rlecs -master`, `rlecs -standby`, and `rlecs -default` commands to override the default master selection criteria. The `rlecs -default` command sets the master/standby override option to default mode. The `rlecs -master` command sets the master/standby override option to master mode. The `rlecs -standby` command sets the master/standby override option to standby mode.

## Chapter 9

---

# Managing the System Using SNMP

An extended LAN containing ATM systems can be managed with a network management station (NMS) using the Simple Network Management Protocol (SNMP). SNMP is a protocol used for communicating between a network management system and an agent (in this case, the ATM system).

Using NMS-specific commands, you can:

- Examine ATM characteristics, including firmware version, hardware status, and module status.
- Examine the characteristics of ports and counters.
- Examine port status and address information.
- Enable or disable a port connection to the LAN.
- Monitor events.

---

## SNMP Management Concepts

If you need additional information about using SNMP management, Table 9-1 lists some additional sources of information.

**Table 9-1: Additional Information About SNMP**

<b>For information about:</b>	<b>Refer to:</b>
SNMP and manageable objects	<i>The Simple Book</i> by Marshall T. Rose
The at, icmp, if, ip, snmp, sys, or udp groups	Appropriate RFC
An SNMP object	Appropriate public or private MIB
Manipulating SNMP objects using your NMS	Appropriate user documentation

SNMP communicates management information between a manager and an agent. SNMP allows the network manager to view or modify (`get` or `set`) information from an agent. The manager requests the information using SNMP, and the agent sends back the information. Note that with SNMP, you can create objects when you set them for the first time.

Requests for information, or commands to modify information, are often accompanied by a password called a community name. Because only users who are retrieving or modifying information know the community name, this is a simple form of security.

The information that can be requested or modified from the agent consists of a number of objects. Taken together, these objects form the Management Information Base (MIB). These MIBs can be loaded on an NMS, and all of the objects defined in the MIBs can then be viewed or modified as appropriate.

The Management Information Base was defined by the Internet Engineering Task Force (IETF) for SNMP management. The MIB is a repository for all data (variables) gathered and used by a network node. It describes the name, object identifier, data type, and accessibility of all data items that can be read or written using SNMP. An object or object identifier's structure is like that of the UNIX file system. It can be represented as a tree-like, branched structure that spreads downward from a root. Below the root is a configuration-dependent number of managed objects that are defined as software resources that enable network services.

MIB object groups are placed at various points in the MIB tree. MIB objects can be referred to by their name or their number in the MIB tree. When you specify a MIB object, you must give its full name or object ID number, or you must specify its leaf name (which can often be done using different methods that depend on your NMS).

To access a specific variable within the MIB, you must supply the complete variable name (object name). These object names mirror the hierarchical structure of the MIB, and are made up of the sequence of the branches of the MIB tree that lead down to the object.

## SNMP Management Concepts

A MIB object can be entered on an NMS by either typing the complete object name, typing the complete object ID, or using a window-based system to click on each correct sequential tree branch until the desired object is fully specified.

## Initial NMS Commands

When you first use the NMS, you might want to:

- Familiarize yourself with your NMS's command syntax.
- Set a default gateway configuration.
- Set your community name (password) for read-only users, read-write users, and traps.

Different software packages for your NMS require different command syntax structures. The syntax often contains an action (such as `set` or `get`), a variable (a MIB object), a type (such as an octet or an integer), and a value. Set commands must often identify an action, an object, the type of string that an object accommodates, and the value that you are giving it.

This section describes how to configure the following parameters to support SNMP management:

- Read-write community name
- Read-only community name
- Trap community name
- NMS IP trap address
- ILMI registration
- Link information

### Displaying the SNMP Agent Configuration

Use the `snmp -c` command to display the SNMP agent configuration. This command would display the community strings and the destination IP trap address.

### Configuring the SNMP Agent

You can configure the SNMP agent by modifying the community strings and destination IP trap addresses using the `snmp` command and its options.

To provide access to the SNMP parameters, you must know the read-write community name, the read-only community name, the trap community name, the NMS IP trap address.



## Modifying the Read-Write Community Name

Use the `snmp -c -rw comm-name` command to modify the read-write community name, where *comm-name* specifies the community name.

## Modifying the Read-Only Community Name

Use the `snmp -c -ro comm-name` command to modify the read-only community name, where *comm-name* specifies the community name.

## Modifying the Trap Community Name

Use the `snmp -c -tc comm-name` command to modify the trap community name, where *comm-name* specifies the community name.

## Adding the NMS IP Trap Address

Use the `snmp -c -ta addr` command to add an NMS IP trap address, where *addr* is the destination IP address for SNMP traps that must be specified in dot notation. You can have up to eight destinations.

## Deleting the NMS IP Trap Address

Use the `snmp -c -td addr` command to delete an NMS IP trap address, where *addr* is the destination IP address for SNMP traps that must be specified in dot notation.

## Action Enabling or Disabling ILMI Registration

Use the `trace -ilmi -on` command to enable ILMI registration for all ports. Use the `trace -ilmi -off` command to disable ILMI registration for all ports. You can also specify that particular ports or paths be enabled or disabled. To make this change effective, you must disconnect and reconnect the cable for the specified port.

```
address registration is enabled on link 3
disconnect and reconnect cable on the link in order
this change to take effect
```

## Displaying Link Information

You can display the link status for all the ports. You can also display the registered addresses.

## Displaying the Status of All Ports

Use the `status -link` command to display the link status for all the ports.

```
link 1.1: NOT CONNECTED HEAD WAIT Carrier Dn ESC N FM N Prefixes OK
link 1.2: NOT CONNECTED HEAD WAIT Carrier Dn ESC N FM N Prefixes OK
```

## Initial NMS Commands

```
link 1.3: REGISTERED   HEAD GOOD Carrier Up ESC Y FM Y All prefs rejected
link 1.4: REGISTERED   HEAD GOOD Carrier Up ESC Y FM Y All prefs rejected
```

## Displaying Registered Addresses

Use the `decnmi -showreg` command to display the registered addresses.

```
no addresses
```

---

## ATM SNMP Implementation

The SNMP implementation in the ATM system consists of two components: the Address Registration Unit (ARU) and the SNMP agent.

The Address Registration Unit is an SNMP client that performs ATM address registration according to the UNI specification. The address registration process establishes a unique ATM address for each of the end nodes connected to the ATM links. Signaling software uses ATM addresses for establishing virtual circuits. Routing software uses ATM addresses for routing calls through the ATM network. The ARU also sends keep-alive messages to its peers to detect UNI down events.

The SNMP agent supports management operations on the ILMI VCI (VCI=16) on each of the ATM links, and also over the Ethernet/IP/UDP path. The access over the ILMI VCI is limited to the following:

- System group of MIB-II
- ATM-FORUM-MIB
- ATM-FORUM-ADDR-REG
- DEC-ATM-MIB

### Address Registration Unit

The ARU maintains an `atmfAddressTable` that contains ATM address information for each of the ATM links. During initialization, the switch sets the table to the empty state. Whenever a carrier is detected on an ATM link, a cold start trap is received, or polling fails for 20 seconds, the switch deletes the entries associated with that link.

ARU sends a `coldStart` trap to its peer at the beginning of the address registration process. It skips sending the `coldStart` trap if the `coldStart` trap is already received from its peer. It sends a `GetNext` request to its peer to verify that no entries are associated with the line in the peer's `atmfNetPrefixTable`. If any entries are found, it restarts the registration process by sending a `coldStart` trap. If the peer's `atmfNetPrefixTable` is found empty, the switch sends SNMP `Set` requests to add one or more entries to the peer's `atmfNetPrefixTable`.

The peer has to perform an SNMP `Set` operation on the local `atmfAddressTable` to establish its ATM addresses. The peer can add additional ATM addresses or delete addresses previously registered by sending SNMP `Set` requests. The peer can also send `Get` and `GetNext` requests at any time during or after the address registration process.

## ATM SNMP Implementation

### **SNMP Agent**

The SNMP agent provides a general SNMP interface to all the management information available. The SNMP agent supports receiving SNMP requests from two sources:

- UDP/IP on Ethernet
- Raw ATM (AAL5) on VCI=16 on ATM links

The SNMP agent provides access to all the MIB variables according to the community name for all the requests received over the IP path. The agent provides read-only access to all the MIB variables and read-write access to a limited set of entries in the ATM address table for raw ATM (SNMP/AAL5) requests received over VCI 16 on an ATM link. This set is limited to the entries corresponding to the line over which the request is received.

The DEC-ATM-MIB contains device and link attributes, such as FLOWmaster support and switch type (whether it is a DIGITAL switch). This information is exchanged during link initialization and used when VCs are set up on the link.

---

## Implemented MIB Object Groups

The MIBs supported by the ATMswitch 900 module include:

- MIB-II (RFC 1213)
- ILMI MIB (UNI V3.0)
- DS1/E1 MIB (RFC 1406)
- DS3/E3 MIB (RFC 1407)
- SONET MIB (RFC 1595)
- ATM MIB (RFC 1695)
- DEC ATM MIB (V1.0)
- DEC ATM Signaling MIB
- DEC ATM Switch Chassis MIB
- DEC LAN Emulation Service MIBs (LES, BUS, LECS)
- Common DEChub 900 Objects MIB

Consult the MIBs directly for complete information.

Table 9-2 lists the object groups implemented by the supported MIBs.

**Table 9-2: MIB Object Groups**

<b>MIB</b>	<b>Object Groups</b>
MIB-II	system, interfaces, at, ip, icmp, tcp, udp, snmp
ILMI MIB	
DS1/E1 MIB	DS1/E1 Near End Group
DS3/E3 MIB	DS3/E3 Near End Group
SONET MIB	sonetObjects, sonetObjectsPath, sonetObjectsVT, sonetConformance
ATM MIB	atmMIBObjects, atmMIBConformance
DEC ATM MIB	ema
DEC ATM Signaling MIB	
DEC ATM Switch Chassis MIB	Excluding power, fan, and temperature groups
DEC LES MIB	decLesConfigGroup, decLesClientStatesGroup, decLesClientGroup, decLesLecGroup, decLesMacAddressGroup, decLesRouteDescrGroup, decLesTrafficGroup, decLesCallStatsGroup, decLesJoinStatsGroup, decLesRegisterStatsGroup, decLesArpStatsGroup, decLesFlushStatsGroup, decLesEventLogGroup

## Implemented MIB Object Groups

<b>MIB</b>	<b>Object Groups</b>
DEC BUS MIB	decBusConfigGroup, decBusClientStatesGroup, decBusTrafficGroup, decBusClientGroup, decBusLecGroup, decBusCallStatsGroup, decBusEventLogGroup
DEC LECS MIB	decElanCConfGroup, decLecsCStatGroup, decLecsCGroup, decLecsCFaultGroup
Common DEChub 900 Objects	

---

## ATM SNMP Agent

This section describes the capabilities of the ATM SNMP agent (`gigaSwitchATMAgent`). It supports RFC1213-MIB and includes the `system`, `at`, `ip`, `icmp`, `tcp`, `udp`, and `snmp` groups with the following variations:

`atEntry`

**Description:** Creation requires `atPhysAddress`.

`ipForwarding`

**Syntax:** integer (1..1)

**Description:** Always set to 1.

`ipDefaultTTL`

**Syntax:** integer (255..255)

**Description:** Always set to 255.

`ipRouteEntry`

**Description:** Creation requires `ipRouteNextHop` and `ipRouteType`.

`ipNetToMediaEntry`

**Description:** Creation requires `ipNetToMediaPhysAddress` and `ipNetToMediaType`.

`tcpConnState`

**Access:** read-only

**Description:** Cannot be set in the ATM system.

The ATM SNMP agent supports the IF-MIB and includes the `ifGeneralGroup`, `ifPacketGroup`, `ifHCFixedLengthGroup`, `ifStackGroup`, `ifInNUcastPkts`, `ifOutNUcastPkts`, `ifOutQLen`, `ifSpecific`, and `ifPromiscuousMode` groups.

