EliteView™

SNMP-Based
Network Management Software
for Windows™

SMC®

User Guide

EliteView 6.0

EliteView™
User Guide

SNMP-Based Network Management Software for Windows™

6 Hughes
Irvine, CA 92618
Phone: (949) 707-2400

January 2001 R01

SMC Networks, Inc.
6 Hughes
Irvine, CA 92618

Trademarks:

SMC®

Licence Agreement

The Purchaser MUST read this agreement before using the software provided in this package. As used in this Agreement the terms "You", "Your", "LICENSEE", and "Purchaser" refer to the licensee of the SOFTWARE and accompanying written materials in this package, as described below. If you do not agree to the terms of this License Agreement, return the package, unused, to the point of purchase for a refund of any license fee which you paid.

1. Grant of license

SMC Networks, Inc. (SMC) grants to you a non-exclusive right to use one copy of the program (hereinafter the "SOFTWARE"). The SOFTWARE is provided by SMC on CD-ROM. SMC also grants to you a non-exclusive right to use the programs on the enclosed CD-ROM (hereinafter the "SOFTWARE") on one local area network. SMC reserves all rights not expressly granted to the LICENSEE.

2. Ownership of license

All SOFTWARE, and each copy thereof, remains the property of SMC and/or one or more of its licensors. This license is not a sale of any SOFTWARE, or of any copy. You disclaim forever any rights of ownership in any SOFTWARE, or in any copy. The only rights that you obtain with respect to any SOFTWARE are those expressly set forth in this License Agreement.

3. Copy restriction

You acknowledge that the SOFTWARE and accompanying written materials are copyrighted and entitled to protection under the copyright laws of nations where the SOFTWARE is installed, and under international conventions. Except for the copying of SOFTWARE as expressly permitted in the next sentence, you may not copy any SOFTWARE or accompanying written materials, or any portion of any of them, or remove any copyright or proprietary notice from any of them; and you agree not to allow any other person or entity to do so. Subject to these restrictions, you may make copies of the SOFTWARE solely for backup purposes.

4. Use restriction

As the LICENSEE, you may physically transfer the SOFTWARE from one computer to another within a licensed local area network. You may not distribute copies of the SOFTWARE or accompanying written materials to others. Furthermore, the LICENSEE is not permitted to modify, disassemble, reverse or de-engineer, or create any derivative works based on the SOFTWARE.

5. Transfer restrictions

This SOFTWARE is licensed only to you, the LICENSEE, and may not be transferred to anyone without prior written consent of SMC. Any authorized transferee of the SOFTWARE shall be bound by the terms and conditions of this Agreement. In no event may you transfer, assign, rent, lease, sell or otherwise dispose of the SOFTWARE on a temporary or permanent basis except as expressly provided herein.

6. Termination

This license agreement is effective until terminated by SMC. This license may be terminated without notice by SMC if the LICENSEE fails to comply with any provision of this license. Upon termination, it is the LICENSEE's obligation to destroy the written materials and all copies of the SOFTWARE. Further use of the SOFTWARE components will be in violation of this license.

Limited Warranty

The Warranty Period for the media on which the SOFTWARE is recorded is for 5 years from the date of its delivery to the original user as evidenced by a receipt.

You must give SMC prompt written notice within the Warranty Period of any warranty claim. Conditioned on your providing prompt written notice to SMC within the Warranty Period, SMC shall, as your sole and exclusive remedy, repair or replace media not conforming to this warranty, or accept return of any such nonconforming media. The election of whether to repair, replace, or refund shall be SMC's, in its sole discretion.

EXCEPT FOR THE EXPRESS WARRANTY SET FORTH ABOVE IN THIS LIMITED WARRANTY SECTION, THERE ARE NO WARRANTIES, EITHER EXPRESS OR IMPLIED. THE SOFTWARE IS PROVIDED AS IS, WITH ALL FAULTS. SMC DISCLAIMS AND EXCLUDES ANY AND ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY AND ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AGAINST INFRINGEMENT AND THE LIKE, AND ANY AND ALL WARRANTIES ARISING FROM ANY COURSE OF DEALING OR USAGE OF TRADE. YOU

ACKNOWLEDGE THAT YOU ARE NOT RELYING ON SMC'S SKILL OR JUDGEMENT TO SELECT OR FURNISH SUITABLE GOODS.

PRODUCTS AND SERVICES PROVIDED BY SMC ARE NOT DESIGNED, INTENDED OR AUTHORIZED FOR USE IN ANY LIFE SUPPORT OR OTHER APPLICATION WHERE PRODUCT FAILURE COULD CAUSE OR CONTRIBUTE TO PERSONAL INJURY OR SUBSTANTIAL PROPERTY DAMAGE. IF YOU APPLY ANY PRODUCT OR SERVICE PURCHASED OR LICENSED FROM SMC TO ANY SUCH UNINTENDED OR UNAUTHORIZED USE, YOU SHALL INDEMNIFY AND HOLD SMC, ITS AFFILIATES AND THEIR RESPECTIVE SUPPLIERS, HARMLESS AGAINST ALL CLAIMS, COSTS, DAMAGES AND EXPENSES ARISING, DIRECTLY OR INDIRECTLY, OUT OF ANY SUCH UNINTENDED OR UNAUTHORIZED USE, EVEN IF SUCH CLAIM ALLEGES THAT SMC OR ANY OTHER PERSON OR ENTITY WAS NEGLIGENT IN DESIGNING OR MANUFACTURING THE PRODUCT.

Limitation of Liability and Damages

IN NO EVENT SHALL SMC BE LIABLE FOR INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES; OR FOR LOST PROFITS, SAVINGS OR REVENUES OF ANY KIND; OR FOR ANY ACTUAL DAMAGES IN EXCESS OF $100,000; REGARDLESS OF THE FORM OF ACTION, WHETHER BASED ON CONTRACT; TORT; NEGLIGENCE OF SMC OR OTHERS; STRICT LIABILITY; BREACH OF WARRANTY; OR OTHERWISE; WHETHER OR NOT ANY REMEDY IS HELD TO HAVE FAILED OF ITS ESSENTIAL PURPOSE, AND WHETHER OR NOT SMC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

SMC Networks, Inc.
6 Hughes
Irvine, CA 92618

About This Guide

Welcome to EliteView network management software (version 6.01). This guide describes the procedures for installing and using this program. Detailed instructions on how to use every module and practical examples make it easy for you optimize the performance and efficiency of your network.

General Manual Coverage

This manual is your guide to using EliteView to manage all your network resources. It covers the following topics:

Introduction to EliteView	Chapter 1
Software Installation	Chapter 2
Getting Started	Chapter 3
Mapping Out Your Network	Chapter 4
Basic Management Tools	Chapter 5
SNMP MIB Management	Chapter 6
Logging Network Statistics	Chapter 7
Managing Events	Chapter 8
Using RMON	Chapter 9

Conventions Used in this Manual

This manual uses a variety of editorial and typographical conventions to help you locate and interpret information easily. These conventions are:

italics	Italicized words refer to keyboard keys you have to press, menu options you have to select to invoke an action, or a section in this manual or another manual you can consult for reference. For example: <*Enter*>, *Reboot*
`Courier New`	Words shown in `Courier New` typeface are used to represent screen displays, filenames, or commands you must enter to initiate an action.

Additional References

System Help

Help files may be accessed directly from EliteView via on-line help. To get help, select *Help* from the menu bar and then click on *Index*. Last-minute information regarding EliteView has been recorded in the **README.TXT** file.

TABLE OF CONTENTS

GLOSSARY
INDEX

CHAPTER 1

INTRODUCTION

EliteView provides a user-friendly interface for managing SMC brand and third-party network devices. This software is specifically designed to support the efforts of the MIS manager, system administrator(s), technical staff responsible for network management and maintenance, and network operators who use the system on a daily basis.

EliteView provides all the tools you need to manage nearly any kind of network. You can readily monitor the traffic load throughout the network and make the changes required to avoid major crises ahead of time. This software is designed around an event-driven architecture, which allows you to define event-handling routines that can automatically manage a wide variety of common network tasks.

General Description

EliteView is based on the industry standard Simple Network Management Protocol (SNMP); and provides protocol support for UDP/IP and IPX. EliteView is a Windows-based application used to manage nearly every component in your network, from internetworking devices down to end-node computer resources.

EliteView manages network devices using the comprehensive Management Information Base. This MIB consists of various MIB modules which define basic system parameters for both general and specific device types.

EliteView is based on a broad conceptual interface that permits it to manage any network device that supports SNMP. By opening your network map and clicking on various objects, specific management interfaces and system information can be readily accessed.

Interface statistics and traffic load can be illustrated by line charts. This information can be automatically recorded by the Log Manager to maintain historical records. A powerful reporting feature is also provided for recording significant information from any management window. Reports can be edited, saved and retrieved again during a later session for subsequent analysis or comparison.

Event management is a key feature of EliteView. By defining specific data filters and thresholds, you can activate event-handling routines that help you keep the network functioning. You can easily shut down malfunctioning ports, switch to backup systems, or reconfigure network connections; and then restore system parameters back to normal values after component problems have been resolved or the traffic falls off.

Management Functions

EliteView is a Windows-based software package. It provides state-of-the-art utilities which allow you to perform the following network management tasks:

- Generate a detailed hierarchial map of your entire network configuration. These maps display the current status of network nodes, and provide a hot link to the management module for each device.
- Maintain centralized boot services that provide network addresses and information on system files to download. Boot services are used to quickly reassign network addresses, and fetch filenames required for downloading frequently modified system software for test devices.
- Monitor and log significant events and statistics. EliteView provides access to common MIB variables, as well as specific parameters for SMC devices. Network statistics can then be displayed in tabular or graphic form.
- Automatically respond to network problems with a variety of actions. By defining thresholds for parameters based on device-specific criteria or traffic loading, you can invoke event handling routines designed to warn the network manager of potential system problems or automatically take corrective action.
- Quickly fetch or set MIB variables for network devices. Data in the Management Information Base (e.g., RFC 1213 for generic internetworking devices) can be managed on an item-by-item basis or with a user-defined profile that fetches or sets a group of related variables in a single action.
- Remotely manage or reconfigure network devices. SMC provides a wide variety of intelligent networking devices which can be remotely managed via an SNMP agent. Software modules based on an advanced graphic user interface are provided to manage every aspect of these devices including hubs and switches. Extensive management functions are also provided for third-party devices using the MIB Browser.

Application Interface

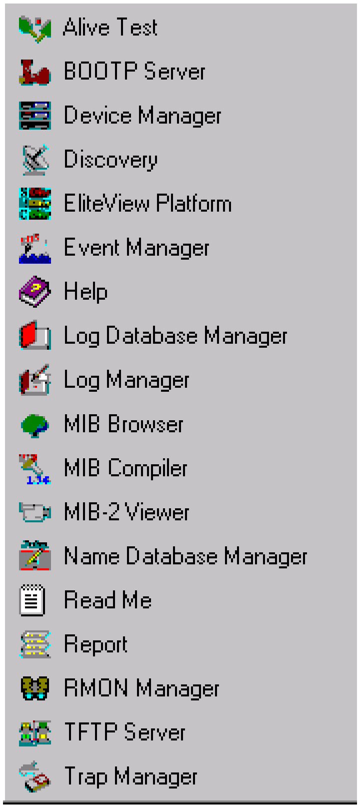
EliteView runs on a personal computer attached to the network you want to manage. Management actions normally occur via the network map, through which you can activate the appropriate software module simply by double-clicking on the concerned device or by selecting a target device and then invoking the appropriate module from the menus. By sending commands across the network, EliteView can directly manage a wide variety of SNMP-based devices.

Using this powerful management tool, you can generate a device map for a complete view of the network, where each device is represented as an icon. Network devices can be added or deleted manually, or located using Discovery. Device icons can be placed anywhere within a map using simple drag and drop. Object attributes can also be easily changed.

