

VX-1000MD Software User Guide



Versa Technology DSLAM Project Group

August 2007

Notice

© 2007 Versa Technology, Inc. All rights reserved.

This manual contains information provided by Versa Technology, Inc., and it is protected by copyright. No part of this publication may be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's personal use, without the prior express written permission of Versa Technology, Inc.

The specification and information in this manual are subject to change without notice. All statements, information and recommendations in this manual are believed to be accurate but are presented without warranty of kind, expressed or implied. Users must take full responsibility for the application of any product.

Versa Technology, Inc. does not warrant that the hardware and software will work properly in all environments and applications, and makes no warranty and representation, either implied or expressed, with respect to the quality, performance, merchantability, or fitness for a particular purpose.

If you find information in this manual that is incorrect, misleading, or incomplete, we would appreciate your comments and suggestions.

Thank you for purchasing Versa's Central Office (CO) Side Access Equipment VX series. The current ADSL technology can't satisfy the needs of the growing market and the number of CO users continues to increase. Henceforth, Versa Technology, Inc. now produces its first generation of ADSL COs, the VX-1000MD based on ATM kernel ADSL technology. This DSLAM system, VX-1000MD IP-DSLAM, is an IP kernel and is based on the latest ITU-T G.992.5 ADSL 2 plus standard.

The main purpose of this user guide is to teach users how to use Versa's VX-1000MD IP-DSLAM software operating system. By learning VX-1000MD configuration examples, users will come to understand the system features and thus be able to perform correct system configurations. Detailed system operating configurations and configuration commands descriptions are included as well.

This user guide also provides detailed information about the VX-1000MD network management and maintenance. In order to understand the operation procedures, the following related knowledge will be demonstrated and discussed:

- LAN
- Ethernet switching and bridging
- Network protocol
- ADSL technology
- IGMP (Internet Group Management Protocol)
- DHCP (Dynamic Host Configuration Protocol)
- VLAN (virtual LAN)
- ATM (Asynchronous Transfer Mode)

If you have any questions during the operation of this system, please contact our technical engineers at Versa Technology, Inc.

Table of Contents

CHAPTER ONE	OVERVIEW	5
1.1	<i>VX-1000MD OVERVIEW.....</i>	5
1.2	<i>SOFTWARE SPECIALTIES</i>	5
1.2.1	<i>ATM.....</i>	5
1.2.2	<i>VLAN.....</i>	6
1.2.3	<i>Management.....</i>	6
1.2.4	<i>Safety specifications.....</i>	6
1.2.5	<i>Uplink and upgrade</i>	6
1.2.6	<i>Maintenance.....</i>	6
CHAPTER TWO	LINE CARD SOFTWARE CONFIGURATION	7
2.1	<i>COMMAND LINE</i>	7
2.1.1	<i>Command Line specialties:.....</i>	7
2.1.2	<i>Command Line Online help</i>	8
2.1.3	<i>Command Line error information</i>	8
2.1.4	<i>Command Line editing.....</i>	9
2.2	<i>TWO WAYS TO VISIT LINE CARD.....</i>	9
2.2.1	<i>Visit Line card by Hyper-Terminal</i>	10
2.2.2	<i>Long distance login through TELNET</i>	12
2.3	<i>UNDERSTANDING COMMAND SYNTAX.....</i>	13
2.3.1	<i>Command input syntax.....</i>	15
2.3.2	<i>Command Alias</i>	15
2.4	<i>CONFIGURATION COMMAND.....</i>	16
2.4.1	<i>System Configuration Commands</i>	16
2.4.2	<i>IP configuration commands</i>	20
2.4.3	<i>Port Configuration Commands</i>	25
2.4.4	<i>Ethernet setup commands.....</i>	30
2.4.5	<i>Setup bridging commands.....</i>	31
2.4.6	<i>rfc1483 configuration commands.....</i>	34

Chapter I Overview

1.1 VX-1000MD Overview

Along with the rapid development of the Internet, broadband technology has quickly become popularized. ADSL broadband technologies have emerged to become the best solution in the last-mile solution market. VX-1000MD is one of the ADSL access products which have been developed by Versa Technology, Inc. It supports the current ADSL, ADSL2 and ADSL2+ technologies. It also provides ADSL access technologies and solutions which have long range, high bandwidth and widely covered area features. The VX-1000MD ADSL relay has 24 ADSL ports and can connect to 24 subscribers instantaneously. It is a flexible network element which makes it also suitable for the rural, low-density subscriber area. As a long range ADSL technology, it supports the following services:

- Internet broadband access
- Video on demand
- Enterprise and campus network
- Countryside area ADSL access

The VX-1000MD is IP core based, using the latest ITU-T G.992.5 ADSL 2plus standard DSLAM system. It has the following features:

- IP access solution that is flexible and easy for network deployment
- Reliable OAM (maintainable, reparable, manageable) functions
- Most reliable design
- Broadband, powerful switching capability

The VX-1000MD ADSL relay provides 24 ADSL ports and instantaneously supports 24 users within a flexible network deployment. It separates the line card and splitter in a standard 1U line card enclosure to support translation between ATM, IP and internal signals. The splitter accomplishes the composition and separation between voice and data signals. The power supply system selectively supports 48V direct current, or 220V alternating current.

1.2 Software Specialties

1.2.1 ATM

- Supports ATM signaling UNI 3.1 & 4.0
- Supports Compliant Traffic management、QoS、Queue and traffic shaping in ATM forum TM4.1
- Supports ATM cell-header translation、VC exchange、VP across connection、ATM cell broadcasting function
- ATM OAM cell processing, congestion/buffer management function support
- Supports multi-protocol encapsulation function in RFC2684 Ethernet based ATM AAL5

- PPP (RFC 2364) over ATM
- RFC 1483
- RFC 1577 (IP over ATM)
- AAL0, AAL2, AAL5
- ATM Service Class: UBR/VBR/CBR
- 32 VCs

1.2.2 VLAN

- Supports VLAN marked IEEE 802.1Q
- Supports port-based VLAN
- Supports isolate-user-VLAN characteristics

1.2.3 Management

- Supports local management through RS232 ports
- Supports command line interface (CLI)
- Supports Telnet long distance deploy
- Supports HTTP based servers and CGI resolved Web browser
- Management support for clients

1.2.4 Safety specifications

- Password protection
- User level control

1.2.5 Uplink and upgrade

- Supports BOOTP Client and BOOT Server upgrade
- Supports WEB upgrades

1.2.6 Maintenance

- Supports debugging information export
- Supports PING

Chapter II Line Card Software Configuration

This chapter introduces Line Card operation and Line Card management using the following command lines.

2.1 Command Line

2.1.1 About CLI

The CLI is the *Command Line Interface* for configuring ISOS modules. It largely replaces the *console commands* that were provided in earlier releases of ISOS. Some console commands are available for use if you have appropriate access permissions set. The CLI provides a command line interface to the ISOS modules that they are modeled in the VMI Information Model.

2.1.2 Using CLI and Console Commands

There are two types of commands available for use in ISOS:

1. CLI commands: For example,

ip show

2. Console commands - Users with appropriate access permissions can enter console mode from the CLI and use the console commands.

There are two types of console commands, and different access permissions exist for each type of command:

- *Usable commands* - Using these commands will not lead to inconsistencies between the Information Model and the underlying system.
- *Blacklisted commands* - using blacklisted console commands can lead to inconsistencies between the Information Model and the underlying system, and should be used **with extreme caution**

Access permissions to the CLI

There are three access level options for CLI users:

- *Default User* - can use CLI commands; cannot use usable console commands or blacklisted console commands.
- *Engineer* - can use CLI commands and usable console commands; cannot use blacklisted console commands.
- *Super User* - can use CLI commands, usable console commands and blacklisted console commands. Can also set up user login accounts, save backup configuration and restore factory settings.

2.1.1 Command Line specialties:

- Local configuration by Console (RS-232) port.
- Local or long distance login configuration by Telnet.

- Command level protection - Different levels of users can only execute current commands authorized to their appropriate levels.
- User can input “?” to get online help at any time.
- Provides FTP service, to simplify uploading and downloading files.
- Provides a function similar to DosKey to execute history commands.
- Command Line interpreter provides multiple intelligent command parse methods such as partial match, context-correlation and etc.

2.1.2 Command Line Online help

The Command Line interface provides online help assistance to users:

To get help information from the above online help, do the following:

In the Command Line prompt, type “?”. All the commands and their descriptions will be listed.

```
> ?
```

Type in one command (position 1) followed with a space and “?”. If position 1 is a command, all keywords and their descriptions will be listed. For example:

```
> ip ?
```

Type in one command (position 1) followed with a space and “?”. If position 1 is a parameter, the related parameter description is listed. If position 1 doesn't have any parameters, then this command is repeated in the very next command line; type Enter to execute.