Seven kinds of interfaces are supported by the current release of the SNMP agent in the ATM system: loopback, Ethernet, ATM, SONET path, SONET line, DS1/E1, and DS3/E3. The `ifGeneralGroup` is supported for interfaces of all types. The `ifPacketGroup` is supported for the loopback and Ethernet interfaces. Also, `ifInNUcastPkts`, `ifOutNUcastPkts`, `ifOutQLen`, and `ifSpecific` are supported for loopback and Ethernet interfaces. The `ifPromiscuousMode` can be read for some interfaces, but promiscuous mode cannot be set for any of them. The `ifIndex` values are assigned as follows:

## ATM SNMP Agent

- The software loopback interface has `ifIndex` 1.
- The Ethernet interface has `ifIndex` 2.
- All other `ifIndex` values are calculated as `BASE + LINK_NUMBER`, where `BASE` is an even multiple of 1000 that identifies the interface layer (refer to Table 9-3). The `LINK_NUMBER` is calculated as:

*port\_number*

Note that decoding the `ifIndex` value is not the only method for getting configuration information. The `ifDescr`, `ifType`, and `ifStackTable` objects and the DEC ATM Switch Chassis MIB all provide help for obtaining this information.

**Table 9-3: Interface Numbering**

<b>Interface Layer</b>	<b>ifIndex Range</b>
ATM	5001-5xxx
SONET path	6001-6xxx
SONET section/line/medium	7001-7xxx
DS3/E3	8001-8xxx
DS1/E1	9001-9xxx

The following variations are supported for the IF-MIB.

`ifAdminStatus`

**Syntax:** integer - up(1), down(2)

**Description:** This is applicable to Ethernet and ATM interfaces only (indexes: 2 and 5XXX).

`ifAdminStatus`

**Syntax:** integer - up(1)

**Description:** This is applicable to loopback, SONET path, and SONET line interfaces only (indexes 1, 6XXX and 7XXX).

`ifHCInOctets`

**Access:** not-implemented

**Description:** Counter64 is an unknown type for SNMPv1.

`ifHCOutOctets`

**Access:** not-implemented

**Description:** Counter64 is an unknown type for SNMPv1.



## ATM SNMP Agent

The ATM SNMP agent supports the SONET-MIB and includes the `sonetMediumStuff` group.

It supports the DS3/E3 and DS1/E1 MIBs, including the near end groups (but not the far end or fractional groups).

It supports the ATM-MIB and includes the `atmInterfaceConfGroup` and `atmInterfaceTCGroup` groups with the `atmInterfaceMaxVpcs`, `atmInterfaceMaxVccs`, `atmInterfaceMaxActiveVpiBits`, `atmInterfaceMaxActiveVciBits`, `atmInterfaceIlmiVpi`, and `atmInterfaceIlmiVci` variations with read-only access.

The ATM SNMP agent supports the ATM-FORUM-MIB and includes the `atmfPhysicalGroup` and `atmfAtmLayerGroup` group. The `PortIndex` value for all tables is equal to the `ifIndex` for the corresponding ATM interface minus 5000. ATM-FORUM-MIB variables can be accessed through both IP and raw ATM.

It supports the ATM-FORUM-ADDR-REG MIB and includes the `atmfAddressGroup` group. The `atmfAddressPort` value is equal to the `ifIndex` for the corresponding ATM interface minus 5000. The ATM-FORUM-ADDR-REG variable can be accessed through both IP and raw ATM.

It supports the DEC-ATM-MIB and includes the `ad` group. The `adpPortIndex` value is equal to the `ifIndex` for the corresponding ATM interface minus 5000. DEC-ATM-MIB variables can be accessed through both IP and raw ATM.

ATM SNMP Agent

## Appendix A

---

# ATM Command Summary

This appendix describes all the ATM commands that you can use to manage your DIGITAL ATM switch. These commands are:

bus	Configure or display the LAN Emulation BUS characteristics
clearflash	Clear all parameters stored in nonvolatile memory (Flash)
clip	Control Classical IP setup
date	Display the current date and time
decnri	Perform various routing functions
diagnose	Check for hardware problems and configuration errors
elan	Create or configure emulated LANs (ELANs)
elanall	Configure or create emulated LANs (ELANs)
errorlog	Display or clear error log entries
h	Display or set the shell history
help	Display help on selected topics
host	Process Internet address and host associations
iam	Set user ID and password
ifconfig	Configure or display network interface parameters
lecs	Control LAN Emulation Configuration Server
les	Configure or display LES characteristics
link	Configure a link
loadconfig	Directs the downloading or uploading of both the kernel and application images
netstat	Show network status
params	Save/restore a parameter file to/from a host system
passwd	Set the user or master password
password_disable	Disable password protection for user and manager accounts
password_enable	Enable password protection for user and manager accounts
plcp	Set the E1 line type
port	Configure a port
pvc	Allows configuration of the PVCs
restart_factory_defaults	Reset the module with its original default settings
rlecs	Display or configure redundant LECS parameters
script	Specify remote command file for execution by the switch

service	Configure ATM Forum Service Registry table
set_gateway_IP_address ("ip_gateway_adr")	Set an IP address
shellLock(1)	Disable Telnet access to the module
shellLock(0)	Enable Telnet access without having to reset the module
show_port_status	Check the operation of the Physical layer protocols, circuitry, and cables
sig	Perform various signaling functions
snmp	Examine and set SNMP MIB variables, or configure SNMP agent
status	Display comprehensive switch status
switch	Configure the ATM system
task	Display task information
trace	Set up tracing capabilities
versions	Display the version numbers for the switch software
whoami	Display the user's name

For these commands, you can specify ports 1 through 4 for your port values on either ATM system.

You have the following options when specifying a slot value:

ATM system (14-slot)	1 through 6, 8 through 14
ATM 5-slot system	1 through 4
ATMswitch 900	1

---

## bus

### Name

`bus` — Configures or displays the LAN Emulation BUS characteristics

### Syntax

```
bus id [ unique_option ]
```

```
bus id [ option_list ]
```

### Description

The `bus` command is used to create and configure the Broadcast and Unknown Server (BUS), enable and disable the BUS, delete a BUS, and display the BUS configuration. Most forms of the `bus` command require that the `id` be specified, where `id` is the index number of the BUS for which you want to configure or display information and has a value between 1 and 99. Any parameters not set by the switch administrator take on the defaults listed in the LAN Emulation Over ATM Specification.

The `bus show` command displays the list of BUSs.

If you choose to specify a unique option, you must specify only one option. You can specify more than one option from the option list, but the options must be separated by a space. If you specify a unique option, you must not specify any options from the option list.

### Unique Options

<code>-create   -c</code>	Creates the specified BUS.
<code>-counters   -co</code>	Displays counters for the specified BUS.
<code>-clients   -cl</code>	Displays clients connected to the specified BUS.
<code>-enable   -en</code>	Enables the BUS and causes the BUS to accept multicast send VCCs.
<code>-disable   -dis</code>	Disables the BUS and causes the BUS to terminate all multicast send and forward VCCs.
<code>-rm</code>	Deletes the BUS and removes the numbered BUS from memory and Flash.

## bus

### Options

-name   -n " <i>string</i> "	Sets the ELAN name to the specified double quoted string that is up to 32 characters long and is in the format of an SNMPv2 display string. The <i>string</i> must be separated from the -name option by at least one space and must be enclosed in double quotes if it contains multiple words.
-noname   -non	Clears the ELAN name.
-type   -t <i>type</i>	Sets the ELAN type; <i>type</i> must be 8023 (Ethernet/IEEE 802.3) or 8025 (IEEE 802.5).
-size   -s <i>size</i>	Sets the ELAN maximum frame size. The maximum size frame that all members of the ELAN must be capable of receiving; <i>size</i> must be 1516, 4544, 9234, or 18190.
-maxfwdrate   -mf <i>rate</i>	Sets maximum forwarding rate from 100-2400 kb/s.
-descript   -des " <i>string</i> "	Sets the description for the BUS to the double quoted string. The <i>string</i> must be separated from the -descript option by at least one space and must be enclosed in double quotes if it contains multiple words.
-nodescript   -nodes	Clears the description for the BUS.

### Examples

To get help on the BUS command, enter:

```
bus -help
```

To create a BUS with an index number of 3, enter:

```
bus 3 -create
```

To specify BUS 2's ELAN type as Ethernet with a frame size of 9234, enter:

```
bus 2 -type 8023 -size 9234
```

To display the connected clients for BUS 1, enter:

```
bus 1 -clients
```

The resulting display looks like the following:

```
Connected clients for Broadcast and Unknown Server 3
Mult Forw in Process
39999900000000008002bb51200.08002b123456.00
Forwarding 39999900000000008002bb51200.08002b123456.00
Releasing 39999900000000008002bb51200.08002b123456.00
```

---

## clearflash

### Name

`clearflash` — Clears all parameters stored in nonvolatile memory (Flash).

### Syntax

`clearflash`

### Description

The `clearflash` command clears the parameters stored in Flash. These parameters include the PVC configuration, the LAN Emulation parameters, and the Classical IP parameters.

---

### NOTE

This command clears the Flash immediately. You are not prompted for verification about command completion.

---

### Examples

To clear the Flash, enter:

```
clearflash
```

clip

---

## clip

### Name

clip — Control Classical IP setup

### Syntax

clip [ *unique\_option* ]

### Description

The `clip` command configures and displays the switch Classical IP (CLIP) configuration. If you do not specify any parameters, the command lists the usage summary.