A full hierarchical representation can be generated by creating submaps that expand to a more detailed view when selected. Moreover, multiple submaps can be opened simultaneously. Each device included in the map can be checked periodically to verify that it is still attached to the network. When any device loses its network connection, its icon will change to indicate device state, and an alarm may be generated.

The standard method of starting EliteView is to double-click on the EliteView icon, open your network map, select a target device, and then invoke the required management module. However, you can directly invoke any of the modules displayed below.

The EliteView program group includes over twenty different modules. The main program, labeled EliteView Platform, serves as the platform through which you display the network map, manage the network, and access any of the other management modules.



Features of EliteView

EliteView includes the following features:

- Windows-based SNMP network management.
- Manages unlimited number of network devices running SNMP agent software.
- Provides detailed information on device parameters, such as statistics for the overall SNMP agent, device component status, and network interface configuration/statistics.
- Management controls are displayed with graphic and text-oriented windows, which can be accessed via the network map, or from a pull-down menu, for better functional grouping and a more intuitive user interface.
- Hierarchical, interactive network management map with unlimited devices and network levels.
- Displays real-time graphical statistics for various counters including network traffic. Monitors the status and traffic load of each attached device, e.g., displaying the number of incoming, outgoing and discarded frames.
- Flexible event management allows you to log relevant factors on device status and traffic.

CHAPTER 2

INSTALLATION

This chapter describes setup procedures for EliteView network management software (version 6.01). EliteView can manage any of SMC's network devices via standard and private MIB definitions; and also manage any third-party device that has a resident SNMP agent via standard MIB definitions. Installation of EliteView software designed to manage specific devices is covered in the corresponding manuals. (Refer to "Additional References.")

Installation for EliteView

EliteView can be readily installed on most Windows-compatible personal computers. The EliteView setup program will guide you through this step-by-step procedure.

System Requirements

Before installing EliteView, please review these minimum computer and network system requirements for a "dedicated" network management system (NMS).

Hardware:

- PC with an 80486DX2-66 CPU and 16MB memory
- 3.5-inch floppy drive and hard drive
- VGA adapter and display
- Mouse
- Network adapter

Software:

- Microsoft Windows 95
- Microsoft Windows 98
- Microsoft Windows NT 4.0
- Microsoft Windows 2000

If you frequently use multiple Windows-based applications, you may need a more powerful environment to run EliteView along with other applications. Otherwise, you may not be able to take advantage of EliteView's full range of management capabilities. Advanced system requirements include:

- Personal computer with a Pentium-133 processor (or higher)
- Local hard disk with a minimum of 40 MB free disk space
- SVGA color monitor with accelerated video adapter

Note: Although EliteView uses about 20 MB of disk space, additional disk space is required for user files.

Using SETUP to Install EliteView

The SETUP program will install EliteView from the distribution CD-ROM. This program decompresses files and copies them to a location you specify on your hard disk.

To Start SETUP:

1. Start Windows.
2. Insert the EliteView installation CD-ROM in your drive.
3. From the Program Manager, choose File menu and select Run.
4. Type D:SETUP in the dialog box.
5. Click on OK to start SETUP.
6. Follow the on-screen instructions to install the software.
7. Follow the instructions to setup the network interface.

Software Installation

EliteView files are placed in the default EliteView directory unless otherwise specified. Disk space required for the selected option is indicated at the bottom of the installation dialog box. During installation you will be prompted to assign the program group.

CHAPTER 3

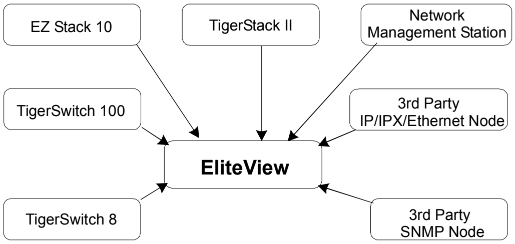
GETTING STARTED

Welcome to the EliteView network management program for Microsoft Windows. EliteView is a powerful network management product that provides detailed device management functions, along with a sophisticated graphic interface. The complete package is marketed as EliteView, which includes the main EliteView program and over twenty core program modules. This chapter provides an overview of the structure, and explains how the various EliteView modules are related.

Overview

EliteView is a flexible network management system based on international and industry standards. It is a Windows-based program that runs on an inexpensive PC platform. This full-featured network management software allows management of SMC or third-party network devices. In addition, it supports an open platform for the development of any kind of management application.

EliteView can manage various SMC network devices and examine the management information base in third-party devices.



Network Interface

EliteView can function on various network protocol stacks for greater flexibility and efficiency. In the Windows environment, EliteView can be configured to run on a Windows TCP/IP package with a standard WINSOCKET interface. At the application level, network devices are managed via SNMP over IP or IPX.

Event-Driven, Modular Architecture

Designed around an event-driven, multi-tasking architecture, EliteView consists of the main program and supplementary modules. Each module works independently or in conjunction with other modules. Running any module is as easy as invoking it from the main EliteView program, or by simply double-clicking on the corresponding icon from the EliteView program group.

EliteView Modules

EliteView includes support for many SMC products, including the EZ Stack, TigerStack, and TigerSwitch, series. This section briefly describes the basic support modules included in EliteView. The modules for specific SMC products are described in the on-line help files.

Alive Test

This module tests the connection to any network node with ICMP messages. It sends an echo request to the specified network node and gathers replies to determine device existence, round trip delay time, and the ratio of successfully returned packets.

BOOTP Server

The BOOTP Server maintains a database of network addresses and a list of corresponding boot files. BOOTP protocol runs on the UDP/IP stack. It is used by network devices to find out their own IP address and the device initialization file(s) to download.

Discovery

This module is used to automatically detect active devices on the network by polling within a specified network address range or community. Discovery sends commands out to the network and waits for responses. When a device responds, EliteView queries its functionality. If it responds correctly, a corresponding bitmap icon is added to the Discovery window.

EliteView Main Program

The main EliteView program provides the primary interface to all EliteView modules. The Tools and Utilities menus provide access to nearly every module under EliteView. The main program also provides all the tools you need to generate a detailed map of your network via menus and tool buttons. Moreover, it supports MDI (Multiple Document Interface) which allows you to simultaneously manage several submaps. After locating the required device on your map, simply double-click on it to invoke the relevant management application.

Event Manager

This module serves as the management center for all events generated under the main EliteView program, Log Manager and Trap Manager. In response to input from these modules, the Event Manager can define and dispatch responses in various forms. Actions may range from sounding an audible signal, displaying an on-screen message, logging the event into the report window, running a user-specified program such as a beeper, fax, pager, email, etc. or logging the event into a database for later analysis.

Log Utilities

The Log Manager can collect the current value of SNMP MIB variables at a specified interval. A wide range of parameters on device status and network traffic can be sampled for selected nodes and stored in the database for long-term analysis. This information is displayed with the Log Database Manager in numeric form (including date, time and data).

MIB Browser

This module is a generic SNMP management tool used to browse device MIBs. By browsing MIBs, you can send commands to get or set information defined in the MIB. You can also set up profiles which are used to fetch or set selected MIB variables as a group.

Information to be recorded into the Log Manager can be selected directly from the MIB. Moreover, the MIB Browser also provides a convenient editing tool which can be used to quickly extract information from the MIB and store it for future reference or prepare it as a technical report.

MIB Compiler

This application compiles textual MIB files into database files specifically formated for EliteView, which allows relevant EliteView modules to access required information.

MIB-2 Viewer

This module provides an easy-to-use windowed interface to the MIB II (RFC 1213) management information database. MIB II is maintained by each device that includes a resident SNMP agent. The traditional approach displays information directly from the MIB, which requires a good understanding of the overall hierarchial tree structure to locate the variables you need. The MIB-2 Viewer, on the other hand, organizes this information in a set of commonly referenced items which are displayed in a convenient and easily understood format.

Name Database Manager

This module provides a convenient means to map an easily remembered mnemonic name to each device in the network. These names are then used in many other EliteView modules, which allows you to conveniently specify any network device or view data using the name associated with each device.

Report

This module displays any system events or user-defined events specified in the Event Manager. The Report window shows all network alarm messages in chronological order. Each entry is stamped with a time and date.

RMON Manager

Remote Network Monitoring allows you to instruct a remote device to collect information or respond to specified events on an independent basis. An RMON-capable device can independently perform a wide range of tasks, significantly reducing network management traffic. It can continuously run diagnostics and log network performance. If an event is triggered, the remote device can automatically notify the network administrator of a failure and provide historical information about the event. If the remote device cannot connect to the management agent, it will continue to perform any specified tasks and pass data back to the management station the next time it contacts the remote device.

TFTP Server

This module is used to download agent software to the requesting device. It can be used to download software to any EZ Stack, TigerStack, and TigerSwitch management module. For all other SMC devices, downloading is performed via out-of-band mode. The TFTP server is also used to perform file transfers between any two stations running EliteView.

Trap Manager

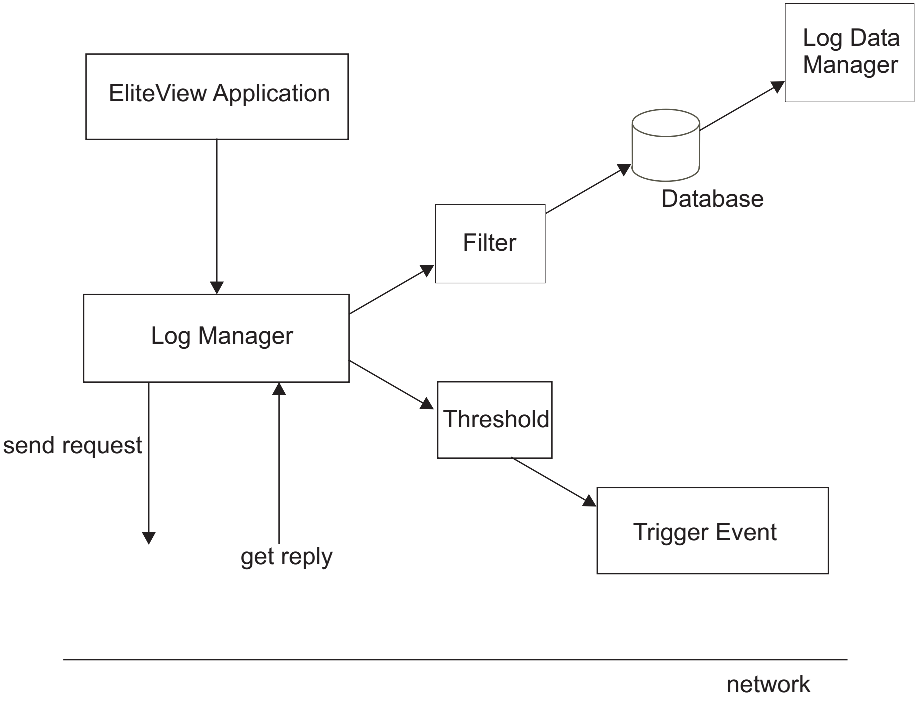
Trap Manager has no tangible user interface. It is only used to pass events to the Event Manager. This module receives trap messages and converts them into events. By default, the Trap Manager generates a "Trap" event and outputs a text message to the Event Manager according to the pattern specified in TRAP.INI. These events are then handled according to the options selected in the Event Manager.

Network Device Management Modules

Each of these modules supports advanced management functions for the corresponding device. On-line help is provided for each of these devices. EliteView currently includes device management modules for the EZ Stack, TigerStack and TigerSwitch series.

Data Logging and Event Management

The Log Manager and Event Manager modules play a key role in network management. The following diagrams depict how they work.