Type a string followed by “?”. All the commands beginning with the string will be listed. For example:

```
> i?
```

```
igmp imdebug ip
```

Type in one command followed by a string with “?”. All the commands containing the keywords beginning with the string, will be listed. For example:

```
> ip s?
```

```
set show
```

Input the first a few characters of the keyword in the command, then press <tab>. It displays the complete keyword with the assumption that the first few letters are not the beginning part of other commands.

2.1.3 Command Line error information

If all user type commands pass syntax checking, the commands are executed, otherwise an error will be flagged with

information for users. The following list 2-1 contains the most common error information:

English error information jq	Error reason
<i>Unrecognized command</i>	No command found
	No keyword found
	Parameter type error
	Parameter overflow
<i>Insufficiency arguments supplied</i>	No input parameter(s)
<i>Error: specified node not found by path</i>	No target found
<i>Too many arguments supplied</i>	Too many input parameters

List 2-1

2.1.4 Command Line editing

Command Line interface provides basic command edit function.

Key	Attribute
Any normal key	Provides basic command input (case sensitive)
Backspace	Deletes one character ahead
→	Moves the cursor one character after
←	Moves the cursor one character ahead
↑ ↓	Displays history of commands
Delete	Deletes the character on the cursor

List 2-2

2.2 Two ways to visit Line card

There are two ways to visit the Line card:

- 1) Connect the serial port on the Line card to the serial ports on the network computers by a serial port cable. Log in to

the Line card using the Hyper-Terminal program in the computer to manage the Line card.

- 2) Long distance login through TELNET

2.2.1 Visit Line card by Hyper-Terminal

Step One: To set up a local configuration environment, connect a PC or a terminal from its port to the MCU Console port on the DSLAM with a standard RS-232 cable, as shown in the following figure:



Figure 2-1

Step Two: Go to “Start” : “All programs—Accessories—Communication—Hyper-Terminal.” The Hyper-Terminal program will then prompt the following window:



Figure 2-2

After inputting the name, click OK, and the following window will be shown:



Figure 2-3

Connect port setup and select the serial ports to communicate with the MCU (usually choose COM1 or COM2) . Please choose the serial port that your PC uses, then click OK and the following window will be shown (Figure 2-4):

Parameter setup is shown in the following figure:

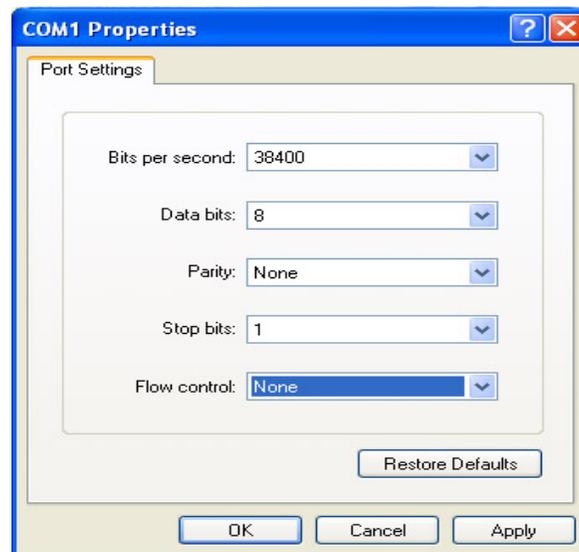


Figure 2-4

Then the screen will display:

```
You must supply a username
Login: _
```

Figure 2-5

Input the user name : admin (default, changeable) , password: admin (default, changeable) , press Enter, and then the "-->" character will be shown, which means that you are in Command Line prompt mode. Then use the Command Line to configure the Line card, as shown in Figure 2-6 below :

```
You must supply a username
Login: admin
Password: *****
Login successful
--> _
```

Figure 2-6



If you login the system using the guest account, then the user name becomes: guest (default, changeable) , password: guest (default, changeable) .

2.2.2 Long distance login through TELNET

Step One: Use IP address path command setup management :

```
> ip set interface iplan ipaddress 192.168.1.32 255.255.255.0
```

```
*/ setupIP address
```

```
> ip add defaultroute gateway 192.168.1.1
```

*/comment : setup path

Step Two: Connect the PC with the system uplink port by internet, so that the PC can successfully ping the system management IP address. See Figure 2-7 below:

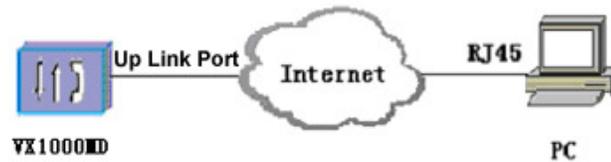


Figure 2-7

Step Three: Open the PC command line prompt, and use the Telnet command to log in to the VX1000MD. For example :

```
C:\telnet 192.168.1.32
```

Likewise with the serial port configuration, type in the login and password correctly, and the Command Line prompt in the line card will be shown.

2.3 Understanding command syntax

2.3.1 CLI Configuration

In order to use the CLI commands, you need to understand the following terms:

- Transport: A transport is a layer 2 session and everything below it. You can create a transport (for example, Ethernet) and attach it to a bridge or router so that data can be bridged or routed via the attached transport.
- Interface: Bridges and routers both have interfaces. A single transport is attached to a bridge or router via an interface.
- Object: An object is anything that you can create and manipulate as a single entity, for example: interfaces, transports, static routes and NAT rules.
- List: Objects are numbered entries in a list. For example, if you have created more than one IP interface, the following command:

```
ip list interfaces
```

produces a list of numbered interface objects. Object numbers are displayed in the first column under the heading ID. For example:

IP Interfaces:

ID	Name	IP Address	DHCP	Transport
1	ipwan	0.0.0.0	disabled	PppoeUp
2	iplan	192.168.1.1	disabled	<BRIDGE>

Attaching a transport to an interface

To attach a transport to a bridge or router, you need to:

1. Create a transport. In the following command, an Ethernet transport is created and named *eth2*, and the port name is specified (*ethernet0*):
`ethernet add transport eth2 ethernet0`
2. Create an interface. In the following command, a bridge interface is created and called *myinterface*:
`bridge add interface myinterface`
3. Attach the transport to the interface. In the following command, the *eth2* Ethernet transport is attached to the *myinterface* bridge interface:
`bridge attach myinterface eth2`

CLI Conventions

The CLI uses standard, intuitive command names that can be used in different instances:

Command	Definition
Add	Use this command to add and name objects (e.g., interfaces or transports). The “add” command requires attributes to be specified as arguments in a certain order. For example, in order to create an Ethernet transport, you need to specify the transport name and system port: <code>ethernet add transport <name> <port></code>
Delete	The “delete” command deletes named objects or numbered objects (displayed using the “list” command): <code>ethernet delete transport {<name> <number>}</code>
Clear	The “clear” command deletes ALL named entities that belong to an object, for example, the following command: <code>firewall clear policies</code> deletes all of the policy objects that belong to the Firewall. You should use the “clear” command with caution - the above example also deletes all validators and port filters that belong to the policies.
Set	The “set” command changes a value or multiple values within the system. For example: <code>ip set interface {<name> <number>} ipaddress <ipaddress></code>

Show	<p>The “show” command lists the current configuration and statistics for an object or module. For example, the command: dhcpserver show subnet {<name> <number>} may give the following output, depending on your DHCP server configuration: Global DHCP Server Configuration: Status: ENABLED Default lease time: 43200 seconds Max. lease time: 86400 seconds Allow BOOTP requests: true Allow unknown clients: true</p>
------	--

2.3.1 Command input syntax

After getting into the Command Line prompt in section 2.2.1, the system will display the “-->” symbol, which means that you can input commands. The steps to input one complete command are shown in the following:

Step One: Input the command into the “-->” prompt; there must be a space between words in the command line.

Step Two : Press “Enter” to execute the command.

【Example】 A command to check the line card IP address:

--> ip list interface (Enter)

Above is an example on how to input a complete command; the result are shown in the following figure below:

```

--> ip l i
IP Interfaces:
  ID | Name      | IP Address | DHCP   | Transport
  ---|---|---|---|---
  1  | iplan    | 192.168.2.2 | disabled | eth1
  ---|---|---|---|---
--> _

```



All commands are case sensitive. Please notice the cases when inputting commands, otherwise errors may occur.

4.3.2 Command Alias

Command alias means you can input a first few letters of the command or keyword, as long as those letters are not the beginning part of more than one command. The line card can recognize the command, and the user can just press Enter to execute the command.