### Unique Options

-show	Displays the CLIP status, including the switch ATM IP address, the switch ATM IP mask, the switch ATM NSAP address, and the server ATM NSAP address.
-enable   -en	Enables the CLIP client on the switch.
-disable   -dis	Disables the CLIP client on the switch.
-ipaddr   -ip <i>address</i>	Sets the IP address for the local CLIP client.
-ipmask   -ip <i>mask</i>	Sets the IP address mask for the local CLIP client.
-server   -s <i>address</i>	Sets the CLIP server address (NSAP).
-pvc 1:] <i>port:vci</i>	Sets up an ABR point-to-point CLIP connection.
-atmarp   -a	Displays the contents of the ARP cache for the ATM interface.
-counters   -co	Displays the IP counters for ATM.
-clrflash   -cf	Clears the CLIP configuration from Flash.



date

---

## date

### Name

date — Displays the current date and time.

### Syntax

date

### Description

The `date` command displays the current date and time. It also indicates how long the system has been running.

### Examples

To display the date and time, enter:

```
date
```

---

## decnni

**Name**

decnni — Performs various routing functions.

**Syntax**

```
decnni [ unique_option ]
```

```
decnni -sr sr_option
```

```
decnni -prefix prefix_option
```

**Description**

The `decnni` command configures static routes and the switch default prefix (ATM NSAP address prefix). The `decnni` command lists the usage summary when no parameter is supplied. Commands must be entered in the format `decnni [-action] address_options`, where *-action* is either `-sr` or `-prefix`. If there are multiple parameters, they can be entered in any order as long as they follow the action and the address.

**Unique Options**

<code>-wrflash   -wf</code>	Saves the DEC NNI parameters to Flash.
<code>-clrflash   -cf</code>	Clears the DEC NNI parameters from Flash.
<code>-showreg   -sreg</code>	Displays addresses registered through ILMI.

**-sr Options**

<code>-show</code>	Displays locally configured routes.
<code>-showall</code>	Displays static routes learned from other switches and locally configured routes.
<code>-conf <i>atmaddress</i> -full   -partial</code>	Configures a new static route.
<code>-port 1:<i>port</i> -exp   -noexp</code>	
<code>-mod <i>atmaddress</i> -full   -partial</code>	Modifies an existing static route.
<code>-port 1:<i>port</i> -exp   -noexp</code>	
<code>-rm <i>atmaddress</i> -full   -partial</code>	Removes a static route.

**-prefix Options**

<code>-conf <i>atmprefix</i> -enable   -disable</code>	Configures the DEC address prefix.
<code>-show</code>	Displays the DEC address prefix.
<code>-rm</code>	Removes the DEC address prefix.

**Examples**

To configure and export the new static route 47.0005.80.123456.1234.1234.1234 for port 3, full match, enter:

decnni

```
decnni -sr -conf 47.0005.80.123456.1234.1234.1234 -full -port  
1:3 -exp
```

To remove the DEC address prefix 47.0005.80.123456, enter:

```
decnni -prefix -rm 47.0005.80.123456
```

diagnose

---

## diagnose

### Name

diagnose — Checks for hardware problems and configuration errors.

### Syntax

diagnose -check

### Description

The `diagnose -check` command provides a simple way to check for hardware problems, such as ports that are down or disabled. It also lets you check configuration errors, such as LAN Emulation configuration errors. This command displays the status.

---

### NOTE

This command does not detect everything that can go wrong. For instance, it does not detect configuration problems involving static routes.

---

### Examples

To check the status and display the results, enter:

```
diagnose -check
```

The resulting display resembles the following:

```
Starting diagnostic run...
```

```
Checking slots and ports...
```

```
Card 1 is OK ... checking ports.
```

```
Up    = (none)                Disabled = (none)
```

```
Down  = (1 2 3)              Testing  = (none)
```

```
Card 3 is OK ... checking ports.
```

```
Up    = (2 3)                Disabled = (none)
```

```
Down  = (1 4)              Testing  = (none)
```

```
Clock card is OK.
```

```
Checking power supply controller...
```

diagnose

Checking power supplies and battery...

Checking fans and temperature...

Checking ILMI connectivity...

Checking LAN Emulation settings...

! LES 2 has no entry in the local LECS database.

\* LES 2 does not have a BUS address.

! BUS 2 is not used by any local LES.

-----

Summary

-----

There were 2 warnings and one problem.

---

## elan

**Name**

elan — Creates or configures emulated LANs (ELANs).

**Syntax**

```
elan [ id ] [ unique_option | [ option_list ] ]
```

**Description**

The `elan` command is used to create and configure emulated LANs (ELANs). Most forms of the `elan` command require that the `id` be specified, where `id` is the index number of the ELAN for which you want to configure or display information and has a value between 1 and 99. Any parameters not set by the switch administrator take on the defaults listed in the LAN Emulation Over ATM Specification. The LES address must be configured to bind the ELAN to its associated LES.

The `elan show` command displays the list of ELANs.

If you choose to specify a unique option, you must specify only one option. You can specify more than one option from the option list, but the options must be separated by a space. If you specify a unique option, you must not specify any options from the option list.

**Unique Options**

-create   -c	Creates the specified ELAN.
-enable   -en	Enables the ELAN.
-disable   -dis	Disables the ELAN.
-rm	Deletes the ELAN from memory and Flash.

**Options**

-default   -def	Marks ELAN as a default ELAN. In case of a tie, the lowest number wins.
-nodefault   -nodef	Removes ELAN from the list of default ELANs.
-name   -n " <i>string</i> "	Sets the ELAN name to the specified double quoted string that is up to 32 characters long and is in the format of an SNMPv2 display string. The <i>string</i> must be separated from the <code>-name</code> option by at least one space and must be enclosed in double quotes if it contains multiple words.
-noname   -non	Clears the ELAN name.
-type   -t <i>type</i>	Sets the ELAN type; <i>type</i> must be set to 8023 (Ethernet/IEEE 802.3) or 8025 (IEEE 802.5).

elan

-size   -s <i>size</i>	Sets the ELAN maximum frame size. The maximum size frame that all members of the ELAN must be capable of receiving; <i>size</i> must be 1516, 4544, 9234, or 18190.
-psdelay   -ps <i>time</i>	Sets the ELAN path switching delay in seconds; <i>time</i> must be 1-8.
-nopsdelay   -nops	Clears the ELAN path switching delay in seconds.
-retry   -r <i>count</i>	Sets the ELAN maximum retry count. Maximum number of LE ARP retries for a given destination; <i>count</i> must be 0-2.
-noretry   -nor	Clears the ELAN maximum retry count.
-segid   -sg <i>id</i>	Sets the local segment ID (for emulated 802.5 LANs).
-nosegid   -nosg	Clears the local segment ID (for emulated 802.5 LANs).
-maxuframe   -unk <i>count</i>	Sets the ELAN maximum unknown frame count. The maximum number of frames for a given unicast MAC address or route descriptor that can be sent to the BUS within the maximum unknown frame time; <i>count</i> must be 1-10.
-nomaxuframe   -nunk	Clears the ELAN maximum unknown frame count.
-ctrltimeout   -tc <i>time</i>	Sets the ELAN control timeout in seconds. The timeout used during LEC initialization; <i>time</i> must be 10-300.
-noctrltimeout   -notc	Clears the ELAN control timeout.
-maxuframetime   -tu <i>time</i>	Sets the ELAN maximum unknown frame time in seconds. (See the <code>-maxuframe</code> option for maximum unknown frame count.) This value must be 1-60.
-nomaxuframetime   -notu	Clears the ELAN maximum unknown frame time.
-vcctimeout   -tv <i>time</i>	Sets the ELAN VCC timeout period in minutes. The inactivity timeout for terminating unused Data Direct VCCs; <i>time</i> must be 1-4294967295.
-novcctimeout   -notv	Clears the ELAN VCC timeout period.
-agingtime   -ta <i>time</i>	Sets the ELAN ARP aging time in seconds. The age time for LE ARP cache entries which have not been reverified; <i>time</i> must be 10-300.
-noagingtime   -nota	Clears the ELAN ARP aging time.
-fwddelay   -td <i>time</i>	Sets the ELAN forward delay time in seconds. The age time when the Topology change state is true for non-local LE ARP cache entries which have not been reverified; <i>time</i> must be 4-15.
-nofwddelay   -notd	Clears the ELAN forward delay time.
-connection   -tp <i>time</i>	Sets the ELAN connection complete time in seconds; <i>time</i> must be 1-10.
-noconnection   -notp	Clears the ELAN connection complete time
-arpwait   -tr <i>time</i>	Sets the ELAN expected LE ARP response time in seconds. The time to wait for an LE ARP response to be returned to a LEC before it retransmits the LE ARP request; <i>time</i> must be 1-30.
-noarpwait   -notr	Clears the ELAN expected LE ARP response time.
-flushtimeout   -tf <i>time</i>	Sets the ELAN flush timeout in seconds. The time to wait for a response to a Flush request; <i>time</i> must be 1-4. LECs should wait at least this long before taking any recovery action.

## elan

<code>-noflushtimeout   -notf</code>	Clears the ELAN flush timeout.
<code>-sendvcc   -mt <i>type</i></code>	Sets the ELAN Multicast Send VCC type. The signaling parameter used when establishing the Multicast Send VCC; <i>type</i> must be ABR (available bit rate) or CBR (constant bit rate).
<code>-nosendvcc   -nomt</code>	Clears the ELAN Multicast Send VCC type.
<code>-sendvccpeak   -mp <i>rate</i></code>	Set the ELAN Multicast Send VCC peak rate in cells/second. The signaling parameter used when establishing the Multicast Send VCC; <i>rate</i> must be 1-16777215. This forward and backward peak cell rate is used only if the <code>-sendvcc</code> option is specified as VBR or CBR.
<code>-nosendvccpeak   -nomp</code>	Clears the ELAN Multicast Send VCC peak rate.
<code>-sendvccavg   -ma <i>rate</i></code>	Sets the ELAN Multicast Send VCC average rate in cells/second. The signaling parameter used when establishing the Multicast Send VCC; <i>rate</i> must be 1-16777215. This forward and backward sustained cell rate is used only if the <code>mt</code> option is specified as VBR.
<code>-nosendvccavg   -noma</code>	Clears the ELAN Multicast Send VCC average rate in cells/second. The signaling parameter used when establishing the Multicast Send VCC; <i>rate</i> must be 1-16777215. This forward and backward sustained cell rate is used only if the <code>-sendvcc</code> option is specified as VBR.
<code>-les <i>addr</i></code>	Sends the LES address; <i>addr</i> is in the format:  <i>aacccdddddssssssssseeeeeeeel</i>  where <i>aa</i> is the AFI, <i>ccc</i> is the DCC, <i>ddddddd</i> is the DSP, <i>sssssssss</i> is the switch ID, <i>eeeeeeeeee</i> is the ESI, and <i>l</i> is the selector; all in hexadecimal format.  The system manager can indicate that the address should be a local LES address. In this case, the syntax for <i>addr</i> is <code>lesnn</code> where <i>nn</i> is the identifier for the LES.