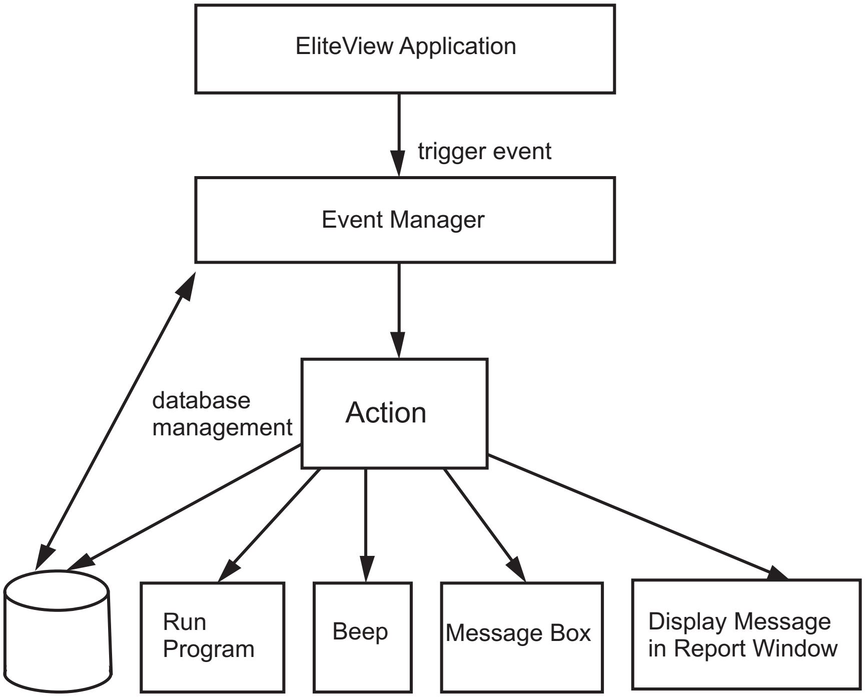


The Log Manager periodically sends requests to target devices according to a fixed polling interval. The target device receives the requests and send replies to the Log Manager.

The Log Manager then processes data in two different ways:

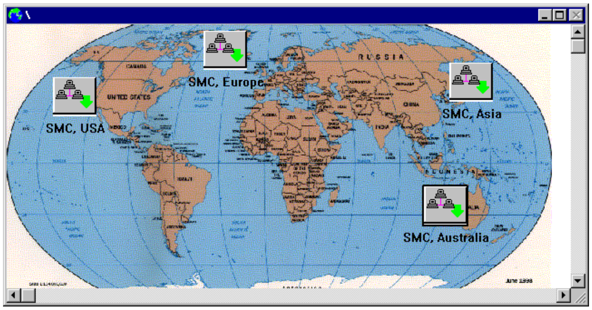
- Data passes through the filter you set; it may be saved in the database depending on the condition specified in the filter.
- Data is checked against a threshold formula. If conditions are satisfied, the Log Manager automatically triggers the event associated with the log process.

How the Event Manager Works



The Event Manager receives input from EliteView applications such as the main EliteView program, the Log Manager and the Trap Manager. Any named event may be triggered simply by satisfying the user-defined threshold formula. Any triggered event is passed on to the Event Manager, which activates the proper response, such as running a program, sounding an audible alarm, displaying a message on screen, displaying a message in the Report window, or writing to the event database.

Starting EliteView

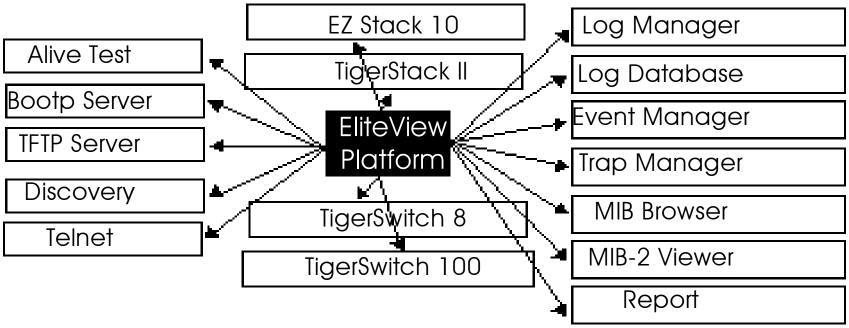


The main EliteView program provides an intuitive interface to other program modules. You can invoke specific management applications (by clicking on the appropriate device icon in the network map), verify current network connections with Discovery, check device response (via broadcast/search) with the Alive Test, or fetch information about selected devices using the MIB Browser.

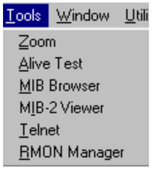
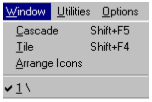
Each module is closely integrated with the main EliteView program, and can be quickly invoked by selecting the required function from the Utilities menus. For certain device-specific applications, first select a target device from the network map and then select the required function from the Tools menu. The overall relationship between the main EliteView program and its submodules is depicted below.

Using the Main EliteView Program

To invoke the main EliteView program click on the EliteView icon. Many functions under the main program provide support for network mapping. These functions include most of the toolbar buttons, along with the File, Edit and Windows menus. Network mapping is described in the next chapter, Defining the Network Configuration. After you map out your network, you will want to use the other functions listed below.



The menu items and tool buttons used on a regular basis are described below. The items used to construct and maintain your network map are described in the next chapter.

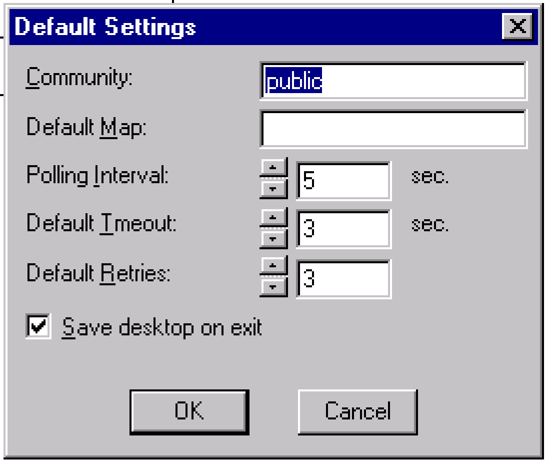
Menu	Label	Description
 Tools Window Utili Zoom Alive Test MIB Browser MIB-2 Viewer Telnet RMON Manager	Tools	**Zoom** - Opens the management module for the selected device. **Alive Test -** Opens the Alive Test for the selected device. **MIB Browser** - Opens the MIB Browser for the selected device. **MIB-2 Viewer** - Opens the MIB-2 Viewer for the selected device. **Telnet** - Opens a connection to another computer on the network through which you can execute programs or access data as though attached locally. **RMON Manager** - Provides access to to all nine RMON groups for recent SMC products that support RMON.
 Window Utilities Options Cascade Shift+F5 Tile Shift+F4 Arrange Icons ✓ 1 \	Window	**Cascade** - Arranges all open EliteView windows in cacaded fashion. **Tile** - Arranges all curretly open EliteView windows in tiled fashion. **Arrange Icons** - Arranges all EliteView icons on the screen. *select window* - Switches to the selected EliteView window.
(*not shown here*)	Utilities	Accesses most EliteView modules.
Options Help Change Settings... Change Password Map Infomation Search Object ✓ Toolbar ✓ Status Bar Config Toolbar...	Options	**Change Settings** - Allows you to define the default map, SNMP community, polling interval, default timeout, default retries, and whether or not to save the desktop when EliteView is closed. **Change Password** - Changes the password required to display the map. **Map Information** - Displays all user-defined parameters for each device included in the current map. **Search Object** - Searches for a network device by address or label (the later of which must be defined in the network map). **Toolbar** & **Status Bar -** toggle buttons to display or hide these items. **Config Toolbar** - Utility used to specify toolbar layout.
(*not shown here*)	floating	By clicking anywhere in the background of the EliteView program you can open a menu that includes various functions relevant to the current module.

Many of the items included in the menu bar are also provided in the toolbar. The following table describes a few of the buttons found in the main EliteView program. The other buttons, which are used for creating network maps, are described in the following chapter.

Button	Label	Description
	Exit	Shuts down the main EliteView program and all subordinate modules.
ⓘ	About	Displays the version number for this module.
?	Help	Provides on-line help for this module.

Configuring Polling Parameters

Before directly accessing devices from the network map, you should specifiy the default settings. These defaults are provided as a convenience for you, and are used by both the Add Object command and the device management modules. Open the *Default Settings* dialog box by selecting *Change Settings* from the Options menu. Set an SNMP



Community: public

Default Map:

Polling Interval: 5 sec.

Default Timeout: 3 sec.

Default Retries: 3

☑ Save desktop on exit

community name and polling parameters which are applicable for your particular network environment. Depending on your current configuration, you may need to provide any of the following information.

1. Define a **Community** describing the administrative relationship between SNMP entities.
2. Specify the **Polling Interval** between sending requests, the **Default Timeout** to wait for a response, and number of **Default Retries** to make contact. The settings displayed here (i.e., 5, 3, 3) should be suitable for most environments.

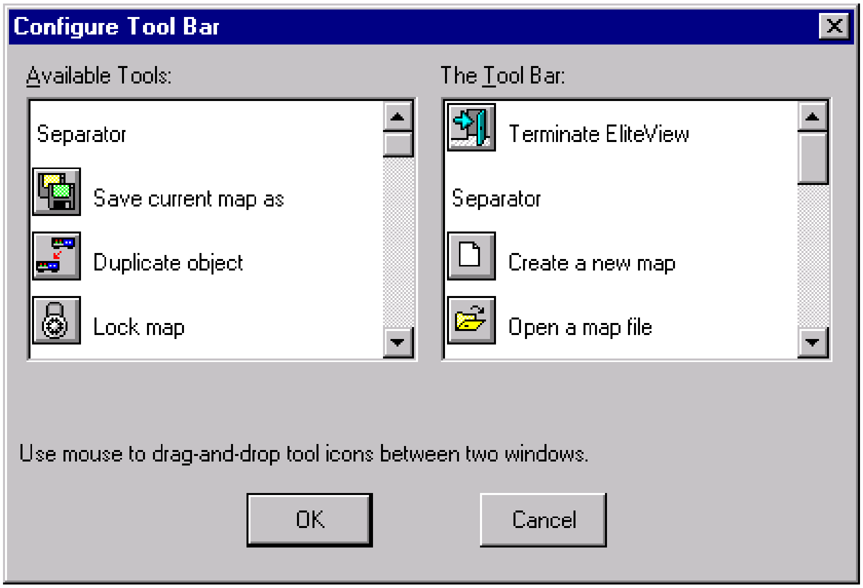
Status Bar

The Status Bar serves two basic functions. It displays the status of any currently executing command, and indicates the function of selected toolbar buttons. **To display the description for any toolbar button** (in the status bar at the bottom of the screen), position your mouse over the toolbar button, and hold down the left mouse button. After viewing the description, slide your mouse off the interface button (without releasing the mouse button). Note that descriptions for toolbar buttons are also provided in relevant sections throughout this manual.

Configuring the Toolbar

The main program and other modules include an option that allows you to modify the toolbar layout to suit your specific needs. Simply drag the icons you want from the *Available Tools* list into the required position in the *Tool Bar* list.

Select any object from the Available Tools list on the left and drag it into position in the Tool Bar layout on the right.



Use mouse to drag-and-drop tool icons between two windows.

CHAPTER 4

DEFINING THE NETWORK CONFIGURATION

Before running any EliteView device management tools, first define the device interconnection hierarchy, network addresses, and mnemonic names for each network node. If you don't already have this information mapped out, then use *Discovery* to help identify each device in your network. This technique may also be used periodically to incorporate changes in the network configuration.

After identifying the basic network configuration, use the Name Database Manager to assign easily remembered names to each network device. And finally, in the last step, create a detailed network map including all intermediate network hierarchy and subordinate devices. This map can then be used to quickly open relevant device management tools by simply double-clicking on a map object.