【Example】 To check the IP address of the line card, input the following command:

--> ip list interface

The above command can use alias as :

--> ip li int

Those two commands accomplish the same function.



When using command alias, you must provide enough letters to make sure that only one command contains these letters at the beginning.

2.4 Configuration Commands

2.4.1 System Configuration Commands

By default, an administrator account exists with access privileges set to allow the administrator to create new accounts, save backup configurations, restore factory defaults and configure the device via the CLI, console and EmWeb.

Adding users:

There are two types of accounts that you can add to the system:

- A dialin account that allows the user to access the system via a dialin connection, for example, using PPP. However, this account cannot log in to the system directly. (THIS IS DISABLED)
- A login account that allows the user to log in to the system directly, but not via a dialin connection.

【command action】 Add login user

【command syntax】 system add login admin

【configuration example】 --> system add login admin

Default attributes	Dialin user	Login user
User can dial in to the system via the CLI	Yes	No
User can log in to the system via the CLI	No	Yes
User can access the web pages	No	Yes
Default access permission set	default user	default user

Displaying all users and their rights:

【command action】 Display all users and their rights

【command syntax】 sys list logins

【configuration example】 --> sys list logins

Configuring user rights:

By default, new users are given a *default* access level. There are three access level options for CLI users:

- *Default user* - Can configure the device using CLI commands; cannot enter usable console commands or blacklisted console commands
- *Engineer* - Can configure the device using CLI commands and usable console commands; cannot use blacklisted console commands
- *Super user* - Can configure the device using CLI commands, usable console commands and blacklisted console commands. Can also set up user login accounts, save backup configurations and restore factory settings.

【command action】 Setup login/dialin user rights

【command syntax】 `system set {login/user} <name> access {default/engineer/superuser}`

【configuration example】 --> system set login admin access superuser

To allow/prevent access to the device via the CLI for an existing user, input the following:

`system set {login/user} <name> mayconfigure {enabled/disabled}`

To allow/prevent access to the device via EmWeb for an existing user, input the following:

`system set {login/user} <name> mayconfigureweb {enabled/disabled}`

Configuring user passwords:

【command action】 Setup the password for the admin user

【command syntax】 `system set user <name> password <password>`

【configuration example】 --> system set user admin password admin

Backup configuration:

This command saves the system configuration to a file. To specify the file that you want to save configuration information in, type `//isfs/` or `//flashfs/` (depending on which directory the backup file is stored in), followed by a *filename* value. If you do not specify a filename, the configuration is saved in the `//isfs/im.conf.backup` file by default.

To prevent a user from freely overwriting the system configuration, only a Super User can use this command.

【command action】 Backup the current configuration

【command syntax】 sys config backup [filename]

【configuration example】 --> sys config backup //flashfs/mybackup

system config restore

Syntax

system config restore {backup [filename]|factory|minimal}

Description

This command attempts to restore all system modules; if you do not have all modules installed, the CLI will display a message telling you which modules could not be restored. The following options are available:

- Super users, Engineers and Default users can restore their backup configuration from the *//isfs/im.conf.backup* file.
- Super users can restore their backup configuration from a different file by typing *//isfs/* or *//flashfs/* (depending on which directory the backup file is stored in), followed by a *filename* value.
- Super users can restore the factory defaults from *//isfs/im.conf.factory*.
- Super users can clear their current configuration by restoring the *//isfs/im.conf.minimal*. Attributes are set to their defaults, and all interfaces and transports are deleted. The *default admin user* is not affected, so you can still login to the system.

Options

The following table gives the range of values for each option that can be specified with this command and a default value (if applicable).

Option	Description	Default value
backup	Restores the backup configuration from the <i>//isfs/im.conf.backup</i> file.	N/A
filename	The name of a file containing an alternative backup configuration to that stored in the <i>//isfs/im.conf.backup</i> file. The filename MUST be preceded by either <i>//isfs/</i> or <i>//flashfs/</i> . Only Super Users can use this command.	N/A
factory	Restores the factory default configuration from the <i>//isfs/im.conf.factory</i> file. Only Super Users can use this command.	N/A
minimal	Clears the current configuration by resetting attributes to their defaults and deletes interfaces and transports. Only Super Users can use this command.	N/A

Example:

```
prompt> system config restore backup
```

Deleting configurations:

This command clears your current configuration. Attributes (including BUN ports) are set to their defaults, and all interfaces and transports

are deleted. The *default admin user* is not affected, so you can still log in to the system. This command has the same effect as entering the *system config restore minimal* command.

Once you have entered this command, you need to save your configuration and restart your device using the *system config save* and *system restart* commands. If you do not save and restart, any subsequent changes you make to the configuration will not be saved.

【command action】 delete all configuration

【command syntax】 sys config clear

【configuration example】 --> sys config clear

Display System information

This command displays the vendor ID, URL, base MAC address and hardware and software version details of the system that you are using. This information is important for customer support.

【command action】 Check system software, hardware version and MAC address

【command syntax】 system info

【configuration example】 Check the basic information of line card

--> system info

Display error operation log

This command displays a system error log. The error log contains the following information:

- The time (in minutes) that an error was made, calculated from the start of your session
- The module that was affected by the error
- A brief overview of the error itself

【command action】 Display error operation log

【command syntax】 sys list errors

【configuration example】 -->sys list errors

Saving Configuration

This command saves the system configuration in the *im.conf* file in FlashFS. This allows all users to create their own backup files. Default, Engineer and Super User-level users can use this command.

【command action】 Saves modified parameters, otherwise data will be lost after restarting

【command syntax】 system config save

【configuration example】 --> system config save

Restart system

This command restarts your system. It has the same effect as pressing the reset button on your DSLAM system.

【command action】 Restart line card

【command syntax】 system restart

【configuration example】 --> system restart

2.4.2 IP configuration commands

The IP stack in DSLAM contains a suite of networking routing protocols for use in embedded networking. It allows you to configure basic connectivity for your network to provide IP routing between interfaces and supports local applications such as Telnet, Webserver and so on.

The dual IP Stack implements the following IPv4 protocols:

- Internet Protocol (IP), including RFC 791
- Includes support for Fragmentation and Reassembly (*RFC 0791* and *RFC 1812*)
- Includes support for Subnetting and Classless Interdomain Routing.
- Internet Control Message Protocol (ICMP) (*RFC 0792*)
- User Datagram Protocol (UDP) - *RFC 768*
- Transmission Control Protocol (TCP) - *RFC 793*
- Address Resolution Protocol (ARP) for Ethernet - *RFC 826* and *RFC 894*.
- Internet Group Management Protocol (IGMP), Version 2 - *RFC 2236*
- Routing Information Protocol (RIP), Version 2 - *RFC 1723*;
- Requirements for IP version 4 Routers - *RFC 1812*
- Requirements for Internet Hosts - *RFC 1122*

In addition, the following IPv4 features are provided:

- TCP MSS Clamp
- Multicast forwarding and IGMP Proxy (*RFC 2236*)
- Unnumbered interfaces for point-to-point links
- Virtual interfaces & secondary IP addresses (multi-homing)
- BSD Socket API or message-based interface for third-party applications
- Better performance for VoIP, using the Fast UDP interface

Creating IP interfaces

You must attach one or more interfaces to the IP stack and attach a transport to it. For IPv4 interfaces, each interface must be configured with an IP address and a subnet mask. Together, these define the range of addresses that can be reached via the

interface without passing through any other routers.

Each interface must have a unique subnet; the range of addresses on each interface must not overlap with any other interface. The only exception to this is unnumbered interfaces, which may be configured on point to point links when there is no local subnet associated with that interface.

To create an IPv4 interface, input the following:

```
ip add interface <name> [<ipaddress> [<subnetmask>]]
```

You do not have to specify the IP address and subnet mask of an interface on creation. Address details can be assigned later using DHCP server or PPP.

A transport is attached to an IP interface using the following command:

```
ip attach <name> <transport>
```

For example, first create an IPv4 interface:

```
ip add interface iplan 192.168.1.1 255.255.255.0
```

Second, create a transport. Details of how to create and configure transports are described elsewhere in this guide. In the example below, an Ethernet transport is created:

```
ethernet add transport ethernet0 ethernet0
```

Attach the transport to the interface:

```
ip attach iplan ethernet0
```

Displaying information about interfaces

To list details about all existing IP interfaces, input the following:

```
ip list interfaces
```

```
IP Interfaces:
  ID | Name      | IP Address  | DHCP  | Transport
-----|-----|-----|-----|-----
   1 | ipwan     | 0.0.0.0     | disabled | PppoeUp
   2 | iplan     | 192.168.1.1 | disabled | ethernet0
-----|-----|-----|-----|-----
```

In this example, the *iplan* interface is your default LAN connection over Ethernet. The *ipwan* interface is your default connection to the WAN, using PPP over Ethernet (PPPoE). Once PPP negotiation starts, the IP address *0.0.0.0* is replaced by an unnumbered interface (IP address *192.168.1.1*, network mask *255.255.255.255*). The interface later retrieves the public IP address via IPCP.