### Examples

To create an ELAN with the index number of 4, enter:

```
elan -create 4
```

To display information about ELAN 3, enter:

```
elan 3
```



---

## elanall

### Name

elanall — Configures or creates emulated LANs (ELANs).

### Syntax

```
elanall id [ unique_option ]
```

```
elanall id [ option_list ]
```

### Description

The `elanall` command is used to create and configure emulated LANs (ELANs). The `elanall id -create` option causes the specified ELAN and its associated BUS and LES to be created. Most forms of the `elan` command require that the *id* be specified, where *id* is the index number of the ELAN for which you want to configure or display information and has a value between 1 and 255. Any parameters not set by the switch administrator take on the defaults listed in the LAN Emulation Over ATM Specification.

All `elanall` commands must contain the index number of the ELAN.

If you choose to specify a unique option, you must specify only one option. You can specify more than one option from the option list, but the options must be separated by a space. If you specify a unique option, you must not specify any options from the option list.

### Unique Options

-create   -c	Creates the specified ELAN, where number identifies the ELAN and has a value of 1-255.
-enable   -en	Enables the ELAN and its associated BUS and LES.
-disable   -dis	Disables the ELAN and its associated BUS and LES.
-rm	Deletes the ELAN and its associated BUS and LES from memory and nonvolatile storage.
-show	Displays the states of the ELAN, BUS, and LES along with the addresses of the LES and BUS.

## elanall

### Options

- name** | **-n** "*string*" Sets the ELAN name to the specified double quoted string that is up to 32 characters long and is in the format of an SNMPv2 display string. The *string* must be separated from the **-name** option by at least one space and must be enclosed in double quotes if it contains multiple words.
- noname** | **-non**  
**-type** | **-t** *type* Clears the ELAN name.  
Sets the ELAN type in the ELAN, BUS, and LES; *type* must be set to 8023 (Ethernet/IEEE 802.3) or 8025 (IEEE 802.5).
- size** | **-s** *size* Sets the maximum ELAN frame size in the ELAN, BUS, and LES. The maximum size frame that all members of the ELAN must be capable of receiving; *size* must be 1516, 4544, 9234, or 18190.
- descript** | **-des**  
"*string*" Sets the description for the LES and BUS to the double quoted string. The *string* must be separated from the **-descript** option by at least one space and must be enclosed in double quotes if it contains multiple words.
- nodescript** | **-nodes** Clears the description for the LES and BUS.

### Examples

To create an ELAN that is associated with a BUS and a LES and has an index number of 4, enter:

```
elanall -create 4
```

To display information about ELAN 3, enter:

```
elanall 3
```

---

## errorlog

### Name

errorlog — Displays or clears error log entries.

### Syntax

```
errorlog [ -slot 1 ] unique_option
```

### Description

The `errorlog` command lets you display or clear error log entries. If you do not specify any switches or options, the command lists the usage summary.

### Unique Options

-all	Displays all the error log entries.
-last <i>number</i>	Displays the specified number of most recent error log entries.
-clear	Clears the errorlog of all entries.

### Examples

To display the last five error log entries for the switch, enter:

```
errorlog -slot 1 -last 5
```

To clear all error log entries for the switch, enter:

```
errorlog -slot 1 -clear
```

h

---

## h

### **Name**

h — Displays the shell history.

### **Syntax**

h

### **Description**

The h command displays the shell history list of the last 20 commands that were entered.

### **Examples**

To display the shell history, enter:

h

---

## help

**Name**

help — Displays help on selected topics.

**Syntax**

help [ *option* ]

**Description**

The `help` command displays a list of available commands and a short description of each one. If you do not specify any options, the command lists the usage summary.

**Option**

`-net`                    Displays the network help information.

**Examples**

To display the network help information, enter:

```
help -net
```

host

---

## host

### Name

host — Processes Internet address and host associations.

### Syntax

```
host [ unique_option ]
```

### Description

The `host` command displays and sets host address/name associations in the system host table. If you do not specify any options, the command lists the usage summary.

### Unique Options

<code>-add <i>name</i> -addr <i>address</i></code>	Associates an Internet address with the host name.
<code>-show</code>	Displays the current system name and Internet address associations.

### Examples

To display the hosts and their IP addresses, enter:

```
host -show
```

To associate the host name `fox` with IP address `16.22.33.192`, enter:

```
host -add fox -addr 16.22.33.192
```

---

## iam

**Name**

iam — Sets the user ID and password.

**Syntax**

```
iam [ option ]
```

**Description**

The `iam` command sets the user ID and password for the system. If you do not specify any options, the command lists the usage summary.

**Options**

`-uid userid -password | -pwd password`      Sets the user ID and password.

**Examples**

To set the user ID to Mark and the password to xyz, enter:

```
iam -uid mark -pwd xyz
```

---

## ifconfig

### Name

`ifconfig` — Configures or displays network interface parameters.

### Syntax

```
ifconfig [ unique_options ]
```

```
ifconfig -interface interface_id [ -address addr [ -  
destaddress addr ] ] [ interface_option ]
```

### Description

The `ifconfig` command is used to redefine the address of an interface or to set other operating parameters.

Note that if you want to redefine the interface address or the netmask, you must reboot for the changes to take effect.

The interfaces are specified as: `snx` (Ethernet), `atx` (ATM), and `lox` (loopback) where `x` specifies the ID of the interface. For example, `sn0`, `at0`.

---

### NOTE

Check the hardware documentation for a listing of the available IDs for each interface.

---

If you do not specify an address or any optional parameters, the `ifconfig` command displays the current configuration for the specified network interface only.

The `-destaddress` option specifies the address of the correspondent on the remote end of a point-to-point link.

### Unique Options

`-show`                                      Displays all configured interfaces.



## ifconfig

### Address Formats

- address | -addr *address* Specifies the IP address to set the network interface to. Must be entered in dot notation (that is, 127.22.24.48).  
**NOTE:** Changing the Ethernet IP address will cause Telnet connections to be lost.
- destaddress | -dest *address* Specifies the correspondent on the other end of a point-to-point link. Must be entered in dot notation (that is, 127.22.24.48).

### Interface Options

- netmask | -nm *mask* Specifies the amount of the address reserved for subdividing networks into subnetworks. Must be entered in dot notation (that is, 127.22.24.48).
- hostip *address* Specifies the host IP address.
- up Marks an interface as working.
- down Marks an interface as not working.
- broadcast | -b [*address*] Specifies the address to use to represent broadcasts to the network. The default broadcast address is the address with a host part consisting of all 1s (ones).  
**NOTE:** The computation of the host part is dependent on netmask.
- metric | -m *number* Sets the routing metric, or number of hops, for the interface as specified by *number*. The default value is 0 (zero), and indicates that both hosts are on the same network. The routing metric is used by the routed daemon. Higher metrics indicate less favorable routes.

### Examples

To configure the Ethernet network interface address to be 16.23.24.25 and mark it as working, enter:

```
ifconfig -iface sn0 -address 16.23.24.25 -up
```

lecs

---

## lecs

### Name

`lecs` — Controls the LAN Emulation Configuration Server.

### Syntax

```
lecs [ unique_option ]
```

### Description

The `lecs` command controls the LAN Emulation Configuration Server (LECS). The LECS is always present and provides complete ELAN support in the absence of the other LAN Emulation servers. The `lecs` command is used to display all information configured at the LECS. It is used to enable and disable the LECS. You would use the `-disable` option if you have another LECS in the network.

### Unique Options

<code>-show   -s</code>	Displays the LECS status.
<code>-enable   -en</code>	Enables the LECS and causes the LECS to respond to configure requests.
<code>-disable   -dis</code>	Disables the LECS and causes the LECS to stop responding to configure requests.
<code>-counters   -co</code>	Displays the LECS counters.
<code>-clients   -cl</code>	Displays clients connected to the LECS.
<code>-clrflash   -cf</code>	Clears the LECS from Flash.
<code>-redundant   -r</code>	Enables the redundant LECS.

### Examples

To enable the LECS, enter:

```
lecs -enable
```

---

## les

### Name

`les` — Configures or displays LES characteristics.

### Syntax

```
les id [ unique_option | [ option_list ] ]
```

### Description

The `les` command is used to create and configure LAN Emulation Servers (LESs). Most forms of the `les` command require that the *id* be specified, where *id* is the index number of the LES for which you want to configure or display information and has a value between 1 and 99. Any parameters not set by the switch administrator take on the defaults listed in the LAN Emulation Over ATM Specification. The BUS address must be configured to bind the LES to its associated BUS.

The `les show` command displays the list of LESs.

If you choose to specify a unique option, you must specify only one option. You can specify more than one option from the option list, but the options must be separated by a space. If you specify a unique option, you must not specify any options from the option list.

### Unique Options

<code>-create   -c</code>	Creates the specified LES.
<code>-enable   -en</code>	Enables the LES and causes the LES to allow the LECS to join the ELAN.
<code>-disable   -dis</code>	Disables the LES and causes the LES to terminate all LEC ELAN memberships and not allow any new joins.
<code>-rm</code>	Deletes the LES and removes the numbered LES from memory and Flash.
<code>-arpcount   -ca</code>	Displays the LES ARP counters.
<code>-flushcount   -cf</code>	Displays the LES flush counters.
<code>-joincount   -cj</code>	Displays the LES join counters.
<code>-regcount   -cr</code>	Displays the LES register counters.
<code>-topocount   -ct</code>	Displays the LES topology change counters.
<code>-vcccount   -cv</code>	Displays the Control VCC counters.
<code>-clients   -cl</code>	Displays the clients connected to the specified LES.
<code>-map</code>	Displays the client address mappings for the specified LES.

les

## Options

-bus <i>addr</i>	Address of the BUS; <i>addr</i> is in the format: <i>aaccdddddssssssssseeeeeeeel</i> where <i>aa</i> is the AFI, <i>cccc</i> is the DCC, <i>ddddddd</i> is the DSP, <i>ssssssssss</i> is the switch ID, <i>eeeeeeeeee</i> is the ESI, and <i>l</i> is the selector; all in hexadecimal format. The system manager can indicate that the address should be a local BUS address. In this case, the syntax for <i>addr</i> is <i>busnn</i> where <i>nn</i> is the identifier for the BUS.
-name   -n " <i>string</i> "	Sets the ELAN name to the specified double quoted string that is up to 32 characters long and is in the format of an SNMPv2 display string. The <i>string</i> must be separated from the -name option by at least one space and must be enclosed in double quotes if it contains multiple words.
-noname   -non	Clears the ELAN name.
-type   -t <i>type</i>	Sets the ELAN type; <i>type</i> must be 8023 (Ethernet/IEEE 802.3) or 8025 (IEEE 802.5).
-size   -s <i>size</i>	Sets the ELAN maximum frame size. The maximum size frame that all members of the emulated LAN must be capable of receiving; <i>size</i> must be 1516, 4544, 9234, or 18190.
-ctrltimeout   -tc <i>time</i>	Set the ELAN control timeout in seconds. The timeout used during LEC initialization; <i>time</i> must be 10-300.
-topotimeout   -tt <i>time</i>	Sets the ELAN topology change timeout in seconds. The timeout used to force an emulated LAN out of the topology change state when no topology change requests are seen; <i>time</i> must be 10-300.
-descript   -des " <i>string</i> "	Sets the description for the LES to the double quoted string. The <i>string</i> must be separated from the -descript option by at least one space and must be enclosed in double quotes if it contains multiple words.
-nodescript   -nodes	Clears the description for the LES.
-arp	Enables the forwarding of LE-ARP requests to all clients.
-noarp	Disables the forwarding of LE-ARP requests to all clients.
-narp	Enables forwarding of LE-NARP requests.
-nonarp	Disables forwarding of LE-NARP requests.