All the tools and techniques required to define your network configuration are described in this chapter.

Quick Guide to Map Building

The simplest approach to creating a network map is listed below:

1. Use Discovery to locate network devices.
2. Move selected objects from Discovery onto the map by either of the following methods -
   - Drag objects directly onto the appropriate map.
   - Use the Get Objects command to fetch objects based on protocol type from the queue of discovered devices.
3. Use the Add Object command to define additional symbolic objects, such as a LAN Segment or Submap.
4. Draw in network connections using the toolbar in the EliteView Platform program.
5. Save your map under an appropriate filename.

Command	Menu
Discovery	Utilities
Add Object(s) to Name Database	
Get Objects	Edit
Add Object	Edit
Modify Object	Edit
Draw Network Connection	
Save Map As	File

Discovery

EliteView can automatically discover any device using a specified protocol (i.e., SNMP over UDP/IP or IPX) by polling within a specified network community or address range.

EliteView's Discovery module sends commands out to the network and waits for responses. Devices are classified based on whether or not they have a resident SNMP agent. When a device responds, EliteView queries for SNMP functionality. If a device has no resident SNMP agent, EliteView adds a generic bitmap (to the window for discovered devices) based on protocol type. However, if a device has an agent then EliteView tries to identify the device type. If the device is recognized, it adds an object icon based on device type, otherwise it adds a generic bitmap to indicate that the object has an SNMP agent and to show the associated protocol type.

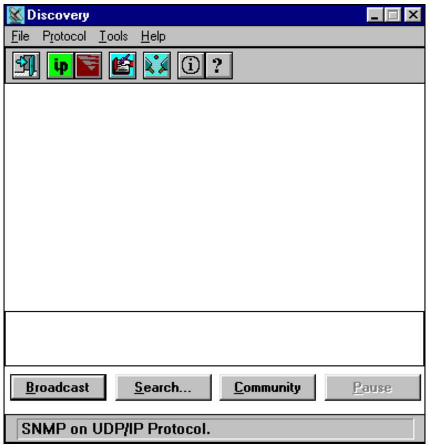
Using Discovery

Use Discovery to build your initial network map or to locate a specific device. After you have located the concerned devices, drag them onto a network map where they can be used to conveniently monitor your network.

You can select Discovery from the Utilities menu in the main EliteView program, or activate it by clicking on the *Discovery* icon in the EliteView program group. Specify the required protocol and then use *Broadcast* or *Search* to locate attached devices.

Using Discovery, EliteView can automatically identify responding devices and label them with the correct address (based on the selected protocol). This indicates the status of current network connections, and also serves to validate the accessiblity of devices for subsequent management. Once a device has been found, simply drag it onto a map and then initiate the relevant management module by clicking on the device icon.

When looking for devices using broadcast, you may need to press the *Broadcast* button several times to ensure that all attached devices have responded. For nodes that do not respond to broadcast queries, use the *Search* function. In general, it may be necessary to search for devices not located in the same network with the EliteView management station.



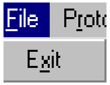
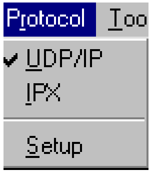
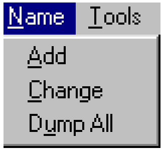
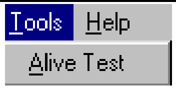
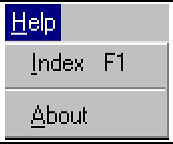
1) Select the appropriate network protocol.
2) Broadcast or Search for attached devices.
3) And drag key devices onto the configuration map for later use.

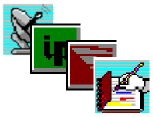
SNMP on UDP/IP Protocol.

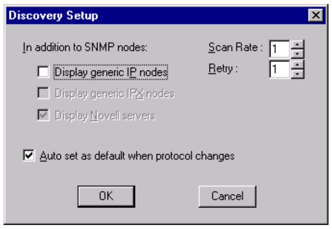
Menu Description

The menus provided for Discovery are briefly introduced below. Toolbar buttons are also described in this section.

Menu Definitions

Menu	Label	Description
	File	**Exit current module -** Closes the Discovery window and exits to the calling program (i.e., EliteView platform or Windows Program Manager).
	Protocol	**Protocol selection and advanced settings** Select the devices to display based on network protototocol - *UDP/IP* - IP & ICMP, SNMP over UDP/IP agent *IPX* - IPX, SNMP over IPX agent *Setup* menu - Toggles display of devices without an SNMP agent; also sets the scan rate and retry count.
	Name	**Name database management -** Provides editing functions for the name database, including adding or modifiying object data. You can also dump all the information gathered by Discovery into the Name database.
	Tools	**Alive Test -** The *Alive Test* is used to directly test device response.
	Help	**Help and Version information -** Provides detailed on-line help and displays the version number for this software module.





Discovery Setup

In addition to SNMP nodes:

☐ Display generic IP nodes

☐ Display generic IPX nodes

☑ Display Novell servers

Scan Rate : 1

Retry : 1

☑ Auto set as default when protocol changes

To automatically discover devices:

1. From the Edit menu, choose Discovery.
2. Choose the appropriate network protocol.
3. If you need to change the search criteria for devices, open the Setup dialog box using the toolbar.

Field Description for Discovery Setup Menu

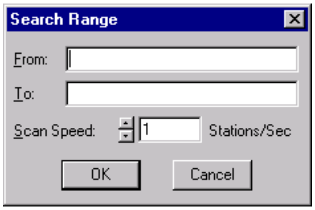
Field	Description
Display IP Node	Display IP Nodes without an SNMP agent.
Display IPX Node	Display IPX Nodes without an SNMP agent.
Display Novell Server	Display Novell Servers.
Scan Rate	The scan rate between broadcast requests.
Retry	The number of times to query for device response.
Auto Set as default when Protocol changes	Save the current setup as default when restarting Discovery.

For TCP/IP Nodes -
1) Broadcast within same network.
2) Search for nodes across routers.
3) Many IP nodes without SNMP can respond to ICMP queries.

For IPX Nodes -
1) Broadcast is sufficient.
2) Search is not available.
3) NetWare Servers use IPX protocol.

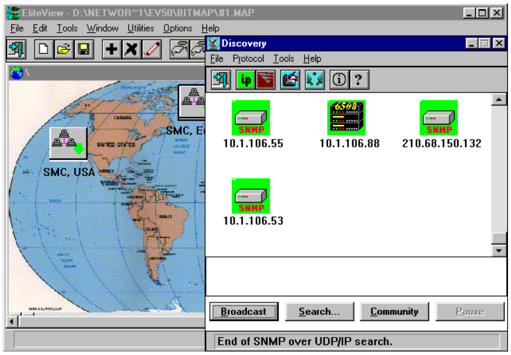
4. Click on Broadcast to transmit a query message and wait for responses from the local network. Broadcast is also adequate for gathering global responses from IPX or Ethernet nodes located on different networks.

   However, to find IP nodes on other networks, use the Search command and provide a specific address range. If you are only interested in a specific range of stations or need to search for stations that are difficult to reach, then click on the Search button, specify the address range, and adjust the scan rate if required. Discovery will search for devices within the specified range.



Scan Speed: 1 Stations/Sec

5. The status line at the bottom of the Discovery window shows the status of the search. To temporarily halt the process, press the Pause button; press Resume to continue.
6. The message End of Search will appear when the discovery process is completed.
7. Move selected objects from Discovery onto the map by either of the following methods -
   - Drag objects directly onto the appropriate map.
   - Use the Get Objects command (under the Edit menu) to fetch objects based on protocol type from the queue of discovered devices.
8. Draw in the existing connections and save the updated map. (Refer to Creating Network Maps for detailed information on building a network map.)

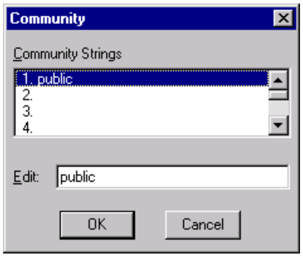


End of SNMP over UDP/IP search.

Adding Community Strings to Discovery

Discovery will search for devices within the specified communities. To include any communities other than “public,” add the appropriate name to the *Community* dialog box. You can include all the communities defined for your network in a broadcast or search command. However, be aware that a blind search creates excess traffic. Using a more conservative search will have less impact on network performance.

Insert access control string for authorised community members



1. public
2.
3.
4.

Edit: public

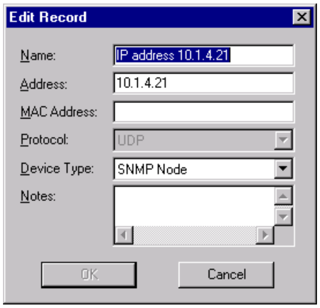
To automatically discover devices in a community:

1. Click on *Community* to display a list of community strings.
2. To add or modify a community string, click on an entry in the list and edit the entry in the *Edit* field. Click *OK* to continue or *Cancel* to abandon the new entry.
3. Click on *Broadcast* to begin searching for devices.

Exiting Discovery

To close this module,

1. From the File menu, choose *Exit*.
2. The Discovery program will be closed.



Name: IP address 10.1.4.21
Address: 10.1.4.21
MAC Address:
Protocol: UDP
Device Type: SNMP Node
Notes:

Name Database Manager

Device names assigned with the Name Database Manager are used in many other EliteView modules to help you readily select or identify network devices. The Name Database Manager can be opened from the Utilities menu under the main EliteView program, or directly from the EliteView group window by double-clicking on the icon shown here.

Also refer to the discussion on Updating the Name Database under the section on Discovery.

You can store information about network devices in the name database, including a name, network address, physical address, network protocol, device type, and informal notes. For normal maintenance, we recommend updating this information via Discovery. However, when you need to view the entire database, you can use the Name Database Manager.

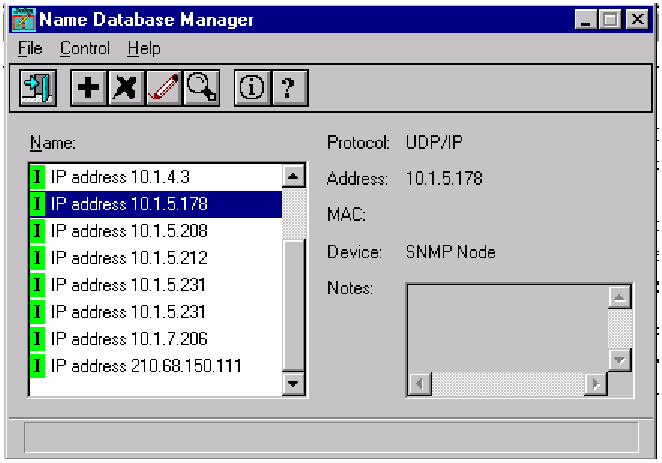
All items included in the menu bar (under the File, Control and Help menus) are also provided in the toolbar. This table describes the basic editing tools.

Adding a New Entry

Click on the Add Record button to open this dialog box. Enter a suitable menmonic name, the network address (i.e. UDP/IP or IPX), the physical device address, the network protocol (i.e. IP or IPX), the functional device type, and any informal notes. Rememeber to indicate specific device types for SMC network devices and generic designations for all other network devices. (Also note that the Get MAC function is only enabled for the UDP/IP protocol under Windows 98, Windows NT 4.0 Service Pack 4 or later, and Windows 2000.)

Deleting Device Entries

Use your mouse to highlight the entries you want to delete and press the delete button. (Note that the same conventions are followed for selecting multiple items as those used by the Windows File Manager.)