To list details about a specific interface, enter:

```
ip show interface <name>
```

For example:

```
ip show interface iplan
```

```
IP Interface: iplan
```

```
  Ipaddr : 192.168.1.1
```

```
  Mask   : 255.255.255.0
```

```
  MTU    : 1492
```

```
  Dhcp   : false
```

```
TCP MSS Clamp : false
Source Addr Validation : false
Accept V1 : false
Send V1 : false
Accept V2 : false
Send V2 : false
Send Multicast : false
```

Attaching the bridge

You can attach the bridge to the router, allowing transports that are attached to the bridge to route through the IP stack. Enter:

```
ip attachbridge <name>
```

The *<name>* value is the name of the IP interface that you want to attach the bridge to, for example:

```
ip attachbridge iplan
```

For more details, see [Configuring the Bridge](#).

Configuring routes

You can manually create a static route to ensure that packets are forwarded to their correct destination. The route specifies a destination network (or single host) together with a mask to indicate what range of addresses the network covers, and a next-hop gateway address or interface. If there is a choice of routes for a destination, the route with the most specific mask is chosen. Routes are created using the following command:

```
ip add route <name> <dest_ip> <netmask> {gateway <gateway_ip> / interface <interface>}
```

To route to a destination which is not on any local network, a route may be added via a gateway (i.e., another router). The gateway IP address must be on the same subnet as one of the router's interfaces. For example, to ensure that packets destined for the *10.0.0.0* subnet are routed via local gateway address *192.168.3.2*, enter the following:

```
ip add route route_gateway 10.0.0.0 255.0.0.0 gateway 192.168.3.2
```

If the route is via an interface attached to a point-to-point link (such as PPPoA), the interface name can be specified instead of a gateway IP address. This can be useful if the gateway IP address or interface IP address is unknown at the time the route is added. For example, to forward packets via local interface *ip_interface*, enter:

```
ip add route route_interface 10.0.0.0 255.0.0.0 interface ip_interface
```

When a system has one LAN interface and one WAN interface, it is very common for only one route to be needed - a *default* route. This default route would forward all packets whose destination was unknown to a gateway router reached via the WAN interface. You can only create one default route.

To create a default route, use the following command:

```
ip add defaultroute {gateway <gateway_ip> / interface <interface>}
```

For example, to create a default route via the gateway used in the earlier example, enter:

```
ip add defaultroute gateway 192.168.3.2
```

To create a default route via the interface used in the earlier example, enter:

```
ip add defaultroute interface ip_interface
```

Checking IP address command

【command action】 Check line card IP address

【command syntax】 ip list interface

【 Notes】 Interface is layer three interface in the line card

【configuration example】 -->ip list interface

check path information

【 related command】 Check line card gateway IP address

【command syntax】 ip list routes

【configuration example】 -->ip list routes

Adding a default gateway in line card

【command action】 Add line card default gateway

【command syntax】 ip add defaultroute gateway <IP address>

【configuration example】 -->ip add defaultroute gateway 192.168.1.1

Deleting gateway address

【command action】 Delete line card gateway address

【command syntax】 -->ip delete route default < gateway address>

【configuration example】 delete line card gateway address 192.168.1.1

-->ip delete route default 192.168.1.1

Modify line card gateway address

【command action】 Modify line card gateway address

【command syntax】 -->ip set route defaultroute gateway < gateway address>

【configuration example】 Modify line card gateway address to 192.168.1.253

-->ip set route defaultroute gateway 192.168.1.253

Add one layer three interface

【command action】 Add 1 layer three interface

【command syntax】 ip add interface <interface name>

【configuration example】 Add an interface named "iplan"

--> ip add interface iplan

Deleting 1 layer three interface

【command action】 Delete 1 layer three interface

【command syntax】 ip delete interface <interface name>

【configuration example】 delete an interface named "iplan"

--> ip delete interface iplan

Configuration IP address in the line card

【command action】 setup the IP address of an Interface in the Line card

【command syntax】 ip set interface iplan ipaddress <IP address>

【configuration example】 setup IP address of Interface named iplan to 192.168.2.1

-->ip set interface iplan ipaddress 192.168.2.1

Subnet mask in the line card configuration

【command action】 setup Interface subnet mask in the Line card

【command syntax】 ip set interface iplan netmask < subnet mask >

【configuration example】 Configure subnet mask of iplan to 255.255.255.0

-->ip set interface iplan netmask 255.255.255.0

Checking subnet mask

【related command】 check subnet mask

【command syntax】 ip show interface iplan netmask

【configuration example】 --> ip show interface iplan netmask

Interface binding configuration

【command action】 Bind a Layer Two (Transport) and Layer Three (Interface) in the line card

【command syntax】 ip attach <Interface name> <Transport name>

【configuration example】 bind an interface named iplan and transport port named eth1

```
-->ip attach iplan eth1
```

Remove interface binding

【command action】 remove interface binding

【command syntax】 ip detach <Interface name>

【configuration example】 remove the binding of an interface named iplan

```
-->ip detach iplan
```

Execute ping command

【command action】 Execute ping command in line card

【command syntax】 ip ping < IP address>

【configuration example】 Ping 192.168.1.254, and display the result

```
--> ip ping 192.168.1.254
```

if ping is successful, then return parameters, for example:

```
ping: PING 192.168.1.254: 32 data bytes
```

```
ping: 40 bytes from 192.168.1.254: seq=0, ttl=255, rtt<10ms
```

if ping isn't successful, then time out, for example:

```
ping: PING 192.168.1.254: 32 data bytes
```

```
ping: Request timed out.
```

2.4.3 Port Configuration Commands

Ports include: User ports a0~a23, ethernet0 uplink port and Ethernet local system main line port.

Check all port names

【command action】 Check all port names

【command syntax】 port list all

【configuration example】 -->port list all

Check user port state

【command action】 Check the state of each port in the line card

【command syntax】 port < port number> show

【configuration example】 Check the state of the 5th user port (a4)

--> port a4 show

Name	Type	Access	Default	Description
Version	Int	Ro	2.07	Version number, encoded as (major * 100 + minor), where “major” and “minor” are the revision numbers.
ActivateLine	String	R/W	None	Used to cause the link to start a new connection or drop the current one. “Start” – Start line activation/handshaking “Abort” – Drop line “None” – Reported for all reads
Connected	Bool	Ro	false	Current connected state: “true” – modem is connected to a remote modem. “false” – modem is not connected to a remote modem.
Whip	String	R/W	Inactive	Used to enable WHIP interface. “TCP”, “Serial”, “Inactive”
ShowtimeLed	Int	R/W		Logical LED number used to report state: Off - Idle Slow blink – Handshake Rapid blink – Training Steady on – Showtime Value should be a logical LED number or 0xff for no showtime LED.
AutoStart	Bool	R/W	true	Set “true” to enable; “false” to disable automatic connection attempts at power-up.
PhysicalPort	Int	R/W	0	The Utopia Level 2 address used by the device.

Defaults	String	R/W	None	Used to reset the attributes listed in section 5.1.2 to default settings based on the selected annex type. This can be used to quickly configure all attributes necessary to connect with the selected annex type. (Note that for SHDSL builds the only selection for the "Defaults" command is SHDSL. The command has the same effect, however, it resets the SHDSL attributes to default settings.) The options are: (for Annex A/G.span build): "AnnexA" "G.Span," for Annex B build: "AnnexB" and for Annex C.x/I-FM build: "AnnexC" "I-FM".
AturConfDownshiftSnrMgn	Int	R/W	30	Used by the CO to control conditions that trigger downstream SRA. The Downstream Downshift Noise Margin ranges from 0 to 31 dB with 0.1 dB steps. dB/10
AturConfUpshiftSnrMgn	Int	R/W	90	Used by the CO to control conditions that trigger downstream SRA. The Downstream Up-shift Noise Margin ranges from 0 to 31 dB with 0.1 dB steps.
AturConfMinUpshiftTime	Int	R/W	30	Used by the CO to control conditions that trigger downstream SRA. This parameter defines the time interval that the downstream noise margin should stay above the Downstream Upshift Noise Margin before the ATU-R attempts to increase the downstream net data rate. The time interval ranges from 0 to 16383 seconds. (0 indicates disabled)
AturConfMinDownshiftTime	Int	R/W	30	Used by the CO to control the conditions that trigger downstream SRA. This parameter defines the time interval that the downstream noise margin should stay below the Downstream Downshift Noise Margin before the ATU-R attempts to decrease the downstream net data rate. The time interval ranges from 0 to 16383 seconds. (0 indicates disabled)
Action	Enum	R/W	Startup	The possible actions are listed below. This is the action that will be given when ActivateLine is set to Start: "Startup" "DELT"

【command action】 Turn on / off ports on the runtime.