## Examples

To create a LES with the index number of 1, enter:

```
les 1 -create
```

To set the name of LES 1 to les\_1 and set its max frame size to 4544, enter:

```
les 1 -name "les_1" -size 4544
```

---

## link

**Name**

link — Configures a link.

**Syntax**

```
link [ option ]
```

**Description**

The `link` command lets you configure a link. If you do not specify any options, the command lists the usage summary.

**Options**

`-maxlvci | -m 1:port:vc` Specifies the maximum VCI for a link.

**Examples**

To specify a maximum VCI of 300 for port 3 enter:

```
link -maxlvci 1:3:300
```

---

## loadconfig

### Name

`loadconfig` — Directs the downloading or uploading of both the kernel and application images.

### Syntax

```
loadconfig [ option ]
```

### Description

The `loadconfig` command is used to specify the location of the switch, host, and load image file. It then directs the downloading of that image from the host to the switch. If you do not specify any options, the command lists the usage summary.

### Options

<code>-conf   -c</code>	<code>-swip <i>address</i></code>	Specifies the switch IP address, switch netmask, host IP address, and the file name of the default configuration to be downloaded using the <code>-download</code> option of the <code>loadconfig</code> command. The <code>-swip</code> and <code>-netmask</code> options are a pair and must appear together on the command line. The <code>-conf</code> option must appear first, followed by the other options in any order.
<code>-netmask   -nm</code>	<code><i>mask</i></code>	
<code>-hostip <i>address</i></code>	<code>-file <i>filename</i></code>	
<code>-download   -d</code>		Downloads the default configuration.

### Examples

To specify that a new image be downloaded from a specified host to the specified switch, enter:

```
loadconfig -c -swip 16.22.13.92 -nm ff000000 -hostip  
16.22.13.99 -file image1
```

To download the new image to the switch, enter:

```
loadconfig -d
```

---

## netstat

### Name

netstat — Shows network status.

### Syntax

```
netstat [ unique_options ]
```

```
netstat -protocol protocol
```

```
netstat -interface interface [ -n | -t ]
```

```
netstat options
```

### Description

The `netstat` command displays the network status of active sockets (default display), a specific interface (interface display), or a specific protocol (protocol display). If you do not specify any options, the command lists the usage summary. If you specify the `-protocol` and `-interface` parameters, you must specify the options for these parameters first.

#### Default Display

The default display lists active sockets for each protocol. The `netstat -showall` command shows the default display and lists the following items:

- PCB address
- Local and remote addresses
- Send and receive queue sizes (in bytes)
- Protocol
- State

Address formats are of the form `host.port` by default or `network.port` when the socket address specifies a network but no specific host address. The host and network address are displayed symbolically unless `-n` is specified.

#### Interface Display

The network interface display format provides a table of cumulative statistics for the following:

- Current flag settings
- Configured Interface and broadcast addresses

## netstat

- Current netmask
- Ethernet address of switch
- Maximum Transmission Unit (MTU) size
- Network address
- Packets received and sent
- Packets received or sent in error
- Collisions (Note that collisions has different meanings for different network interfaces.)

### Protocol Display

The protocol display format provides a table of cumulative protocol-specific statistics.

The TCP protocol display format provides a table of cumulative statistics for the following:

- Packets received and sent
- Connection requests and accepts
- Connections established and closed
- Retransmit, persist, and keepalive timeouts

The UDP protocol display format provides a table of cumulative statistics for the following:

- Input and output packets
- Errors, including incomplete header, bad data length field, bad checksum, broadcasts received without ports, and full socket

The IP protocol display format provides a table of cumulative statistics for errors, including badsum, tooshort, toosmall, badhlen, badlen, fragments, fragments dropped or timed out, forwards, can't forward, and redirects sent.

### Unique Options

-showall | -s                      Displays a list of active sockets for each protocol.



## netstat

### Flags

<code>-interface   -iface interface</code>	Displays the state of the specified interface. With the <code>-s</code> option; displays the DNA Data Link Layer counters for the network interface and the adapter's status and characteristics.
<code>-protocol   -p protocol</code>	Displays statistics for the specified protocol. Supported protocols are UDP, TCP, and IP.

### Options

<code>-arp</code>	Displays the current ARP table.
<code>-memory   -m</code>	Displays information about memory allocated to data structures associated with network operations.
<code>-routing   -r</code>	Displays the host's routing tables. When used with the <code>-s</code> option, it shows the host's routing statistics instead of the routing tables.

### Examples

To display a list of active sockets for each protocol, enter:

```
netstat -showall
```

To display the statistics for the TCP protocol, enter:

```
netstat -protocol tcp
```

To display the state of the Ethernet interface, enter:

```
netstat -interface sn
```

params

---

## params

### Name

params — Saves or restores a parameter file to or from a host system.

### Syntax

```
params [ unique_option ]
```

### Description

The `params` command saves or restores a parameter file to or from a host system. The host's IP address, port, and file name must be specified. If you do not specify any options, the command lists the usage summary.

### Unique Options

<code>-restore   -r -ip <i>addr</i> -file <i>name</i> -port <i>y</i> -overwrite</code>	Retrieves the parameter file previously saved at the host using the <code>params</code> command with the <code>-save</code> parameter and programs them into the Flash if they are set up manually.
<code>  -nooverwrite</code>	
<code>-save   -s -ip <i>addr</i> -file <i>name</i> -port <i>y</i></code>	Transfers all the parameters currently stored in the Flash as a file to the specified host and port.

### Examples

To save all parameters from Flash to the host 16.22.23.192 as file `saved_params`, enter:

```
params -save -ip 16.22.23.192 -file saved_params -port 0
```

---

## passwd

**Name**

passwd — Sets the user or master password.

**Syntax**

passwd [ *option* ]

**Description**

The `passwd` command sets the user or master password for a switch. If you do not specify any options, the command lists the usage summary.

**Options**

-user   -u	Changes the user password.
-master   -m	Changes the master password.

**Examples**

To change the user password, enter:

```
passwd -user
```

password\_disable

---

## **password\_disable**

### **Name**

`password_disable` — Disable password protection for user and manager accounts.

### **Syntax**

`password_disable`

### **Description**

Use this command to disable password protection for both user and manager accounts. The command can be executed from either the console port or a Telnet connection but will be used only to control Telnet access. The factory default is `password_disabled`.

### **Example**

To disable the password protection , enter:

```
password_disable
```

---

## password\_enable

### Name

password\_enable — Enable password protection for user and manager accounts.

### Syntax

```
password_enable
```

### Description

Use this command to enable password protection for both the user and manager accounts. The command can be executed from either the console port or a Telnet connection but will be used only to control Telnet access. The factory default is password\_disabled.

If the protection is enabled, the Telnet login process prompts you for the account login name and password. The account login name must be either `user` or `manager`, and these login names cannot be changed. The default password is `username` for the user account and `managername` for the manager account.

### Example

The following example shows the Telnet login process using the default values for the user account:

```
ATMswitch login: user

Password: username

ATMswitch->
```

plcp

---

## plcp

### Name

plcp — Sets the E1 line type.

### Syntax

```
plcp [ unique_option ]
```

### Description

The `plcp` command sets the E1 line type. If you do not specify any options, the command lists the usage summary.

### Unique Options

<code>-slot 1</code>	Displays the E1 line types for all ports in the switch.
<code>-on   -off -port 1:port</code>	Sets PLCP for the specified port.

### Examples

To force PLCP for port 3 enter:

```
plcp -on -port 1:3
```

---

## port

**Name**

`port` — Configures a port.

**Syntax**

`port` [ *unique\_option* ]

**Description**

The `port` command is used to configure and display port characteristics. You can enable or disable a port, set or display its transmit clock timing, set or display its OAM or Physical layer framing support, or display its SUNI or cell counters. If you do not specify any options, the command lists the usage summary.

**Unique Options**

<code>-disable</code>   <code>-dis</code> <code>-port 1:port</code>	Disables the specified port.
<code>-enable</code>   <code>-en</code> <code>-port 1:port</code>	Enables the specified port.
<code>-suni</code> <code>-port 1:port</code>	Displays the SUNI counters for the specified port.
<code>-cell</code> <code>-port 1:port</code>	Displays the cell counters for the specified port.
<code>-transclock_timing</code>   <code>-tct</code> <code>-port 1:port</code>	Displays or sets the transmit clock timing for the specified port.
<code>-oam</code>	Displays the state of OAM support.
<code>-oam</code> <code>-port 1:port</code>	Sets up OAM support for the specified port.
<code>-plframing</code>   <code>-plf</code> <code>-slot 1</code>	Displays the Physical layer framing type for the specified slot.
<code>-plframing</code>   <code>-plf</code> <code>-show</code>	Displays the Physical layer framing type for the current slot.
<code>-plframing</code>   <code>-plf</code> <code>-sdh</code>   <code>-sonet</code> <code>-port 1:port</code>	Modifies the Physical layer framing support for the specified port.

**Examples**

To enable port 3, enter:

```
port -enable -port 1:3
```

To set up SONET support for port 2, enter:

```
port -plf -sonet -port 1:2
```

---

## pvc

**Name**

`pvc` — Allows configuration of the PVCs.

**Syntax**

`pvc [ unique_option ]`

**Description**

The `pvc` command allows configuration of the permanent virtual circuits (PVCs). It is used to configure and manipulate point-to-point and point-to-multipoint Available Bit Rate (ABR) and Constant Bit Rate (CBR) PVCs. PVCs can be created, deleted, displayed (by port, or by switch), enabled, and disabled. You can display the allocation range, current usage, circuit mapping, in state, and out state of PVCs. If you do not specify any options, the command lists the usage summary.