Name:

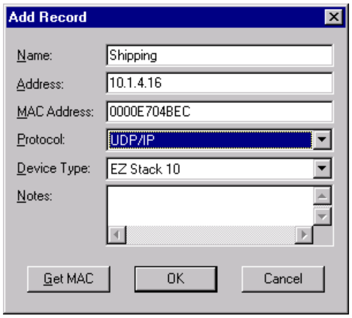
- IP address 10.1.4.3
- IP address 10.1.5.178
- IP address 10.1.5.208
- IP address 10.1.5.212
- IP address 10.1.5.231
- IP address 10.1.5.231
- IP address 10.1.7.206
- IP address 210.68.150.111

Protocol: UDP/IP
Address: 10.1.5.178
MAC:
Device: SNMP Node
Notes:

Updating Device Entries

Button	Label	Description
➕	Add Record	Adds a new entry into the name database.
✖	Delete Record	Deletes an entry from the name database.
✎	Update Record	Updates a current entry in the name database.
🔍	Search Records	Searches for specified record(s) in the name database

Use your mouse to highlight the entry you want to update. Then change any of fields in the dialog box as described above under Adding a New Entry. Note that when device information is dumped from Discovery, the Name field is automatically filled in using the network address. Therefore, you should update each entry to include a meaningful name for these devices. You can do this directly from Discovery, but it may be easier using the Name Database Manager, where the entire list of devices is readily availble for reference.



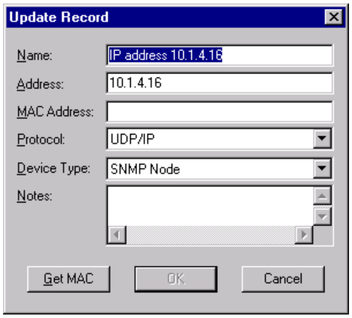
Name: Shipping
Address: 10.1.4.16
MAC Address: 0000E704BEC
Protocol: UDP/IP
Device Type: EZ Stack 10
Notes:

Searching for Device Entries

You can easily display all device entries that meet your specified search criteria. Simply click on the Search button and fill in the following parameters:

- Search Mode - ALL or MATCHED
- Sort Key - Device Type, MAC address, Name or Protocol
- Sort Order - Ascending or Decending
- Search Keys - Name, Address, MAC Address, Protocol and Device type

If you select a search mode of ALL, then the search keys are disabled and all entries in the name database will be displayed. If you select MATCHED search mode, you should include the device name, network address, physical MAC address, protocol type, or device type, or any combination of these. Also remember that the search for device name is case sensitive.

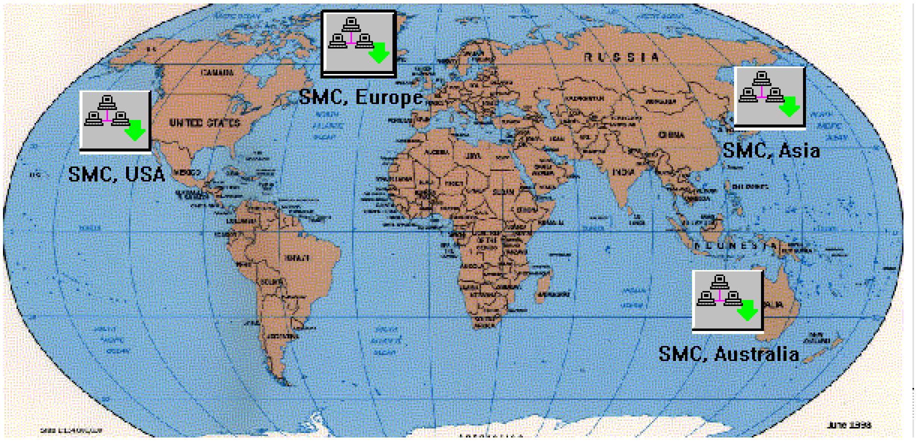


Name: IP address 10.1.4.16
Address: 10.1.4.16
MAC Address:
Protocol: UDP/IP
Device Type: SNMP Node
Notes:

Creating Network Maps

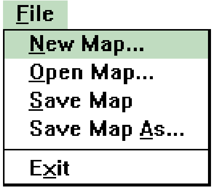
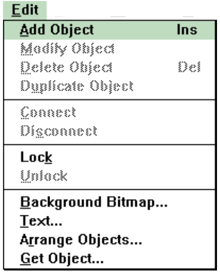
Network maps are like road maps, they visually depict the entire network hierarchy. Network administrators use them to trace out the connections between various network devices, and to quickly activate dedicated management tools for a selected device. These maps describe the status of network devices, their physical location, and their logical organization. This section describes how to create, maintain and use network maps.

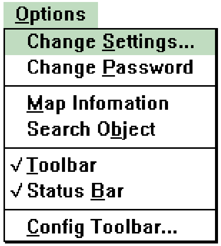
You can organize network maps using any number of hierarchial levels.



The main EliteView program is used as the primary interface to most of the EliteView modules. However, this is also where you create and edit network maps. The following menus are used for map functions.

Menu Description for Map Functions

Menu	Label	Description
 File New Map... Open Map... Save Map Save Map As... Exit	File	**New Map** - Initializes required parameters to create a new network map. **Open Map** - Opens an existing network map. **Save Map** - Saves the current map along with any changes. **Save Map As** - Saves the current map under a new name. **Exit** - Exits EliteView, closing all subordinate modules.
 Edit Add Object Ins Modify Object Delete Object Del Duplicate Object Connect Disconnect Lock Unlock Background Bitmap... Text... Arrange Objects... Get Object...	Edit	**Add Object** - Adds a new object to the map based on a textual description. **Modify Object** - Allows you to modify any parameters for a map object. **Delete Object** - Deletes the selected map object. **Duplicate Object** - Copies the selected object, after which you should reposition it on the map and modify any required parameters. **Connect** - Draws in a device connection from the currently selected object to the next object you click on. **Disconnect** - Removes a connection from the currently selected object to the next object you click on. **Lock** - Prevents any changes from being made to the current map. **Unlock** - Allows changes to be made to the current map. **Background Bitmap** - Selects a bitmap file to use for the map background. **Text** - Allows you to edit text that will be displayed on the map. **Arrange Objects** - Orders objects in the map according to your specifications. **Get Objects** - Moves objects from queue of devices (found by Discovery) onto the network map based on selected protocol type or network identifier.

Menu	Label	Description
 Options Change Settings... Change Password Map Infomation Search Object √ Toolbar √ Status Bar Config Toolbar...	Options	**Change Settings** - Allows you to define the default map which will be automatically displayed every time you open the main EliteView program. **Change Password** - Changes password required to display the current map. **Map Information** - Displays all user-defined parameters for each device included in the current map by means of the Report Program. **Search Object** - Locates specified object within map based on label or address.

Many of the items included in the menu bar are also provided in the toolbar. The following table describes these basic tools.

For information on menu items or buttons that appear under the main EliteView module but are not described in this section, refer to “Using the Main EliteView Program” in the preceeding chapter.

Button	Label	Description
✚	Add New Object	Adds a new object to the current map.
✖	Delete Object	Deletes an object from the current map.
✏	Modify Object	Modifies the description for an object.
	Connect Object	Connects the selected object to another object.
	Disconnect Object	Disconnects the selected object from another object.

Editing Map Objects

The Edit menu provides all the tools you need to compose a full-scale hierachical map of your entire network. You may add or modify object descriptions, draw in physical connections, specify a bitmap to display as the background image for your map, edit any labels or legends required for the map, and then lock it to prevent further modification.

Adding a Map Object

The best approach to adding a map object is to first locate the target device using Discovery, drag the object onto your network map, and then enter the additional information using the Modify Object function. However, you may also add a new object based entirely on the textual description provided under the Add New Object dialog box.

1. Click on the *Add New Object* button (or press <Insert>).
2. Select an object type. Specific types are provided for SMC network devices. EZ Stack, TigerStack and TigerSwitch devices are SMC products. For all other network entities, select the appropriate generic device type as described below.

Device Type	Description
IP Node	Any device connected via IP network protocol.
LAN Segment	A network backbone (i.e., view-only object).
SNMP Node	Any network device that supports general SNMP/UDP/IP or functions.
Submap	A “hot link” to a submap.

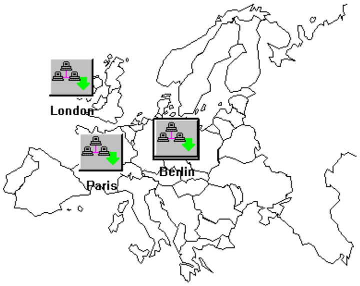
3. Fill in the dialog box with the object attributes described below.

Field Definitions for the Add Object Dialog Box

Attribute	Description	Example
Label	Enter up to 16 characters. This label appears below the device icon.	Mktg001, Bldg1020, Hub42.32
Address	The network address of the device. Enter in IP, IPX or Ethernet address notation, depending on selected protocol type, where IP - internet address and IPX - network number : node number.	IP: 192.72.24.05 IPX: 000ACC01: 000000000001
Community	The community string used to access the device. This text string must match the community string stored in the device. (An incorrect community string will prevent access to the device.)	public
Protocol	The network protocol of UDP/IP or IPX.	UDP
Polling Interval	The interval between polling (in seconds). Setting a low value (2 seconds or less) will generate excessive network traffic and make EliteView seem very slow and unresponsive. While a very high value will make EliteView insensitive to changes in device status.	5
Timeout	After sending an SNMP request, EliteView will wait for an appropriate response (in seconds). If the device does not respond before the specified timeout, EliteView will assume that the device is no longer accessible.	3 (values greater than 3 are supported, but not recommended)
Retries	When a device does not respond within the *Retries* limit, the device is assumed to be off-line. The event "Connection Lost" is announced and the icon turns red. EliteView will continue polling for responses (unless *Monitor* is turned off, as described in the next parameter).	10
Monitor	If this box is checked, then it will be polled at the specified time interval. If this box is not checked, then polling is disabled. Note that the resources of the network management station may become overtaxed if you attempt to monitor an excessively large number of stations.	Yes (checked)

Sample Configuration

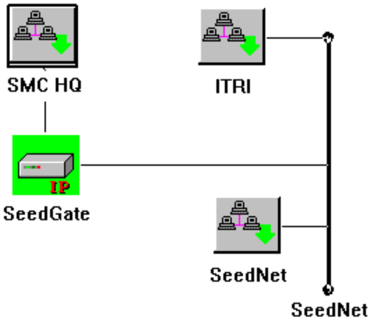
For large networks, you should break the map up into several pieces that can be opened independently. The following figure shows an example of the network map for our offices in Europe. The submap icons are logical links to other maps. In larger networks, you can represent the overall configuration as logical network segments. To view a subordinate map, double-click on the corresponding submap icon. The display shows the path name for the selected submap in the title bar.



Submaps may also be used to view different maps of the same network. For example, you might use several maps showing -

- "\Network" as the physical view
- "\Organization" as the logical network hierarchy
- "\Offices" as the network organized by offices

At the next lower level you can depict the network backbone for the selected network. A LAN Segment is a special kind of view-only object that is designed to show a common backbone on a network. The LAN segment is also commonly used to depict various networks based on different protocols (e.g., UDP/IP or IPX) connected to a common backbone.



To position the LAN segment on your map, you can stretch it by dragging the length handles longer or shorter, or angle it by rotating the end points.

At the lowest level, place the actual network devices and draw in all interconnections. This gives you an accurate picture of the network and also lets you activate applicable management software modules. For SMC's nework devices, the corresponding device management module will be activated when you double-click on the device icon.

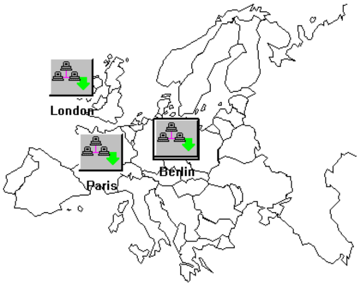
Modifying Objects

1. Select a network device by clicking on its corresponding icon.
2. Click on the *Modify Object* button.
3. Modify any of the required parameters. Refer to the table on "Field Definitions for the Add Object Dialog Box." Be sure to follow the guidelines in this table when setting the Polling Interval or Timeout.
4. Press OK to continue, or Cancel to abort any changes.