【command syntax】 turn off port: port < port number> set ActivateLine Abort

turn on port: port < port number> set ActivateLine Start

【configuration example】 Turn off the fourth user port

--> port a3 set ActivateLine Abort

Turn on the fifth user port

--> port a4 set ActivateLine Start

Modify port speed

【command action】 Modify port upload / download speed:

(Modify port download speed in the interleave mode)

【command syntax】 port < port number> set AtucChanConfInterleaveMaxTxRate < speed>

【configuration example】 Set up the fourth user port download speed to 8064K (8M) in the interleave mode

-->port a3 set AtucChanConfInterleaveMaxTxRate 8064000

Modify port upload speed in the interleave mode:

【command syntax】 port < port number> set AturChanConfInterleaveMaxTxRate < speed>

【configuration example】 Set up the fourth port upload speed to 1024K (1M) in the interleave mode

-->port a3 set AturChanConfInterleaveMaxTxRate 1024000

Modify port download speed in fast mode:

【command syntax】 port < port number> set AtucChanConfFastMaxTxRate < speed>

【configuration example】 Set up the fourth port download speed to 8064K (8M) in the interleave mode

-->port a3 set AtucChanConfFastMaxTxRate 8064000

Modify upload speed in fast mode:

【command syntax】 port < port number> set AturChanConfFastMaxTxRate < speed>

【configuration example】 Set up the fourth port upload speed to 1024K (1M) in the interleave mode

-->port a3 set AturChanConfFastMaxTxRate 1024000



After modifying a port parameter, you must save and restart the port for the changes to take effect.

Setting ADSL mode

The VX1000 supports different ADSL annex types which can be used to tune the performance of the link under different conditions. Since in firmware it supports automatic negotiation with CPE, the tuning process usually doesn't need administrator intervention. However one situation does need the manual configuration, and that is the annexM mode.

ADSL/ADSL2/ADSL2+ support different downstream data rates, but they all have the same upstream data rate limit, which is about 1mbps. AnnexM relocates the spectrum to support maximum 2mbps upstream data rate. In the current VX1000 product, AnnexM mode works flawlessly with AnnexM-able CPE, however, when using the VX1000 in AnnexM mode with a normal ADSL modem, the link may hang indefinitely. The administrator will need to configure the port to normal ADSL mode if the line suffers hang up under AnnexM mode.

The command to change Annex type is:

```
port <port name> set AtucConfGsAnnexType G.Dmt.Bis
port <port name> set AtucConfGsStandard G.Dmt.BisPlus.Auto
```

This sets the ADSL port to run under normal ADSL/ADSL2/ADSL2+ mode.

```
port <port name> set AtucConfGsAnnexType AnnexM
port <port name> set AtucConfGsStandard G.Dmt.BisPlus.Auto
```

will set the port in AnnexM mode which supports 2mbps upstream rate.

Setup VLAN ID in Line card

【command action】 Setup VLAN ID of ethernet_local

【command syntax】 port ethernet_local set Tag <VLAN ID>

【Notes】 The "T" in Tag must be capitalized.

【configuration example】 Setup the VLAN ID in ethernet_local to 2

```
-->port ethernet_local set Tag 2
```

【related command】 Check VLAN ID of ethernet_local

【command syntax】 port ethernet_local show

Setup VLAN ID on ADSL port

This command will set a VLAN tag on an ADSL port, such that any packets received from the ADSL port will have a VLAN on it. This can be used to differentiate user traffic on uplink.

Command syntax: rfc1483 set transport <transport name> vlantag <vlan number>

Transport name corresponds to the specific PVC in an ADSL port. The default setting has following mapping:

Transport name	ADSL Port	PVC
wb0	a0	0/35
wb1	a1	0/35
wb2	a2	0/35
wb3	a3	0/35
wb4	a4	0/35
wb5	a5	0/35

wb6	a6	0/35
wb7	a7	0/35
wb8	a8	0/35
wb9	a9	0/35
wb10	a10	0/35
wb11	a11	0/35
wb12	a12	0/35
wb13	a13	0/35
wb14	a14	0/35
wb15	a15	0/35
wb16	a16	0/35
wb17	a17	0/35
wb18	a18	0/35
wb19	a19	0/35
wb20	a20	0/35
wb21	a21	0/35
wb22	a22	0/35
wb23	a23	0/35

2.4.4 Ethernet setup commands

Ethernet setup command is mainly for transport port settings.

Add transport port

【command action】 Add one Transport virtual Ethernet port to a specific port

【command syntax】 ethernet add transport < name > < port name>

【configuration example】 Add a transport port named eth0 to ethernet0

--> ethernet add transport eth0 ethernet0

Display all transport port

【command action】 Display all transport ports

【command syntax】 ethernet list transports

【configuration example】 --> ethernet list transports

Check transport port

【 related command 】 Check the port corresponding to transport port

【 command syntax 】 -->ethernet show transport <transport port name>

【 configuration example 】 Check the port corresponding to transport port named eth0t

-->ethernet show transport eth0

Delete transport port

【 command action 】 Delete one Transport virtual Ethernet port

【 command syntax 】 ethernet delete transport < transport port name>

【 configuration example 】 Delete one transport port named eth0

--> ethernet delete transport eth0

2.4.5 Setup bridging commands

The DSLAM bridge operates in the D-bridge mode. As specified in IEEE Standard 802.1D-1998, the principal elements of Bridge operation are:

- Reception, filtering and transmission of frames between the separate MACs of the bridged interfaces connected to a device's ports.
- Maintenance of the information needed to make filtering and relaying decisions.
- Management of the above

DSLAM bridge elements are managed via the CLI and EmWeb. The bridge can be configured to support IGMP snooping functionality. IGMP snoop enables intelligent forwarding of multicast traffic, instead of flooding all ports with multicast packets. This leads to efficient network bandwidth use.

Display all layer two interfaces

【 command action 】 List all layer two interface names and the corresponding transport port(s)

【 command syntax 】 bridge list interfaces

【 configuration example 】 -->bridge list interfaces

Bridge Interfaces:

ID	Name	Filter Type	Transport	Group ID
1	wan0	All	wb0	-1
2	wan1	All	wb1	-1
3	wan2	All	wb2	-1
4	wan3	All	wb3	-1
5	wan4	All	wb4	-1
6	wan5	All	wb5	-1
7	wan6	All	wb6	-1
8	wan7	All	wb7	-1
9	wan8	All	wb8	-1
10	wan9	All	wb9	-1
11	wan10	All	wb10	-1
12	wan11	All	wb11	-1
13	wan12	All	wb12	-1
14	wan13	All	wb13	-1
15	wan14	All	wb14	-1
16	wan15	All	wb15	-1
17	wan16	All	wb16	-1
18	wan17	All	wb17	-1
19	wan18	All	wb18	-1
20	wan19	All	wb19	-1
21	wan20	All	wb20	-1
22	wan21	All	wb21	-1
23	wan22	All	wb22	-1
24	wan23	All	wb23	-1
25	uplink	All	eth0	-1

The *name* field shows the name of the interfaces; *filter type* shows what kind of packets can pass through the interface; *Transport* shows the name of the transports; *Group ID* shows which group the interface belongs to (-1 means no group association for the interface).

Show the interface

This command shows the detail of the interface:

```
bridge show interface <interface name>
```

```
--> bridge show interface uplink

Bridge Interface:  uplink
Filter Type:      All
Port Filter:      All
Transport:        eth0
LinkMode:         Uplink
PortGroup:        -1
Leave Mode:        Normal
```

Port filter shows which port that this port will forward the packets to; *LinkMode* shows if this interface is in uplink mode or not. If an interface is in uplink mode, then all other ports will forward packets to this interface. *Leave Mode* shows IGMP leave message handling; *Normal* means to wait sometime before serving the IGMP forwarding, *fast* means to shutdown

IGMP forwarding as soon as it receives the IGMP leave message (see IGMP snooping section for details).