Note that circuit setup commands must be entered in the `pvc -circuit_type -links -params` format. Optional parameters must follow required parameters and can be entered in any order. When you create point-to-multipoint ABR PVCs, you must specify the root link and flow control. When you create point-to-multipoint CBR PVCs, you must specify the root link, the bandwidth (in megabits per second or cells per second), and the the branch VCI. When creating branch ABR or CBR PVCs, you must specify a previously created root link and a branch link (where the branch ABR link can specify flow control).

**Unique Options**

<code>-flash   -f</code>	Displays the PVCs contained in nonvolatile memory (Flash).
<code>-all   -a [ -internal   -i ]</code>	Displays all active VCs for the switch; include internal PVCs with the optional <code>-internal</code> flag.
<code>-all   -a [ -internal   -i ] -slot 1</code>	Displays all active VCs for the specified slot; include internal PVCs with the optional <code>-internal</code> flag.
<code>-all   -a [ -internal   -i ] -port 1:port</code>	Displays all active VCs for the specified port; include internal PVCs with the optional <code>-internal</code> flag.
<code>-range   -r -port 1:port</code>	Displays the VCI allocation ranges for the specified port.
<code>-usage   -u -link 1:port:vci</code>	Displays the current usage for the specified VC.
<code>-map   -m -port 1:port</code>	Displays the circuit mapping parameters for the specified port.
<code>-state   -st -link 1:port:vci</code>	Displays the in and out state for both ends of the circuit.
<code>-enable   -en -link 1:port:vci</code>	Enables the specified virtual circuit.
<code>-disable   -dis -link 1:port:vci</code>	Disables the specified virtual circuit.
<code>-rm   -d -pp   -root -link_1 1:port:vci</code>	Deletes point-to-point or root addresses from the VC address mapping.
<code>-rm   -d -branch [-link_1 1:port:vci] -link_2 1:port:vci</code>	Deletes branch addresses from the VC address mapping.



pvc

<code>-abr -link_1 1:port:vci -link_2 1:port:vci [-nodup] [-fcin] [-fcout]</code>	Sets up an ABR point-to-point circuit and saves it to Flash. The default state for this circuit is duplex (use the <code>-nodup</code> option to change this state). The <code>-fcin</code> and <code>-fcout</code> options are used to specify that FLOWmaster flow control be used for your input and output ports.
<code>-cbr -link_1 1:port:vci -link_2 1:port:vci -bw mbs   -bwcps cps -rbw mbs   -rbwcps cps [-nodup]</code>	Sets up a CBR point-to-point circuit and saves it to Flash; the bandwidth ( <i>mbs</i> ) must be supplied and must not be 0. (0 specifies an ABR circuit.) The total CBR bandwidth for an OC-3 link is 126 Mb/s. The maximum bandwidth for an OC-12 link is 400 Mb/s.
<code>-inputabr -link_1 1:port:vci [-fcin]</code>	Sets up an ABR point-to-multipoint circuit and saves it to Flash. The <code>-fcin</code> option is used to specify that FLOWmaster flow control be used for your input port.
<code>-inputcbr -link_1 1:port:vci -bw mbs   -bwcps cps -bvci vci</code>	Sets up a CBR point-to-multipoint circuit and saves it to Flash. The bandwidth ( <i>mbs</i> ) must be supplied and must not be 0. (0 specifies an ABR circuit.) The total CBR bandwidth for an OC-3 link is 126 Mb/s. The maximum bandwidth for an OC-12 link is 400 Mb/s. Specifying the branch VCI ( <code>-bvci</code> ) as 0 will direct the software to choose the branch VCI.
<code>-outputabr -link_1 1:port:vci -link_2 1:port:vci [-fcout]</code>	Adds an ABR point-to-multipoint branch circuit and saves it to Flash. The <code>-fcout</code> option is used to specify that FLOWmaster flow control be used for your output port.
<code>-outputcbr -link_1 1:port:vci -link_2 1:port:vci</code>	Adds a CBR point-to-multipoint branch circuit and saves it to Flash.

## Examples

To create a half duplex ABR point-to-point circuit from port 3, VCI 300 to port 2, VCI 400 and save it to Flash memory, enter:

```
pvc -abr -link_1 1:3:300 -link_2 1:2:400 -nodup
```

To set up a full duplex CBR point-to-point circuit with 20 Mb/s bandwidth and 20 Mb/s bandwidth in the reverse direction with VCI 0 (the VCI will be set internally) and save it to Flash memory, enter:

```
pvc -cbr -link_1 1:3:0 -link_2 1:3:0 -bw 20 -rbw 20
```

To set up a full duplex CBR point-to-multipoint circuit with 20 Mb/s bandwidth port 1, with input VCI 250 and save it to Flash memory, enter:

```
pvc -inputcbr -link_1 1:1:250 -bw 20 -bvci 500
```

To add a branch to the circuit in the preceding example, enter:

```
pvc -outputcbr -link_1 1:1:250 -link_2 1:1:500
```

pvc

Note that whenever a CBR or ABR branch is added, the input VCI must be specified completely to correctly associate the branch. If a CBR branch is added, the branch VCI must correspond to the one chosen during input creation (or it must be specified as zero).

To delete the ABR point-to-point circuit created in the first example from the VC address mapping, enter:

```
pvc -d -pp -link_1 1:3:300
```

To delete the CBR branch in the fourth example 4 from the VC address mapping, enter:

```
pvc -d -branch -link_1 1:1:250 -link_2 1:1:500
```

---

## restart\_factory\_defaults

### Name

`restart_factory_defaults` — Reset the module with its original default settings.

### Syntax

`restart_factory_defaults`

### Description

When you execute this command, password protection will be disabled, previously stored passwords will be deleted, and the module will reboot. If the module is rebooted again, it will come up with all of the user-created settings except password enable and the user-defined passwords. When the module comes up with default settings, any configurable parameter may be modified. Any parameter that has been modified will then show the changed parameter value upon subsequent resets.

### Example

To reset to the original default settings, enter:

```
restart_factory_defaults
```

---

## rlecs

**Name**

rlecs - Configures or displays redundant LECS parameters

**Syntax**

```
rlecs [ unique_option ]
```

**Description**

The rlecs command controls the redundant LAN Emulation Configuration Server (RLECS). The RLECS is used as a standby server in case the master LECS goes down. The rlecs command is used to display and configure the rLECS parameters. The RLECS is enabled using the lecs -redundant command.

**Unique Options**

-params   -p	Displays configuration information.
-rflash   -rf	Displays configuration information stored in Flash.
-wrflash   -wf	Writes the RLECS configuration information to Flash.
-stats   -s	Displays the RLECS statistics.
-master   -m	Sets the master selection mode to master.
-standby   -sb	Sets the master selection mode to standby.
-default   -d	Sets the master selection mode to default.
-restore   -r	Enables the restoration option. This option lets a master LECS that has gone down and that has been replaced by a slave LECS resume its role as the master LECS once it becomes operational again.
-norestore   -nr	Disables the restoration option.
-remote   -ra <i>addr</i>	Sets the remote RLECS address; <i>addr</i> is in the format:  aaccdddddssssssssssseeeeeeeeeee1 where aa is the AFI, cccc is the DCC, dddddddd is the DSP, ssssssssssss is the switch ID, eeeeeeeeeee is the ESI, and 1 is the selector; all in hexadecimal format.
-hello   -hi <i>time</i>	Specifies the Hello Timer interval in seconds; <i>time</i> must be 1-10 and less than the Timeout Timer interval value.
-timeout   -tout <i>time</i>	Sets the Timeout Timer value in seconds; <i>time</i> must be 1-20 and greater than the Hello Timer interval.
-deadfactor   -df <i>value</i>	Specifies the number of timeouts to wait for a response from the RLECS before determining that the RLECS is not available; <i>value</i> must be 1-100.
-cons   -c <i>value</i>	Specifies the number of connection attempts to the RLECS before deciding that the RLECS is not available; <i>value</i> must be 1-100.

---

## script

**Name**

`script` — Specifies remote command file for execution by the switch.

**Syntax**

```
script ("username", "user-password", "host-address",  
"host-directory", "filename")
```

**Description**

The `script` command is used to specify that a remote command file be executed on the switch. For example, this command can be used to facilitate VC definition at startup. If you do not specify any options, the command lists the usage summary.

---

**NOTE**

The host address can be specified as an actual IP address or as 0. If 0 is specified, the file is loaded from the BootP server.

---

---

## service

### Name

`service` — Configures ATM Forum Service Registry table.

### Syntax

```
service service-type -create index [ option ]
```

```
service service-type -delete index [ options ]
```

### Description

The `service` command provides a way to control the services that are advertised to the switch's UNI ports using the ATM Forum Service Registry MIB. (In the context of this command, a service is any service reachable at an ATM address, whether or not it is implemented by the switch.) The `service -show` command displays the contents of the service registry. If you do not specify any options, the command lists the usage summary.

In the registry, each service is identified by the combination of its service type (an object identifier like that used with the `snmp` command) and an index. For convenience, you can type in `lecs` and `ans` for the LAN Emulation Configuration Server (LECS) and ATM Name Server (ANS) entries.

You can control the services that are available to specific ports by using the `-port` qualifier. If you create an entry using `-port`, the service is advertised only to the listed ports.

If you modify an entry using `-create ... -port`, the switch starts advertising the service to the listed ports. If you modify an entry using `-delete ... -port`, the switch stops advertising the service to the listed ports.

### Options

<code>-address <i>addr</i></code>	Sets or changes the ATM address of the specified entry. This option only applies to the <code>-create</code> version of the <code>service</code> command.
<code>-port 1:<i>port</i></code>	Specifies the port for which a <code>-create</code> or <code>-delete</code> option applies. You can use several <code>-port</code> qualifiers as part of a single command.

### Limitations

The `service` command does not let you configure the value of `atmfSrvcRegParm1`, a MIB object that is not used when advertising LECS or ANS services.

Deleting a service from the registry does not keep clients who know the ATM address from calling the server.

service

### **Examples**

To create an entry for a local LECS, enter:

```
service lecs -create 1 -address local
```

To create an entry for a LECS running on a workstation, enter:

```
service lecs -create 2 -address  
399999000000008002bb51200.08002b123456.00
```

set\_gateway\_IP\_address("gateway-ip-addr")

---

## **set\_gateway\_IP\_address(" *gateway-ip-addr*")**

### **Name**

set\_gateway\_IP\_address(" *gateway-ip-addr*") — Set a gateway IP address.

### **Syntax**

set\_gateway\_IP\_address(" *gateway-ip-addr*")

### **Description**

Use this command when the IP address will not be set through BootP. Enter an IP gateway address enclosed in quotation marks as shown below.