Deleting Objects

1. Select any map object by clicking on its icon.
2. Click on the *Delete Object* button, or press the Delete key on your keyboard.

If an icon cannot be deleted, the map view has been locked. From the Edit menu, choose Unlock and repeat step 2 again.

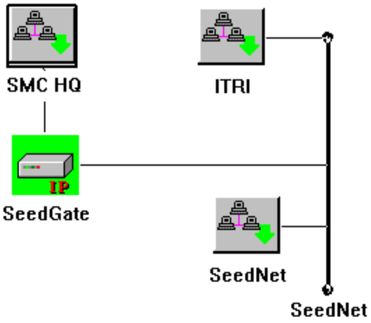


Duplicating Objects

1. Select any map object by clicking on its icon.
2. From the Edit menu, choose Duplicate Object. A copy of the object appears in the upper left corner of your map.
3. Drag the duplicate object to its new location and draw in any corresponding network connections.
4. For most applications, you will want to modify the object definition. Therefore, click on the Edit Object button and modify field parameters such as Label and Address.

Moving Objects

Use "drag and drop" to move an object. Select any map object by clicking on its icon. Holding the left mouse button down, move the outline of the device icon to its new location, and release the mouse button. If an icon cannot be moved, the map view has been locked. From the Edit menu, choose *Unlock* and try again.

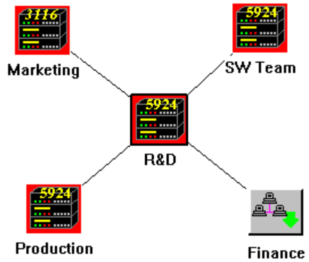


Tip: When creating multiple views of the same network, use *Duplicate Object* to make copies of objects, use the mouse to "drag and drop" icons to a new location or another submap, and then use *Edit Object* to modify the object's description.

Object Status

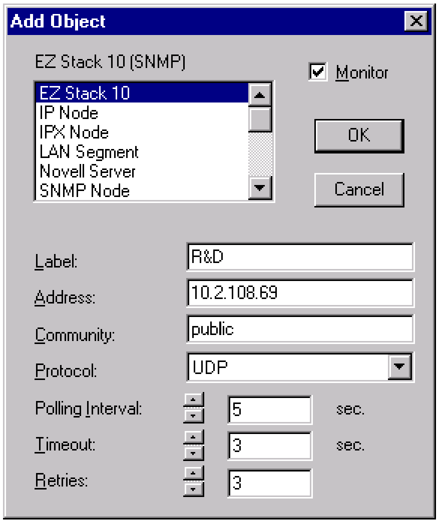
When an object is first added to the map, the device is “offline.” If an object's monitor flag is enabled, EliteView will begin polling the device immediately. When the object first responds, device status changes to “online” and a “device up” event is generated. If an object fails to respond within the specified timeout and retry limits, device status changes to “offline” and a “connection lost” event is generated. Changes in the status of objects at lower hierarchial levels in the network map are also reflected in the icons at higher levels (i.e., submap icons change color to reflect changes in subordinate devices).

Note: To post changes in object status to the Report window, specifiy *Device Up* or *Connection Lost* in the Action list for the Event Manager. For more information on defining event responses, see Managing Events, Chapter 8.



Map Limitations

EliteView maps are designed without any arbitrary program limitations. Practical limits are set by the available system resources.



EZ Stack 10 (SNMP)

☑ Monitor

Label: R&D

Address: 10.2.108.69

Community: public

Protocol: UDP

Polling Interval: 5 sec.

Timeout: 3 sec.

Retries: 3

CHAPTER 5

NETWORK TOOLS

EliteView supports a wide range of network tools that can be accessed directly from the device maps (see Creating Network Maps, Chapter 4) or from the Window's Program Manager. This chapter describes utilities designed to allow a device to identify its own IP address, to help the network manager verify the existence of a device in the network, and to update device software over the network.

Setting Addresses with the BOOTP Server

BOOTP is a protocol (running on the UDP/IP stack) used by network devices to find out their own IP address and identify files which are to be subsequently downloaded to the client device. Typically, IP addresses are assigned manually by the network administrator and recorded in the device's permanent storage for ready reference.

For many network sites, managing IP addresses can be a chore. The network manager needs a convenient way to access every device and dynamically assign its logical address from a central location. Since every device has a distinct physical network address, a server can run a special network protocol that lets each device lookup its own IP address based upon its physical address. Although there are many different address assignment protocols, BOOTP is one of the most popular ones.

The BOOTP Protocol

The complete BOOTP protocol provides a wide variety of information services. However, EliteView's BOOTP Server only provides an IP address for device recognition and a filename for subsequent downloading.

Once a request is received, the BOOTP server uses the client station's physical address as a key to find the client station's IP address. It replies with the corresponding IP address and a path/ filename for a generic or specific device initialization file.

Here is how EliteView's BOOTP protocol works:

1. A client station needs its IP address or filename information.
2. The client station sends a BOOTP request. Since it may not know its own IP address at this time, it may send out a request via broadcast.
3. The EliteView BOOTP Server receives the request and uses the client station's physical address as a key to lookup the client station's IP address. Next, the BOOTP Server looks up the filename for the client station.
4. The BOOTP Server sends a reply message back to the MAC address initially provided by the client.

A client station may frequently lookup a filename with BOOTP. For example, a filename may be needed by the client station to download operating system software from a dedicated file server using another protocol (e.g., TFTP). Since EliteView also provides a TFTP Server, the service request can be completed entirely via EliteView.

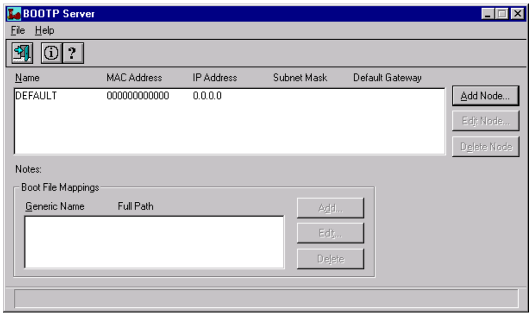
If the client station provides a generic name, such as "unix" or "hubware," the BOOTP Server will reply with the corresponding filename in the server. This allows multiple file download services for many kinds of devices. If the client station does not provide a generic name, the BOOTP server returns the *DEFAULT* generic filename.

Starting the BOOTP Server

To open the BOOTP Database,

Choose BOOTP Server from the Utilities menu in the main EliteView program, or directly from the EliteView program group.

The dialog box for the BOOTP Server will display.



Name	MAC Address	IP Address	Subnet Mask	Default Gateway
DEFAULT	000000000000	0.0.0.0		

The screen has two parts. The upper half is the list of address mappings, while the lower half is the list of file mappings. Each client station has the following attributes:

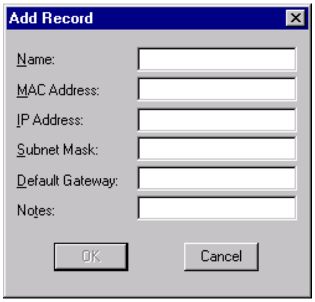
Parameters	Description	Example
Name	Node name. This value is not used by BOOTP, but is useful for the network manager.	NetWareSV 2
MAC Address	Physical address of this node.	0000e80a3e9c
IP Address	IP network address	192.219.74.32
Subnet Mask	This mask identifies the host address bits used for routing to specific subnets.	255.255.0.0
Default Gateway	The gateway must be defined if the device is not located in the same IP segment as the BOOTP server.	10.1.0.254
Note	A short memo field.	SNMPDRV2.BIN
Boot File Mappings	List of filename mappings. Generic names map to an actual filenames.	

Select an entry in the node list to display the Note and filename mapping for the selected node.

Each node in the BOOTP Server can have its own filename mapping list. This provides maximum flexibility for the network administrator. Default mapping is also provided, where in most cases per-node special mapping is not required. Click on the *DEFAULT* name to set both the IP and MAC addresses to zero; this represents the default file mapping list.

To exit the BOOTP server, click on the *Exit* button.

Adding and Modifying Node Information



The BOOTP Server starts with a single entry called DEFAULT. No file mapping is initially defined for this entry. To provide BOOTP service based on generic information, add the entries you require to the file map for the DEFAULT node. To service specific nodes, enter data for each node.

To add a new node:

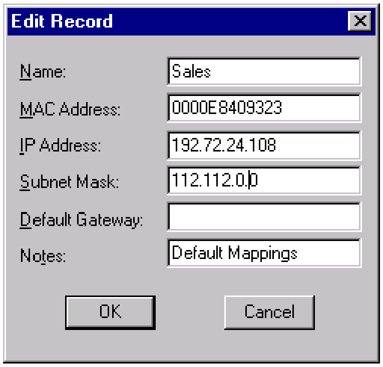
1. Click the Add Node button in the BOOTP Server dialog box. This will bring up the Add Record dialog box.
2. Enter the following information in the record fields.

Parameters	Description	Example
Name	Node name. Enter up to 14 characters.	NetWare Server 2
MAC Address	Physical address of the device.	0000E8C0A301
IP Address	IP network address.	192.255.74.32
Subnet Mask	This mask identifies the host address bits used for routing to specific subnets.	255.255.0.0
Default Gateway	The gateway must be defined if the device is not located in the same IP segment as the BOOTP server.	10.1.0.254
Note	Enter a note of 100 characters or less.	Backup file server at Bldg 100-34.4

3. Click OK to accept the changes or Cancel to abort them. New node information appears in the node address list.

To modify a node:

1. Click on the required node in the node address list.
2. Click the Edit Node button.
3. Make any changes to the current information.
4. Click on OK to accept the changes.


Edit Record

Name: Sales

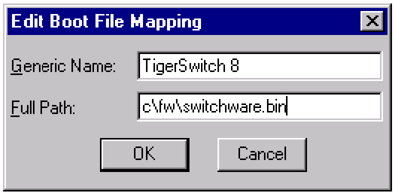
MAC Address: 0000E8409323

IP Address: 192.72.24.108

Subnet Mask: 112.112.0.0

Default Gateway:

Notes: Default Mappings



Generic Name: TigerSwitch 8
Full Path: c:\fw\switchware.bin

To delete a node:

1. Select a node entry by clicking on its address entry.
2. Click on the Del Node button.

The node entry and all associated file mappings are removed.

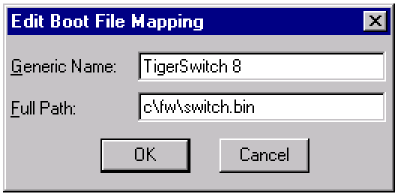
Adding Filename Mappings

The file mapping list shows every node defined.

To add a filename mapping:

1. Click on an entry in the node address list.
2. Click the *Add...* button in the File Mappings field to insert a new file map entry.
   - The *Add File Mapping* dialog box will appear.
3. Input the generic name and filename, including the path.
4. Click *OK* to store the new mapping.
   - The new entry will appear in the file mappings list.

To view the note and file mapping(s) for a node, click on the required entry in the address list.



Generic Name: TigerSwitch 8
Full Path: c:\fw\switch.bin

To change filename mappings:

1. Select the file mapping.
2. Click on the *Edit* button.
3. Change information in the Edit File Mapping dialog box.
4. Click *OK* to accept changes.

To delete a file mapping:

1. Select the file mapping.
2. Click on *Del.*

The filename mapping will be removed.

Default Information

EliteView's BOOTP Server provides flexible filename mapping. However, you may find it most convenient to establish a common default for most nodes on the network.

To define a default address with IP and physical addresses,

1. Select the "default" address (0.0.0.0).
2. Define file mappings applicable to all nodes on the network.

Every BOOTP request to lookup a filename will be checked in this priority:

1. Consult the *specific* node address.
2. If no address is found for the specified node, consult the *default* file mappings.