Add one layer two interface

【command action】 Add a layer two interface to some port

【command syntax】 bridge add interface < layer two interface name>

【configuration example】 Add a layer two interface named wan0 to a0 port

```
-->bridge add interface wan0
```

Add a layer two interface named uplink to ethernet0

```
-->bridge add interface uplink
```

Delete one layer two interface

【command action】 Delete 1 layer two interface

【command syntax】 bridge delete interface < layer two interface name>

【configuration example】 Delete 1 layer two interface named wan9

```
--> bridge delete interface wan9
```

Binding Configuration between layer two interface and transport port

【command action】 Bind a layer two interface corresponding to the port with the corresponding virtual Ethernet port

【command syntax】 bridge attach < layer two interface name> < virtual Ethernet port name>

【configuration example】 Bind a layer two interface named wan0 corresponding to a0 port with a virtual Ethernet port named wb0

```
--> bridge attach wan0 wb0
```

Bind a layer two interface corresponding to ethernet0 port with the corresponding virtual Ethernet port

```
--> bridge attach uplink eth0
```

Remove layer two interface binding

【command action】 Remove a layer two interface binding related to some port

【command syntax】 bridge detach < layer two interface name>

【configuration example】 Remove the binding of layer two interface named wan0 corresponding to a0 port

```
--> bridge detach wan0
```

Remove the binding of a layer two interface named uplink corresponding to ethernet0 port

--> bridge detach uplink

2.4.6 rfc1483 configuration commands

The RFC 1483 transport provides the simplest method of connecting end stations over an ATM network. RFC 1483 can either be used in bridged mode, where Ethernet packets are encapsulated in AAL-5 (ATM Adaptation Layer-5) PDUs (Protocol Data Units), or routed mode, where IP packets are encapsulated in AAL-5 PDUs.

It is most often used in bridged mode, which allows MAC-level Ethernet to be bridged directly to RFC 1483. In routed mode, the RFC 1483 transport provides a simple point-to-point link for routing IP packets, but without any of the authentication or configuration features of PPP.

Adding transport ports

This command creates a named RFC1483 transport and allows you to specify the following:

- The ATM port that will transport RFC1483 data. (ATM ports are initialized in the initbun file in FlashFS, or using the *bun set port* console command.)
- VPI (Virtual Path Identifier)
- VCI (Virtual Circuit Identifier)
- LLC or VcMux encapsulation (optional)
- Bridged or Routed (optional)

The port/VPI/VCI combination must be unique for each transport.

The following table gives the range of values for each option, which can be specified with this command and a default value (if applicable):

Option	Description	Default value
name	An arbitrary name that identifies the transport. It can be made up of one or more letters, or a combination of letters and digits, but it cannot start with a digit.	N/A
port	The system port that is used to transport ATM data.	N/A
vpi	A field in the ATM header. The VPI is used to identify the virtual path that a circuit belongs to. The VPI can be any value between 0 and 4095.	N/A

vci	Part of the ATM header. The VCI is a tag that identifies which channel a cell will travel over. The VCI can be any value between 1 and 65535.	N/A
llc	Logical Link Control encapsulation method.	llc
vmux	VC Multiplexing encapsulation method.	
bridged	Traffic type that is going to be transmitted/received.	bridged
routed	Traffic type that is going to be transmitted/received.	

【command action】 Add virtual Ethernet port (transport) on ADSL port

【command syntax】 rfc1483 add transport <transport name><ADSL port name > <VPI value> <VCI value>< MULTIPLEXING METHOD >bridged

【Notes】<VPI value> <VCI value>< MULTIPLEXING METHOD >VPI/VCI value and ATM MULTIPLEXING METHOD used by ADSL port

【configuration example】 Add a virtual Ethernet port named wb0 to a0 port for bridging

-->rfc1483 add transport wb0 a0 8 81 llc bridged

Delete transport port

【command action】 Delete virtual Ethernet port corresponding to ADSL port

【command syntax】 rfc1483 delete transport <transport name>

【configuration example】 Delete transport port wb0 corresponding a0

--->rfc1483 delete transport wb0

Configuration user port VLAN

【command action】 Setup user port VLAN ID number

【command syntax】 rfc1483 set transport <transport name> vlantag <ID number>

【Notes】 ID number range 1~4095

【configuration example】 Setup VLAN ID on the a0 port to 1014

-->rfc1483 set transport wb0 vlantag 1014

Check user port VLAN ID

【related command】 Check user port VLAN ID

【command syntax】 rfc1483 show transport <transport name>

【Notes】 The corresponding transport name for a0~a7 user port is wb0-wb7

【configuration example】 Check VLAN ID on the a0 port

```
-->rfc1483 show transport wb0
```

2.4.7 SNMP access CLI commands

The table below lists the *SNMP* commands provided by the CLI:

Command	Description/Console command
snmp add communityname	<i>snmp add communityname on page 1137</i>
snmp delete communityname	<i>snmp delete communityname on page 1139</i>
snmp list communitynames	<i>snmp list communitynames on page 1140</i>
snmp add trapdestination	<i>snmp add trapdestination on page 1141</i>
snmp delete trapdestination	<i>snmp delete trapdestination on page 1143</i>
snmp list trapdestinations	<i>snmp list trapdestinations on page 1144</i>
snmp set config syscontact	<i>snmp set config syscontact on page 1145</i>
snmp set config sysdescr	<i>snmp set config sysdescr on page 1146</i>
snmp set config syslocation	<i>snmp set config syslocation on page 1147</i>
snmp set config sysname	<i>snmp set config sysname on page 1148</i>
snmp set config sysobjectid	<i>snmp set config sysobjectid on page 1149</i>
snmp show config	<i>snmp show config on page 1151</i>

snmp add communityname

Syntax

```
snmp add communityname <name> <community> [{read | write} [<ipAddress>]]
```

Description

This command allows you to create a community name. A community name acts as a password for the specified community, and is used to authenticate information sent between a manager and an agent. The name is sent in every packet between the manager and the agent.

The *read* and *write* options determine whether information can be read, or both read and written by a manager. If an IP address

is specified, the community name is valid only for SNMP requests issued from that IP address. The same community name can be configured several times with different IP addresses, allowing access with the same community name from a number of different machines.

Options

The following table describes the options that can be specified in this command and displays the default values (if applicable).

Option	Description	Default value
name	An arbitrary name that identifies the SNMP community. It can be made up of one or more letters, or a combination of letters and digits, but it cannot start with a digit.	N/A
community	The SNMP name of a community. The name is a string of up to 255 characters (no spaces).	N/A
read	Sets the access rights of the community/manager to read-only.	read
write	Sets the access rights of the community/manager to read-write.	
ipAddress	The IP address of the manager from which requests are restricted by this entry. This should be an IPv4 address in the range 0.0.0.0 - 255.255.255.255. 0.0.0.0 means no restriction on source address.	0.0.0.0

Example

```
prompt> snmp add communityname tonyget tony.read read 192.168.88.110
```

snmp delete communityname

Syntax

```
snmp delete communityname <name>
```

Description

This command deletes a community name previously created using the *snmp add communityname* command.

Options

The following table describes the options that can be specified in this command, and displays the default values (if applicable).

Option	Description	Default value
name	A name that identifies an existing community. To display community names, use the <code>snmp list communitynames</code> command	N/A

Example

prompt> **snmp delete communityname tonyget**

snmp list communitynames

Syntax

snmp list communitynames

Description

This command displays information about the SNMP community names previously created using the command [snmp add communityname](#) on page 1137. It displays the following information about community names:

- name ID
- name or password assigned to a community
- default SNMP community name
- Access set; *read* or *write*
- IP address of the SNMP Manager (*0.0.0.0* means none set)

Example

prompt> **snmp list communitynames**

```

-----
ID | Name | SNMP Community Name | Access| SNMP Manager
---|-----|-----|-----|-----
1 | public | public | read | 0.0.0.0
2 | tonyread | tony.read | read | 192.168.88.110
3 | tonyset | tony.write | write | 192.168.88.110
-----

```

snmp add trapdestination

Syntax

snmp add trapdestination <name> <community> <ipaddress> [v1|v2[<portNumber>]]

Description

This command allows you to set up an SNMP trap destination. A trap is a message sent by the SNMP agent to a specified IP address to indicate the occurrence of a significant event, such as a specifically defined condition or a threshold that was reached. This command allows you to determine where traps are sent.

Options

The following table describes the options that can be specified in this command and displays the default values (if applicable).