ATMswitch-> **set\_gateway\_IP\_address(" *ip\_gateway\_addr*")**

### **Example**

To set a gateway IP address, enter:

**set\_gateway\_IP\_address("16.22.57")**



shellLock(1)

---

## shellLock(1)

### **Name**

shellLock(1) — Disable Telnet access to the module.

### **Syntax**

shellLock(1)

### **Description**

Use this command to disable Telnet access to the module. This command can only be given from the console port and will not be preserved if you reboot the module.

### **Example**

To disable Telnet access, enter:

```
shellLock(1)
```

shellLock(0)

---

## shellLock(0)

### Name

shellLock(0) — Enable Telnet access without having to reset the module.

### Syntax

```
shellLock(0)
```

### Description

Use this command to enable Telnet access without having to reset the module.

### Example

To enable Telnet access, enter:

```
shellLock(0)
```

---

## show\_port\_status

### Name

`show_port_status` — Check the operation of the Physical layer protocols, circuitry, and cables.

### Syntax

```
show_port_status
```

### Description

Use this command to gather the status indicated by the module LEDs when direct observation is not possible, such as during remote access.

### Example

To show the status and display the results, enter:

```
ATMswitch-> show_port_status
```

The resulting display resembles the following:

Port	PHY	FRU/MODE
1	GREEN	OFF
	working	-
2	OFF	OFF
	no input	-
3	OFF	OFF
	no input	-
4	OFF	OFF
	no input	-
5	OFF	OFF
	no input	-
6	OFF	OFF
	no input	-
7	GREEN	OFF
	working	-
8	OFF	OFF
	no input	-

value = 0 = 0x0

sig

---

## sig

### Name

sig — Configures various signaling characteristics.

### Syntax

```
sig [ signaling_option ] [ options ]
```

### Description

The `sig` command configures various signaling characteristics. If you do not specify any options, the command lists the usage summary.

Note that the command line must specify the signaling option (`-uni`, `-stack`, or `-ncc`) first before specifying any of its options.

### -uni Options

<code>-ver <i>version</i> -port 1:<i>port</i></code>	Configures the UNI version for a link. <b>NOTE:</b> The value for <i>version</i> must be one of the following: 0           autoconfigure mode 30           UNI version 3.0 31           UNI version 3.1 40           UNI version 4.0
<code>-status</code>	Displays protocol status of links.
<code>-dflash   -df</code>	Displays the database stored in Flash.
<code>-wrflash   -wf</code>	Writes the configuration to Flash.
<code>-clrflash   -cf</code>	Clears the configuration in Flash.
<code>-rflash   -rf</code>	Restores the configuration from Flash.

### -stack Options

<code>-qunirec   -qur <i>size</i></code>	Sets the QSAAL receive queue size for a UNI link.
<code>-qunitrans   -qut <i>size</i></code>	Sets the QSAAL transmit queue size for a UNI link.
<code>-qdecnnirec   -qdr <i>size</i></code>	Sets the QSAAL receive queue size for a DEC NNI link.
<code>-qdecnnitrans   -qdt <i>size</i></code>	Sets the QSAAL transmit queue size for a DEC NNI link.
<code>-maxbgns   -mb <i>number</i> - port 1:<i>port</i></code>	Sets the QSAAL maximum number of BGNs to transmit for establishing QSAAL connectivity.
<code>-timerpoll   -tpoll [<i>interval</i>] -port 1:<i>port</i></code>	Sets the QSAAL POLL timer interval (default = 1 sec).
<code>-timernoresp   -tnor [<i>interval</i>] -port 1:<i>port</i></code>	Sets the QSAAL no response timer interval (default = 10 secs).
<code>-timeralive   -talive [<i>interval</i>] -port 1:<i>port</i></code>	Sets the QSAAL keep alive timer interval (default = 2, should be less than or equal to no response timer).

sig

-timercci   -tcci [ <i>interval</i> ] - port 1: <i>port</i>	Sets the QSAAL timer interval between each BGN (default = 10).
-q2931   -q2-statistics   -stats -port 1: <i>port</i>	Displays the receive and transmit statistics for each Q2931 PDU type.
-qsaal   -qs -statistics   -stats -port 1: <i>port</i>	Displays the receive and transmit statistics for each QSAAL PDU type.
-qsaal   -qs -state -slot 1	Displays the state of QSAAL for each port on the specified slot.
-qsaal   -qs -state -show	Displays the state of QSAAL for each port.
-qsaal   -qs -conf   -c -slot 1	Displays the QSAAL timers and traffic information for each port on the specified slot.
-qsaal   -qs -conf   -c -show	Displays the QSAAL timers and traffic information for each port.

### **-ncc Options**

-resources   -r	Displays the list of NCC resources currently available along with their total, used, available, and capacity values.
-svc	Displays the current number of SVCs established since the switch was last booted (this is a running total).
-consperport   -cpp -port 1: <i>port</i>	Displays information pertaining to the INGRESS and EGRESS SVC connections for the specified slot and port.
-connections   -cons	Displays the SVC connection matrix which shows the switch slots and ports that are currently involved in active connections.
-threadinfo   -thr <i>threadid</i>	Displays the information pertaining to the specified NCC INGRESS and EGRESS threads (if <i>threadid</i> = 0, then display information for all threads) ( <i>threadid</i> range: 0 - 10).
-calls   -cs	Displays the list of CALLS currently in use by the switch.
-call   -c <i>callid</i>	Displays the contents of the specified CALL from the NCC CALL database ( <i>callid</i> range: 1 - 2088).
-parties   -ps	Displays the list of PARTIES currently in use by the switch.
-party   -p <i>partyid</i>	Displays the contents of the specified PARTY from the NCC PARTY database ( <i>partyid</i> range: 1 - 320).
-partiesoncall   -pc <i>callid</i>	Displays the list of PARTIES currently associated with the specified CALL ( <i>callid</i> range: 1 - 2088).
-causecode   -cc <i>code</i> - verbose   -v <i>flag</i>	Displays the description of the specified ATM Forum cause code (if <i>causecode</i> = 0 or not supplied, then display a description of all of the ATM Forum cause codes. If <i>verbose</i> flag = 0 or not supplied, then display a terse description, else display a verbose description).

### **Examples**

To set the UNI version to 3.1 for port 2, enter:

sig

```
sig -uni -ver 31 -port 1:2
```

To set the QSAAL receive queue size to 500 for a UNI link, enter:

```
sig -stack -qunirec 500
```

To display the list of CALLS currently in use by the switch, enter:

```
sig -ncc -calls
```

---

## snmp

### Name

snmp — Examines and sets SNMP MIB variables, or configures SNMP agent.

### Syntax

```
snmp [ unique_option ]
```

```
snmp [ option_list ]
```

### Description

The `snmp` command provides a simple way to examine and set the switch's SNMP MIB variables from the console. It also provides a way of setting the community strings that control access to the MIBs, and the destination IP address for traps. If you do not specify any options, the command lists the usage summary.

This command is intended for use by managers who have a basic familiarity with SNMP and low-level MIB operations (Get, Get-Next, Set). It is not intended to replace GUI network management tools such as the clearVISN MultiChassis Manager.

### Unique Options

<code>-get   -g <i>variable</i></code>	Gets the value of the specified object.
<code>-set   -s <i>variable -to value</i> [<i>variable -to value</i>]</code>	Sets object to a new value. You can set multiple objects to new values.
<code>-next   -n <i>variable</i></code>	Gets the value of the next object after the specified one.
<code>-walk   -w [ <i>variable</i> ]</code>	Walks the switch's MIBs, one screen at a time. You can specify a starting point. If you do not, the walk will pick up where the last one left off.
<code>-mibs   -m</code>	Briefly lists the MIBs implemented by the switch, as an aid for explorations using the <code>-walk</code> option.

### Options

<code>-config   -c</code>	Shows SNMP agent configuration (community strings and destination IP address for traps).
<code>-rw <i>community</i></code>	Sets the community string (password) for read-write SNMP access.
<code>-ro <i>community</i></code>	Sets the community string (password) for read-only SNMP access.

## snmp

<code>-ta <i>addr</i></code>	Sets the destination IP address for SNMP traps that the switch generates. Must be entered in dot notation (that is, 127.22.24.48).
<code>-td <i>addr</i></code>	Deletes the destination IP address for SNMP traps that the switch generates. Must be entered in dot notation (that is, 127.22.24.48).
<code>-tc <i>community</i></code>	Sets the community string (password) to use as part of the SNMP traps the switch generates.

You can enter several `-config`, `-rw`, `-ro`, `-ta`, and `-tc` options as part of one command.

### Variable Syntax

Variables can be specified using either symbolic names (“sysObjectID”) or strings of numeric IDs (“1.3.6...”). In addition, strings can be used as indices. For example, `myObject . "FOO"` will be treated as `myObject . 3 . a . b . c`, where a ... c are the ASCII codes for the letters ‘F’, ‘O’, ‘O’.

Hexadecimal strings can be used as indices. For example, `myObject . '08-00-2b-00-00-01'` will be treated as `myObject . 8 . 0 . 43 . 0 . 0 . 1`. Note that each pair of digits is treated as a subidentifier, whether or not you separate digits with dashes. Thus, `myObject . 'ffff'` is treated as `myObject . 255 . 255` and not as `myObject . 65535`.

### Value Syntax

Values can take on any of the following forms:

- Integer.
- String (use double-quotes).
- Hexadecimal string (use single-quotes). This is useful when the variable that you need to set is an ATM address.
- Object identifier (see the Variable Syntax section above).
- RowStatus enumeration name (from RFC 1443).
- TruthValue enumeration name (from RFC 1443).

### Restrictions

Symbolic names for enumeration constants are not currently supported, except in very limited cases, as described in the Value Syntax section.

### Examples

To show the interfaces installed on the switch, enter:

```
snmp -walk ifDescr
```



---

## status

### Name

`status` — Displays comprehensive switch status.

### Syntax

```
status [ unique_option ]
```

### Description

The `status` command displays comprehensive switch status. If you do not specify any options, the command lists the usage summary.

### Unique Options

<code>-summary   -s</code>	Displays a summary of the switch's status.
<code>-link   -l</code>	Displays switch status per link.

### Examples

To display a summary of the switch's status, enter:

```
status -summary
```

---

## switch

### Name

`switch` — Configures the ATM system.

### Syntax

```
switch [ unique_option | [ option_list ] ]
```

### Description

The `switch` command is used to configure and display system-wide resources and services. You can set or display the watchdog timer, reboot the switch software, set or display the virtual path and EFCI modes, display the slot configuration, display the OAM support, and display the Physical layer framing support. If you do not specify any options, the command lists the usage summary.