In addition to the explicit default file mappings, the BOOTP Server also provides implicit default file mapping. When a node is included in the address list and the client station provides no generic filename, it is asking for a default file mapping that you must provide. A *DEFAULT* filename must be defined for all stations requiring this type of mapping. If a *DEFAULT* generic name is not defined, the request is ignored.

Probing Devices with the Alive Test

The Alive Test serves as a basic network monitor. It determines link status by sending packets between the network management station (i.e., your PC) and the target node (e.g., gateway, hub or node). This test can be initiated from the Tools menu in the main EliteView program, from the Discovery module, or directly from the EliteView program group.

The Alive Test can be used with any IP or IPX device - including devices that do not support SNMP (like gateways). It cannot be used with the Ethernet protocol. To discover the existence of a device, the Alive Test uses "ICMP echo" for UDP/IP networks, and the IPX Diagnostic command for IPX nodes. If a device responds correctly, it returns the message to the sender. When the echoed message is received by the sender, it can determine:

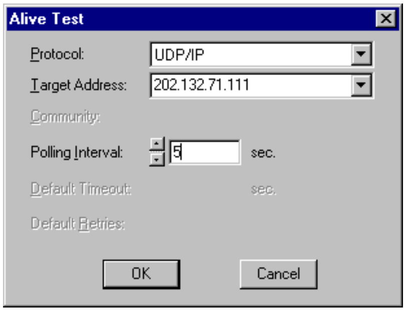
- Existence of the target device
- Round trip delay time
- Relative network throughput (transmission speed, etc.)
- Return ratio (percentage of packets correctly returned)

To select a target device,

If you open the Alive Test from the EliteView program group or from the Discovery module, select the network protocol as UDP/IP or IPX. Then specify the target address and polling interval.

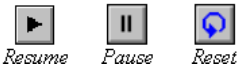
However, if you activate the Alive Test from within the main EliteView program then follow these steps -

Initialization window displayed when opening the Alive Test from the EliteView program group or from Discovery.



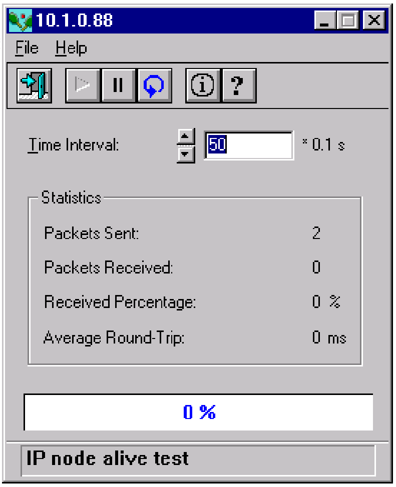
Protocol: UDP/IP
Target Address: 202.132.71.111
Polling Interval: 5 sec.

1. Open a map containing the target device.
2. Select the device with your mouse.
3. From the Tools menu, select Alive Test.



Note that network protocol, target address, and polling interval default to the object description as defined in the EliteView map.

In a few seconds, a dialog box opens showing device status.



10.1.0.88

Time Interval: 50 * 0.1 s

Statistics

Packets Sent:	2
Packets Received:	0
Received Percentage:	0 %
Average Round-Trip:	0 ms

0 %

IP node alive test

To adjust parameters for the Alive Test:

- Adjust the Time Interval by clicking the up/down arrow to increase or decrease its value. Time interval is the duration (in seconds) between the transmission of query messages from your PC, the network management station (NMS).
- Click Pause to halt the Alive Test temporarily.
- Click Resume to re-start the Alive Test.
- Click Reset to clear all the statistics.
- Click Set to specify a gateway for the target device.
- To exit the Alive Test, click on Exit.

The Alive Test collects a number of statistics. These include:

Statistics	Description	Examples
Packets Sent	Number of messages sent by Alive Test.	7
Packets Received	Number of echoed messages received by Alive Test.	6
Received Percentage	Ratio of messages received to messages sent.	85%
Average Round Trip	Average time interval between the original message sent and the echoed message received	900 ms

Problem Solving with the Alive Test

The Alive Test helps determine a number of network conditions.

1. Symptom: No response with the Alive Test.

   Condition:
   No response is ever received during an Alive Test.
   (packets received is 0).

   Possible Cause:
   The device does not exist or there is a cabling problem between your network management station and the target device.

2. Symptom: No response with EliteView (main program), but responds to the Alive Test.

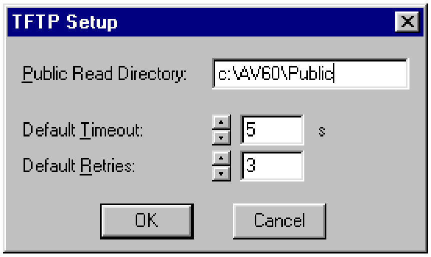
   Condition:
   A device responds to Alive Test (packets received is not 0).

   Possible Causes:
   a. The target device does not support SNMP.
   b. The community string for the target device does not match the setting for the object in the EliteView map.

Downloading Files with the TFTP Server

Network devices frequently include embedded firmware (software stored in ROM or flash memory) required for their operation. For example, SMC's EZ family of manageable devices (e.g., the EZ Stack 10) all include memory for an SNMP agent.

The trivial file transfer protocol (TFTP) is the most common standard for downloading files to network devices. SMC uses TFTP to download files for most of its manageable networking products.



TFTP Setup

Public Read Directory: c:\AV60\Public

Default Timeout: 5 s

Default Retries: 3

Starting the TFTP Server

To start the TFTP Server, choose TFTP Server from the Utilities menu in the main EliteView program or directly from the EliteView program group.

Using the TFTP Server

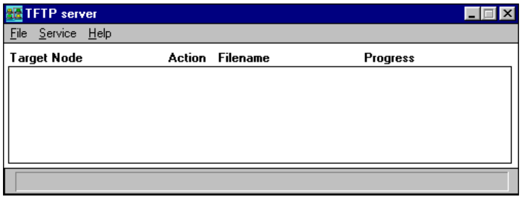
EliteView's TFTP Server provides a public directory for general downloading. The default directory is C:\EV50\PUBLIC. Only files in this directory can be downloaded to a target device or transfered to another server.

To configure the TFTP server, choose Setup from the File menu. The TFTP Setup dialog box will open, displaying options for the download directory, the default timeout to wait for a service response, and the default number of retries before terminating a connection attempt as described below.

Parameters	Description	Example
Public Read Directory	Default directory for all files for public downloading.	C:\EV50\PUBLIC
Default Timeout	Maximum elapsed time (in seconds) TFTP will wait until it gets a response from a target device. The optimal value depends on your specific network.	5 seconds
Default Retries	Maximum number of attempts TFTP will try to get a response from the target device before declaring that the session has failed.	3

Viewing the TFTP Process List

When the TFTP server receives a file transfer request, an entry appears in the process list window. For each entry, the following parameters are displayed:



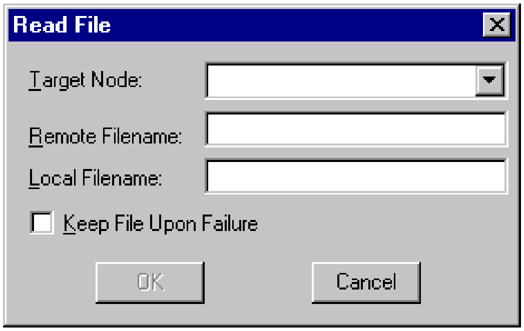
TFTP server

When the TFTP server receives a file transfer request, an entry appears in the process list window. For each entry, the following parameters are displayed:

Parameters	Description	Example
Target Node	The IP address of the device that initiates the TFTP file transfer session.	192.74.255.74
Action	Indicates file download or simple transfer.	SEND
Filename	The file being accessed.	SNMPDRV.BIN
Progress	Current status of the file transfer (based on a percentage of the file that has already been transferred).	10%

Fetching Files from Other Servers

You can use the TFTP server as a client to receive files from other TFTP servers.



To start a transfer session:

1. Choose *Read File From...* from the Service menu.
2. A dialog box will appear requesting additonal information. Check the box *Keep File Upon Failure* to save a partial file transfer.

   The required parameters are defined below.

Parameters	Description	Example
Target Address	IP address of the target TFTP server. This is the source of the original file.	192.74.255.74
Remote Filename	The file you want from the TFTP server. The path is restricted to the EliteView public directory for security reasons.	SNMPDRV.BIN
Local Filename	Name of the file after it has been received.	SNMPDRV2.BIN
Keep File Upon Failure	Click on this check box to save partial results when file transfer fails.	

3. Click on the OK button to begin the transfer process. The Progress report is less complete here, because TFTP cannot determine file size beforehand when transferring an arbitrar.05y file.

Telneting to Other Computers on the Network

Telnet is a TCP/IP application protocol that allows you to access a remote computer system (e.g., a UNIX or SUN workstation) as though you were attached locally via a serial terminal. As long as you have a user account on the target system and the necessary user priviledge, you can execute any text command.

If you require frequent access to a particular workstation, you may want to include it in your network map as.03 an IP Node.

To telnet into a workstation using your network map,

1. Select the appropriate IP node from the network map with your mouse.
2. Open the Tools menu under the main EliteView window and click on Telnet.
3. Log into the remote workstation.

CHAPTER 6

SNMP MIB MANAGEMENT

For all of SMC's intelligent network devices which include an SNMP-based management agent, you can use the device management modules in EliteView to easily access and manage detailed network information. EliteView's map module allows you to intuitively "zoom in" on objects to see low-level details on device hardware/software configuration and associated network interface parameters. For all SNMP-based devices (both ours and third-party products), you can access the complete SNMP Management Information Base (MIB) using the MIB Browser utility.

EliteView provides three basic MIB management utilities:

- ***MIB Compiler*** - Used to update or add modules to MIB database.
- ***MIB-2 Viewer*** - Displays MIB-II variables based on a functional grouping.
- ***MIB Browser*** - Provides full access to all MIB variables, such as MIB II, Bridge MIB, as well as SMC private MIBs.

This chapter provides detailed information on managing your device database. The following sections use objects you have added to your network map. If you do not have a network map, you may want to turn to "Creating Network Maps" in Chapter 4. For more information about managing specific devices, refer to the appropriate EliteView users guide.

MIB Compiler

The MIB Compiler is used to maintain the MIB database used by EliteView. Definitions for standard objects, network devices, or private third-party devices can be compiled and included in this database. All device management applications under EliteView consult this database when accessing devices. Under normal use, rely on the setup program for new management applications to automatically adjust the MIB database. However, if you need to modify the database yourself, the compile operation can be carried out interactively, or as a batch process. Specific MIB databases can also be unloaded when they are no longer in use, or out of date.

Tip: Compiling changes the MIB. Quit EliteView before running this process to make sure no module accesses the database while it is being compiled.