Option	Description	Default value
name	A name that identifies an SNMP trap destination. To display trap destination names, use the snmp list trapdestinations	N/A

	command.	
community	The SNMP name of a community. The name is a string of up to 255 characters (no spaces).	N/A
ipaddress	IP address of the machine that you want traps to be sent to.	N/A
v1	The trap version used is v1: trap.	v1
v2	The trap version used is v2: notification.	
portNumber	The port number (from the range 0-65535) to which traps are sent. A value of 0 means that traps are sent to the SNMP agent's default trap port number (usually 162).	0

Example

```
prompt> snmp add trapdestination tonyv1 tony.trap 192.168.88.110 v1 162
```

snmp delete trapdestination

Syntax

```
snmp delete trapdestination <name>
```

Description

This command deletes an SNMP trap destination previously created using the `snmp add trapdestination` on page 1141 command.

snmp list trapdestinations

Syntax

```
snmp list trapdestinations
```

Description

This command displays information about the SNMP trap destinations previously created using the command [snmp add trapdestination](#). It displays the following information about trap destinations:

- ID
- trap name
- SNMP trap community name
- IP address of the destination machine
- SNMP version supported (v1 or v2)
- port used to send SNMP trap messages

Example

```
prompt> snmp list trapdestinations
```

```
-----
ID | Name | SNMP Trap Community Name | Destination | Ver | Port
----|-----|-----|-----|----|-----
```

1 | tonyv2c | tony.trap | 192.168.88.110 | v2c | 0

2 | tonyv1 | tony.trap | 192.168.88.110 | v1 | 162

snmp set config syscontact

Syntax

snmp set config syscontact <contactDetails>

Description

This command allows you to set contact details for the person responsible for maintaining your SNMP agent. When a request for contact details is received from the SNMP manager, the details set by this command are returned.

Options

The following table describes the options that can be specified in this command and displays the default values (if applicable).

Option	Description	Default value
contactDetails	Contact details (e.g., telephone number, email address) for the person responsible for maintaining the SNMP agent system. The details are represented by a string of up to 255 characters (no spaces). To display the contact details, use the <code>snmp show config</code> command.	N/A

Example

prompt> snmp set config syscontact ck@email.com

snmp set config sysdescr

Syntax

snmp set config sysdescr <systemDescription>

Description

This command allows you to set a description of your SNMP agent. When a request for the agent's description is received from the SNMP manager, the description set by this command is returned.

Options

The following table describes the options that can be specified in this command and displays the default values (if applicable).

Option	Description	Default value
	A description of the SNMP agent system. The description is represented	

systemDescription	by a string of up to 255 characters (no spaces). If you wish to add spaces, enclose the description in double quotation marks (" "). To display the system description, use the snmp show config command.	N/A
-------------------	---	-----

Example

prompt> **snmp set config sysdescr "Versatek DSLAM"**

snmp set config syslocation

Syntax

snmp set config syslocation <systemLocation>

Description

This command allows you to configure the location of your SNMP agent. When a request for the agent's location is received from the SNMP manager, the location set by this command is returned.

Options

The following table describes the options that can be specified in this command and displays the default values (if applicable).

Option	Description	Default value
systemLocation	A name that identifies the location of the SNMP agent system. The location is represented by a string of up to 255 characters (no spaces). If you wish to add spaces, enclose the description in double quotation marks (" "). To display the system location, use the snmp show config command.	N/A

Example

prompt> **snmp set config syslocation "Versatek #102 91710"**

snmp set config sysname

Syntax

snmp set config sysname <systemName>

Description

This command allows you to set a name for your SNMP agent. When a request for the agent's object name is received from the SNMP manager, the name set by this command is returned.

Options

The following table describes the options that can be specified in this command, and displays the default values (if applicable).

Option	Description	Default value
systemName	<p>A name that identifies the system that the SNMP agent is running on. The name is a string of up to 255 characters (no spaces). If you wish to add spaces, enclose the description in double quotation marks (" "). To display the system name, use the <code>snmp show config</code> command.</p>	N/A

Example

```
prompt> snmp set config sysname "Chino 220"
```

snmp set config sysobjectid

Syntax

```
snmp set config sysobjectid <OID>
```

Description

This command allows you to configure the object IDs (or OIDs) included in *get* *get-next* and *set* requests. An object ID identifies one of the variables contained in the SNMP agent's database (or MIB).

Options

The following table describes the options that can be specified in this command, and displays the default values (if applicable).

Option	Description	Default value
OID	<p>A series of non-negative integers that identifies individual variables contained in the SNMP agent's database. Conexant has control over all OIDs starting with 1.3.6.1.4.1.1218 (assigned by the Internet Assigned Numbers Authority (IANA)). You can refine OIDs by adding more components at the end of the integer.</p>	N/A

Example

```
prompt> snmp set config sysobjectid 1.3.6.1.4.1.4154.1.6
```

snmp show config

Syntax

```
snmp show config
```

Description

This command displays information about the system on which your SNMP agent is running.

Example

prompt> **snmp show config**

```
sysdescr: Versatek DSLAM
sysobjectid: 1.3.6.1.4.1.4154.1.6
syslocation: Chino #102 91710
syscontact: ck@email.com
sysname: Chino 220
authentraps: disabled
```

2.4.8 IGMP Snoop Support

Overview

An IGMP Snoop switch provides the benefit of conserving bandwidth on those segments of the network where no node has expressed interest in receiving packets addressed to the multicast group address. This is in contrast to the normal switch behavior where multicast traffic is typically forwarded on all interfaces.

The IGMP Snooping switch listens to IGMP reports, queries and leave messages sent between hosts and a multicast router, to identify the interfaces that are members of multicast groups. Based on this information, it adds/deletes multicast entries from its filtering database, ensuring that multicast traffic is only forwarded to interfaces identified as members of the specific multicast group.

In ISOS, IGMP Snoop is implemented in two modes - Proxy and Snoop-only, with 'Snoop-only' being the default mode. The 'Proxy mode' is supported by means of IGMP proxy-reporting, where the reports received from the downstream hosts are summarized and then the switch reports its own state in response to the upstream queries from multicast routers. The switch also acts as a Querier, generating queries periodically on the downstream interfaces. The 'Snoop only' mode is implemented by snooping through the IGMP packets and forwarding the IGMP packets received on the upstream interfaces to all other interfaces. As the queries received from the upstream interfaces are forwarded to the downstream interfaces, periodic queries are not generated, unlike the proxy mode. The IGMP packets received on a downstream interface are also forwarded to all of the upstream interfaces.

IGMP Snoop also supports two leave processing modes for each bridge interface - Fast and Normal, with 'Normal' being the default mode. In the 'Fast' mode of leave processing, on receiving a leave message on a downstream interface, IGMP Snoop will simply delete the interface from the group membership information, and the 'Leave' message is forwarded to the upstream interfaces. 'Fast' Mode for an interface should be configured when it is known that there is only one host behind the interface. Fast leave processing helps to reduce the latency involved in removing an interface from the group membership information. In the 'Normal' mode of leave processing, on receiving a leave message on a downstream interface, IGMP snoop will repeatedly generate group-specific queries on the interface. Failure to receive any membership report in response will result in deletion of the interface from the group membership information.

Benefits of IGMP Snooping

In D-bridge, all multicast packets are treated like broadcast packets that are forwarded on all ports in the forwarding state. This results in less efficient utilization of network bandwidth, as multicast traffic is forwarded on interfaces where no node has any interest in receiving the packet.

IGMP snooping enables intelligent forwarding of multicast traffic (instead of flooding to all ports). Multicast packets that belong to a layer 2 multicast group are only forwarded to an interface if a host on that interface has expressed interest in the same group. This significantly reduces flooding of multicast data, resulting in better utilization of network bandwidth and improved bridge performance.

bridge set igmpsnoop mode

Syntax

```
bridge set igmpsnoop mode {snooponly|proxy}
```

Description

This command sets the mode for IGMP snoop functionality in the bridge. The current IGMP Snoop mode setting is displayed by the bridge show command.

Options

The following table gives the range of values for each option, which can be specified with this command, and a default value (if applicable).

Option	Description	Default value
snooponly	The 'Snoop only' mode is implemented by snooping through the IGMP packets and forwarding IGMP packets received on upstream interfaces to all other interfaces. IGMP packets received on a downstream interface are also forwarded to all upstream interfaces.	snooponly
proxy	'Proxy mode' is supported by means of IGMP proxy-reporting where the reports received from the downstream hosts are Summarized, and the switch then reports its own state in response to upstream queries from multicast routers. The switch also acts as a Querier, generating queries periodically on the downstream interfaces.	snooponly

bridge set igmpsnoop

Syntax

```
bridge set igmpsnoop {enabled|disabled}
```

Description

This command enables/disables the IGMP Snoop functionality in the bridge. When the IGMP snoop functionality is enabled, all of the attached bridge interfaces are designated as downstream interfaces. The downstream interface implements the Router portion of the IGMP protocol. The current IGMP Snoop setting is displayed by the *bridge show* command.