### Unique Options

<code>-timeout   -t</code>	Displays the system watchdog timeout value.
<code>-timeout   -t <i>time</i></code>	Modifies the system watchdog timeout value.
<code>-reboot   -r</code>	Reboots the switch software.
<code>-efcimode   -efci -show</code>	Displays the explicit forward congestion indication (EFCI) state of the line card.
<code>-efcimode   -efci -set -slot 1</code>	Sets the switch to EFCI mode.
<code>-efcimode   -efci -clear -slot 1</code>	Clears the EFCI state of the switch.

### Options

<code>-slot</code>	Displays slot configuration.
<code>-oam</code>	Displays the OAM support matrix.
<code>-plframing   -plf</code>	Displays the SDH/SONET support matrix.

### Examples

To display the slot configuration and the OAM support matrix, enter:

```
switch -slot -oam
```

To set the system watchdog timer for 1 second, enter:

```
switch -timeout 1
```

---

## task

### Name

`task` — Displays task information.

### Syntax

```
task [ unique_option ]
```

### Description

The `task` command displays information pertinent to all tasks on the system or for a particular task. A particular task can be specified by task ID or task name. The `-showall`, `-name`, and `-id` options display the task's name, entry point, task ID, priority, status, program counter, stack pointer, error count, and delay time. The `-stack` option displays the task's name, entry point, task ID, maximum stack size, current stack size, largest task size, stack margin, and heap memory usage. If you do not specify any options, the command lists the usage summary.

### Unique Options

<code>-showall</code>   <code>-all</code>	Displays the snapshot of current tasks with associated information for each task.
<code>-id</code> <i>taskid</i>	Displays complete information for the specified task (if <i>taskid</i> = 0, then the information for the current task is displayed).
<code>-name</code>   <code>-n</code> <i>taskname</i>	Displays complete information for the specified task.
<code>-stack</code> <i>taskid</i>	Displays stack usage for the specified task (if <i>taskid</i> not supplied, then the stack usage of all the tasks is displayed).

### Examples

To display the complete info for task 80779856, enter:

```
task -id 80779856
```

To display the complete info for the shell task, enter:

```
task -name tShell
```

trace

---

## trace

### Name

trace — Sets up tracing capabilities.

### Syntax

```
trace -ilmi ilmi_option
```

```
trace -snmp snmp_option
```

```
trace -sig signaling_option
```

```
trace -ncc ncc_option
```

### Description

The trace command is a debugging tool meant for troubleshooting interoperability problems. It assumes that you are very familiar with ATM Forum specifications and/or SNMP. If you do not specify any options, the command lists the usage summary.

### ILMI Options

If you do not specify any options, then the command displays the usage summary for ILMI tracing command options.

-on   -off -port 1: <i>port</i>	Enables or disables ILMI tracing for the specified port.
-on   -off -path 1: <i>port</i>	Enables or disables ILMI tracing for the specified path.
-on   -off -all	Enables or disables ILMI tracing for all ports.
-keepalive   -k	Shows keep-alive activity for traced ports.
-nokeepalive   -nok	Does not show keep-alive activity for traced ports.
-variables   -vars	Shows Get-Responses for traced ports.
-novariables   -nov	Does not show Get-Responses for traced ports.
-ticks   -noticks	Does or does not show timestamps in tracing messages.
-show	Displays the ILMI tracing summary.
-help   -h	Displays help on ILMI tracing command options.

### SNMP Options

If you do not specify any options, then the command displays the usage summary for SNMP tracing command options.

-on   -off -port 1: <i>port</i> [-limit <i>value</i> ]	Enables or disables SNMP tracing for the specified port.
-on   -off -ip [-limit <i>value</i> ]	Enables or disables SNMP tracing for this IP.
-on   -off -console [-limit <i>value</i> ]	Enables or disables SNMP tracing for the console.
-on   -off -rmon [-limit <i>value</i> ]	Enables or disables SNMP tracing for the RMON.

## trace

-on | -off -all [-limit *value*] Enables or disables SNMP tracing for all ports.  
-show Displays the SNMP tracing summary.  
-help | -h Displays help on SNMP tracing command options.

### Signaling Options

-level *level* -port 1:*port* Enables tracing of various components of the signaling interface. The most useful and important for network diagnostics are levels 2, 20, 21, 22, 23, and 24. All other levels are for internal tracing. Tracing can be set for either all ports or specific ports. Tracing values consist of:

- 2 - call processing
- 20 - Q2931 on specific port
- 21 - disable trace on specific port
- 22 - QSAAL on specific port
- 23 - enable realtime PDU tracing
- 24 - enable management indications

-level *level* Enables internal tracing. Tracing values consist of:

- 0 - disable tracing
- 1 - link up/down
- 2 - call processing
- 3 - restart recovery
- 4 - SSCOP and Q2931 PDUs
- 5 - SSCOP PDUs
- 6 - Q2931 PDUs
- 7 - SSCOP PDUs on specified port
- 8 - Q2931 PDUs on specified port
- 9 - elaborated link up/down

trace

### NCC Options

<code>-level level -port 1:port</code>	Enables tracing on the specified slot and port at the specified level. Refer to the NCC trace help command for a description of the various tracing levels available. If both the slot and port are specified as zero (0), then tracing is enabled at the specified level for all ports. Tracing levels range from 0-4.
<code>-display   -disp -port 1:port</code>	Displays the tracing level for the specified slot and port. Refer to the NCC trace help command for a description of the various tracing levels available. If both the slot and port are specified as zero (0), then the tracing level for all ports is displayed.
<code>-help   -h</code>	Displays information describing the levels and types of tracing available in the switch. They include:

LEVEL	TYPE	DESCRIPTION
0	disabled	trace disabled
1	level1	trace transmitted or received PDUs
2	level2	decode transmitted or received PDUs
3	level3	detailed trace on failures
4	XLATE PDU	trace UNI 3.0/3.1 translations

### Examples

To enable tracing of QSAAL on port 3, enter:

```
trace -sig -level 22 -port 1:3
```

To display the current tracing level for port 2, enter:

```
trace -ncc -display -port 1:2
```

---

## versions

**Name**

`versions` — Displays the version numbers for the switch software.

**Syntax**

`versions`

**Description**

The `versions` command displays information about the current software present on the switch. It displays the version numbers and dates for the Boot ROM and the application images along with the host IP address and the host download control file name.

**Example**

To display the version number for the Boot ROM and the firmware image, enter:

```
versions
```

whoami

---

## whoami

### **Name**

whoami — Displays the user's name.

### **Syntax**

whoami

### **Description**

The whoami command displays the user's name.

### **Examples**

To display the user's name, enter:

```
whoami
```



# Index

---

## A

- ATM address prefix
  - altering 7-11
  - configuring the default address prefix 7-11
  - managing 7-10

## B

- Box Ethernet IP addresses and subnet masks
  - displaying 2-2
  - setting 2-3
- Box SLIP IP and host IP addresses
  - displaying 2-2

## C

- Cell counters, displaying 2-8
- CLIP
  - configuring 2-4
  - deleting saved parameters 2-5
  - displaying the ARP cache contents 2-7
  - displaying the CLIP configuration status 2-6
  - displaying the IP counters 2-6
  - enabling the CLIP client 2-5
- Commands
  - bus 8-23
  - clearflash 5-7
  - clip 2-4
  - date 5-2
  - decnni 7-9 to 7-10
  - elan 8-6
  - elanall 8-6
  - errorlog 5-5
  - ifconfig 2-2 to 2-3
  - lecs 8-4
  - les 8-14
  - loadconfig 2-9 to 2-10
  - params 5-6
  - passwd 5-3
  - port 2-8 to 2-9, 4-3
  - pvc 3-2 to 3-3, 3-6 to 3-7
  - rlecs 8-28

- script 3-9
- sig 6-2, 6-4, 6-6 to 6-7
- switch 4-2
- versions 2-9

## D

- Default address prefix
  - configuring 7-11
  - deleting 7-11
  - displaying 7-11

## E

- Error logs
  - clearing all logs 5-5
  - displaying error logs 5-4

## F

- Firmware images
  - displaying version numbers 2-9
  - forcing downloads 2-10
  - setting up download information 2-9
- Flash storage
  - erasing 5-7
- Framing support
  - displaying SDH/SONET support 4-5
  - modifying SDH/SONET support 4-5

## L

- LAN Emulation
  - BUS 8-22
  - configuration overview 8-2
  - LECS 8-3
  - LES 8-11
  - redundant LECS 8-27

## M

- Management features
  - LAN emulation 1-3
  - management interfaces 1-3

- command line interface 1-3
- console commands 1-3
- menu interface 1-4
- SNMP management interface 1-4
- routing 1-2

Module configuration, displaying 4-2

## N

Network interface parameters

- configuring 2-3
- displaying configured interfaces 2-2

## P

Passwords

- changing passwords 5-3
- changing the manager password 5-3
- changing the user password 5-3

Port error counters, displaying 2-9

Ports

- disabling 4-3
- enabling 4-4

## PVCs

- creating ABR PVCs 3-6
- creating CBR PVCs 3-7
- deleting 3-8
- disabling 3-8
- displaying active PVCs 3-5
- displaying PVCs 3-3
- displaying PVCs stored in Flash 3-3
- enabling 3-8
- managing 3-2

## Q

Q.2931, displaying link statistics 6-6

Q.SAAL

- displaying link statistics 6-4
- displaying link status 6-5

## R

Routing 1-2

- addressing guidelines 7-6
- altering an existing static route 7-10

- altering the ATM address prefix 7-11
- configuring a static route 7-9
- configuring the default address prefix 7-11
- deleting a static route 7-10
- deleting the default address prefix 7-11
- displaying routing statistics 7-9
- displaying static routes 7-10
- displaying the default address prefix 7-11

## Routing features 7-2

- address autoconfiguration 7-7
- automatic topology discovery 7-2
- configurable address prefixes 7-8
- configurable default prefix 7-7
- exporting static routes 7-5
- IISP gateway functionality 7-5
- IISP support 7-5
- parallel links between switches and VC load sharing 7-3
- shortest path routing 7-2
- static routing 7-3

## S

Scripts, using 3-9

## SNMP

- adding an NMS IP trap address 9-5
- deleting an NMS IP trap address 9-5
- disabling ILMI registration 9-5
- displaying registered addresses 9-6
- displaying the status of all links 9-5
- enabling ILMI registration 9-5
- modifying the read-only community name 9-5
- modifying the read-write community name 9-5
- modifying the trap community name 9-5

## SVC connections

- displaying connection summary 6-7
- displaying link connections 6-7

Switch software, rebooting 4-6

## T

Time, displaying time since initialization 5-2

Transmit timing source

- displaying 4-7

modifying 4-7

## **U**

### **UNI**

- clearing the UNI configuration 6-3
- configuring the UNI version for each port 6-2
- displaying the stored configuration 6-3
- displaying the UNI status 6-2
- restoring the UNI configuration 6-3
- storing the UNI configuration 6-3