Options

The following table gives the range of values for each option, which can be specified with this command, and a default value (if applicable).

Option	Description	Default value
enabled	This option enables IGMP Snoop functionality in the bridge.	disabled
disabled	This option disables IGMP Snoop functionality in the bridge.	disabled

bridge set interface igmpsnoop leavemode

Syntax

```
bridge set interface {<name>|<number>} igmpsnoop leavemode
{fast|normal}
```

Description

This command specifies the IGMP snoop leave processing mode of a bridge interface.

Options

The following table gives the range of values for each option, which can be specified with this command, and a default value (if applicable).

Option	Description	Default value
name	A name that identifies an existing bridge interface. To display interface names, use the <i>bridge list interfaces</i> command.	N/A
number	A number that identifies an existing bridge interface. To display interface names, use the <i>bridge list interfaces</i> command. The number appears in the first column under the heading ID.	N/A

fast	In the 'Fast' mode of leave processing, on receiving a 'Leave' message on a downstream interface, IGMP Snoop shall simply delete the interface from the group membership information.	Normal
normal	In the 'Normal' mode of leave processing, on receiving a Leave message on a downstream interface, IGMP snoop shall repeatedly generate group specific queries on the interface. Failure to receive any membership report in response will result in deletion of the interface from the group membership information.	Normal

bridge set igmpsnoop lastmemberqueryintvl
Syntax

```
bridge set igmpsnoop lastmemberquerintvl <queryinterval>
```

Description

This command sets the last member query interval, which is the maximum response time inserted into group-specific queries sent in response to leave group messages. It is also the amount of time between group-specific query messages. The current last member query interval setting is displayed by the *bridge show* command.

Options

The following table gives the range of values for each option, which can be specified with this command, and a default value (if applicable).

Option	Description	Default value
queryinterval	This can be any value between 1 and 255 seconds.	1 second

bridge set igmpsnoop mcastroutertimeout
Syntax

```
bridge set igmpsnoop mcastroutertimeout <timeout>
```

Description

This command sets the multicast router timeout interval, which is the time a dynamic multicast router interface remains an upstream interface after receiving an IGMP Query with a non-zero source IP address. If an IGMP Query with a non-zero source IP address is not received on the dynamic multicast router interface during this time interval, the dynamic multicast router interface (upstream) is reverted back to a downstream interface. The multicast router timeout interval setting is displayed by the *bridge show* command.

Options

The following table gives the range of values for each option, which can be specified with this command, and a default value (if applicable).

Option	Description	Default value
timeout	This can be any value between 1 and 65535 seconds.	400 seconds

bridge set igmpsnoop queryintvl

Syntax

bridge set igmpsnoop queryintvl <queryinterval>

Description

This command sets the interval between general queries that are periodically generated on downstream interfaces. The query interval setting is displayed by the *bridge show* command.

Options

The following table gives the range of values for each option, which can be specified with this command, and a default value (if applicable).

Option	Description	Default value
queryinterval	This can be any value between 2 and 255 seconds but is constrained by queryresponseintvl. The queryintvl and queryresponseintvl are related as follows: queryresponseintvl < queryintvl	125 seconds

bridge set igmpsnoop queryresponseintvl

Syntax

bridge set igmpsnoop queryresponseintvl <responseinterval>

Description

This command sets the maximum response time inserted in the general queries that are periodically generated on downstream interfaces. The current query response interval setting is displayed by the *bridge show* command.

Options

The following table gives the range of values for each option, which can be specified with this command, and a default value (if applicable).

Option	Description	Default value
response interval	This can be any value between 1 and 254 seconds but is constrained by queryintvl. The queryresponseintvl and queryintvl are related as follows: queryresponseintvl < queryintvl	10 seconds

bridge set igmpsnoop robustnessvar

Syntax

bridge set igmpsnoop robustnessvar <robustness>

Description

This command sets the robustness variable, which allows tuning for the expected packet loss on a subnet. If a subnet is expected to be lossy, the robustness variable may be increased. The current robustness variable setting is displayed by the *bridge show* command.

Options

The following table gives the range of values for each option, which can be specified with this command, and a default value (if applicable).

Option	Description	Default value
robustness	This can be any value between 2 and 255.	2

bridge set igmpsnoop v1timer

Syntax

bridge set igmpsnoop v1timer <v1timer>

Description

This command sets the IGMP Version 1 query router present timeout, which is the time a host must wait after hearing a Version 1 query before it may send any IGMPv2 messages. The current v1 timer setting is displayed by the *bridge show* command.

Options

The following table gives the range of values for each option, which can be specified with this command, and a default value (if applicable).

Option	Description	Default value
v1timer	This can be any value between 1 and 65535 seconds	400 seconds

bridge add igmpsnoop mcastrouterintf

Syntax

bridge add igmpsnoop mcastrouterintf <interface>

Description

This command adds a static multicast router interface. A multicast router interface is also called an upstream interface and a multicast router is connected to this interface. The upstream interface implements the Host portion of the IGMP protocol. The IGMP membership reports and leaves group messages are forwarded on the upstream interfaces.

Options

The following table gives the range of values for each option, which can be specified with this command, and a default value (if applicable).

Option	Description	Default value
interface	The name of a bridge interface that has previously been added and attached to a transport using the <i>bridge add interface</i> and <i>bridge attach</i> CLI commands, respectively.	N/A

bridge delete igmpsnoop mcastrouterintf

Syntax

bridge delete igmpsnoop mcastrouterintf <interface>

Description

This command deletes a previously added static multicast router using the *bridge add igmpsnoop mcastrouterintf* CLI command. The interface reverts to a downstream interface after deletion.

Options

The following table gives the range of values for each option, which can be specified with this command, and a default value (if applicable)

Option	Description	Default value
interface	The name of a bridge interface that has previously been added and attached to a transport using the <i>bridge add interface</i> and <i>bridge attach</i> CLI commands, respectively.	N/A

bridge list igmpsnoop static mcastrouterintfs

Syntax

bridge list igmpsnoop static mcastrouterintfs

Description

This command displays all of the static multicast router interfaces that were previously added using the *bridge add igmpsnoop mcastrouterintf* CLI command.

bridge list igmpsnoop mcastrouterintfs

Syntax

bridge list igmpsnoop mcastrouterintfs

Description

This command displays all the multicast router interfaces. These interfaces are either static or dynamic. Static multicast router interfaces are added through the *bridge add igmpsnoop mcastrouterintf* CLI command. A downstream interface is added as a dynamic multicast router interface on receiving an IGMP query message with a non-zero source IP address on the interface. After multicast router timeout on the dynamic multicast router interface, it again reverts back to a downstream interface.

bridge list igmpsnoop groupinfo

Syntax

bridge list igmpsnoop groupinfo

Description

This command displays the following information for each downstream interface:

- Interface Name
- Group Address

The interface of bridge with the higher layer (for example, IP) is also designated as a downstream interface, and shown as "LocalIntf" in the output of this command.

bridge list igmpsnoop interfacestats

Syntax

bridge list igmpsnoop interfacestats

Description

This command displays the following statistical information for each bridge interface:

- Interface Name
- Number of IGMP V2 Queries transmitted
- Number of IGMP V1 Queries transmitted
- Number of IGMP V2 Queries received
- Number of IGMP V1 Queries received
- Number of IGMP Leave messages received
- Number of IGMP Leave messages transmitted
- Number of IGMP V2 Reports received
- Number of IGMP V1 Reports received
- Number of IGMP V2 Reports transmitted
- Number of IGMP V1 Reports transmitted
- Number of IGMP messages discarded

The interface of the bridge with the higher layer (for eg., IP) is also designated as a downstream interface, and shown as

"LocalIntf" in the output of this command.

bridge show igmpsnoop groupinfo

Syntax

bridge show igmpsnoop groupinfo <interface>

Description

This command displays the IGMP multicast group membership information on the specified downstream interface. Additionally, in Q-bridge mode it also displays the VLAN Name for the multicast groups.

Note: If the interface specified is an upstream interface, no group information will be displayed.

Options

The following table gives the range of values for each option, which can be specified with this command, and a default value (if applicable).

Option	Description	Default value
interface	The name of a bridge interface that has previously been added and attached to a transport using the <i>bridge add interface</i> and <i>bridge attach</i> CLI commands, respectively.	N/A