MIB Database

MIB.DBF, **MIB.DBF** and **MIB.MDX** - You can find these files in the EliteView directory. MIB text files have the following format:

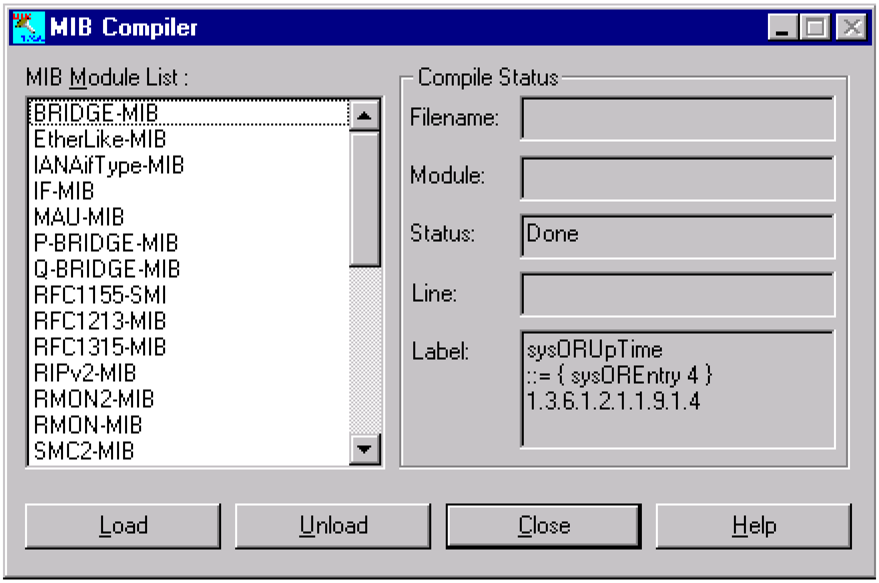
```
<MIB Name> BEGIN
   .
   .
   .
   END
```

Each MIB text file can contain several MIB modules, which we may call <MIB name>. EliteView accesses external MIB variables via this name.

Starting the MIB Compiler

To run the MIB Compiler,

Select MIB Compiler from the EliteView program group.
It will automatically load the current MIB database as shown below.



MIB Compiler

MIB Module List :

- BRIDGE-MIB
- EtherLike-MIB
- IANAifType-MIB
- IF-MIB
- MAU-MIB
- P-BRIDGE-MIB
- Q-BRIDGE-MIB
- RFC1155-SMI
- RFC1213-MIB
- RFC1315-MIB
- RIPv2-MIB
- RMON2-MIB
- RMON-MIB
- SMC2-MIB

Compile Status

Filename:

Module:

Status: Done

Line:

Label: sysORUpTime
::= { sysOREntry 4 }
1.3.6.1.2.1.1.9.1.4

The following table describes each function.

Item	Description	Sample Display
MIB Module List	Shows all the modules used by EliteView.	
Compile Status		
Filename	The filename for a module to add or update	smc8608.mib
Module	The name of a module as recognized by the MIB database.	SMC8608-MIB
Status	Shows the current compile status.	Merge
Line	The current line being compiled.	145
Label	The macro currently being processed.	
Functions		
Load	Loads an MIB text file into the database.	Filename: C:\EV60\MIB\SMC.MIB
Unload	Unloads an MIB module from the database. If the specified module has any dependencies; i.e., has other modules attached to it, the compiler will ask whether or not you want to unload the specified module and all of its dependencies. This is analogous to deleting a subdirectory – You cannot delete a subdirectory without first deleting the files it contains.	Current MIB: RFC1213-MIB

To load a new MIB:

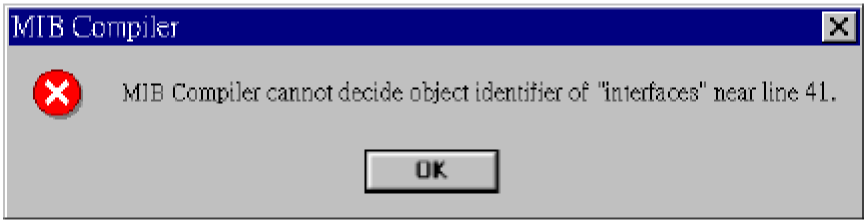
1. Select *Load* from the MIB Compiler.
2. Type the full name of the MIB file in the *Filename* field.

   Each MIB file defines an MIB module. The MIB name is indicated at the beginning of the file as in the example below, where the name of the module is given as RFC1213-MIB.

   ```
   RFC1213-MIB DEFINITIONS::=BEGIN...
   ```

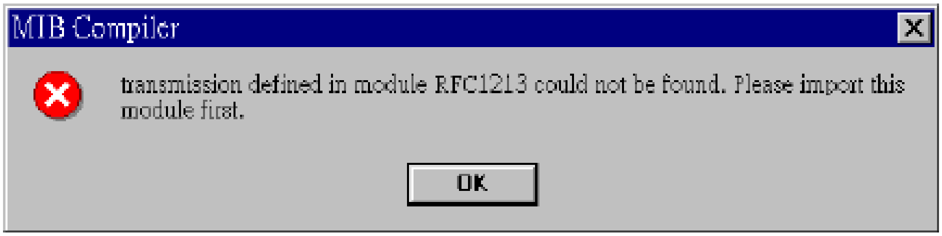
All MIBs are defined under the same tree, making MIB definitions related to one another. The EliteView MIB database identifies MIB objects with names and modules. Each object has a distinct name within a module, and each module may be loaded or unloaded at any time.

The MIB Compiler reads the specified MIB file and displays the names of the MIB objects as the file is scanned. If any error occurs during this process, it will stop scanning and display the object label and the line number near where the error occurred as in the example below:



MIB Compiler cannot decide object identifier of "interfaces" near line 41.

If an object identifier is not defined, the following message may appear on the screen:



transmission defined in module RFC1213 could not be found. Please import this module first.

If an unknown object identifier is found in the definition for OBJECT-TYPE, it will be reported after all MIB objects are checked as in the following error message. In the example given below, you should check the last line of the OBJECT-TYPE macro section.

```
Error: Lost connection for node sysObjectID
```

To unload MIB modules,

You can unload an MIB module when it is no longer needed, or when you wish to update or replace it. Highlight the MIB you want to unload, list currently loaded modules, and then select Unload. If other MIB modules link to the module you wish to unload, EliteView will ask you to unload these modules first before you can successfully unload the required module.

To view the MIB Module List

Use the scroll bar next to the *MIB Module List* on the MIB Compiler. This list shows the modules currently loaded in the MIB compiler.

Things to remember when using the MIB Compiler:

- The system always loads the current MIB database.
- To replace an MIB with its new version, first unload the older version and then load the new version.
- If an error occurs during the loading or unloading process, the MIB Compiler automatically skips the erroneous command and moves on to the next one.

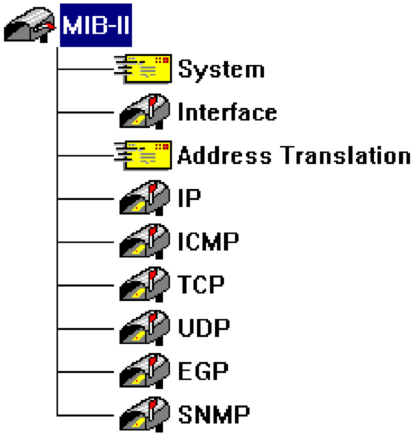
MIB-2 Viewer

The MIB-2 Viewer is a generic SNMP monitoring tool used to browse MIB II (i.e. the Management Information Base defined by RFC 1213). By browsing through this MIB, you can access information recorded in MIB II for any SNMP-compliant device attached to your network. However, to set MIB variables, remember to use the MIB Browser.

Note that although all SNMP devices support MIB II, most groups are optional and may not be implemented. Only System, Interface and SNMP groups are likely to be found on all systems.

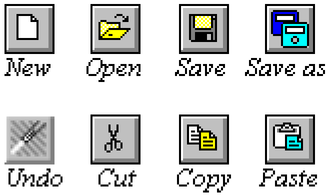
You can invoke the MIB-2 Viewer from the Tools menu in the main EliteView program (preferably after selecting a device from the EliteView map), or by clicking on the appropriate icon in the EliteView program group. If you are not opening the MIB-2 Viewer directly from the EliteView map, then you must also fill in the device interface parameters in the MIB2VIEW initialization dialog box, including protocol type, target address, SNMP community, and polling specifications.

Once loaded, the MIB-2 Viewer begins searching for definitions for the specified object in EliteView's MIB database. After the target data has been retrieved, open the directory branches along the path leading to the required variable by double-clicking on each intervening node. (Note that the icons for collapsible nodes are highlighted at the top.)

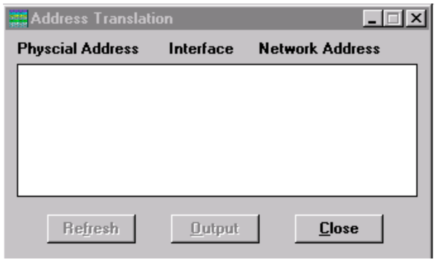


After opening the required window, you can readily view all the key variables associated with the selected topic. To copy MIB data into the Output window, just press the Output button. You can edit and save the information copied to the Output window using the buttons provided in toolbar.

The toolbar for the MIB Viewer contains two basic button groups for file management and output editing. After creating a status report via cut & paste and manual annotation, be sure to save your file before exiting EliteView. Note that a brief description of every toolbar button is provided in the Status bar at the bottom of the screen. (Refer to the description on how to use the Status Bar in Chapter 3.)



A description of the menus and a few of the display screens used by the MIB-2 Viewer is provided below. For more detailed information of specific variables, refer to the MIB Browser or the appropriate RFC.



Menu Bar

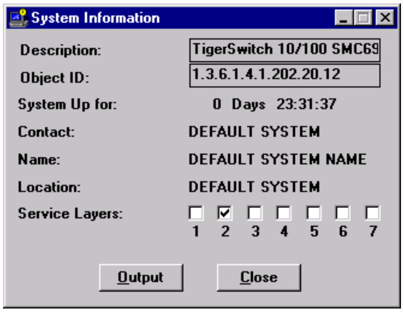
The menu bar for the MIB-2 Viewer provides five key menus, namely, File, Edit, Search, Window and Help. Clicking on any of these items will open a pull-down menu from which you can invoke corresponding commands.

Menu	Description
File	Contains commands to open and save report files (*New, Open, Save, Save As*), and to exit the host management program (*Exit*).
Edit*	Contains standard editing commands used in conjunction with the Output Window.
Search*	Contains editing commands used to find or replace specified text in the Output Window.
Window	Contains standard commands for arranging your windows (*Cascade, Tile*) and icons (*Arrange Icons*), or switching to another window.
Help	Used to invoke available on-line help functions, or to display the revision number for your current version of EliteView.

** Only displayed in conjunction with the Output Window.*

MIB-2 Directory

The information displayed in the MIB-2 directory is presented in easily understood graphic windows. A few of these windows are fully described below. For further details on the other directory entries, please refer to textual descriptions provided by the MIB Browser.



Description: TigerSwitch 10/100 SMC69
Object ID: 1.3.6.1.4.1.202.20.12
System Up for: 0 Days 23:31:37
Contact: DEFAULT SYSTEM
Name: DEFAULT SYSTEM NAME
Location: DEFAULT SYSTEM
Service Layers: ☐ 1 ☑ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

System Information

System information is extracted from the System Group in MIB II (RFC 1213). It provides data on the SNMP agent installed on the device being monitored.

System Information Window

System Information provides data on the SNMP agent.

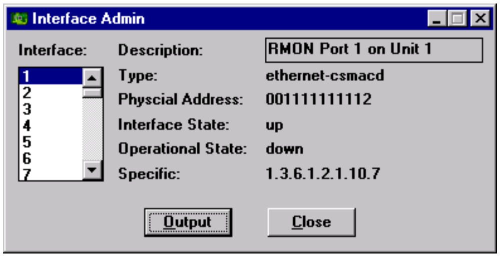
Field Descriptions for System Information Window

Field	Description
Description	Name of the management agent, version number and release date.
Object ID	Unique identifier for device model.
System Up for	Time since the SNMP agent was last re-initialized.
Contact	Name of contact person for this monitored node; and how to make contact.
Name	Administrative designation for this node.
Location	Physical location of this node.
Service Layers	Internet protocol services offered by this node. The OSI model includes - (1) Physical, (2) Data Link, (3) Network, (4) Transport, (5) Session, (6) Presentation, and (7) Application layers.

Interface Administration

Data for this window is extracted from the Interface Group in MIB II (RFC 1213). Each MAC frame type supported by a physical network interface is listed as a unique logical network interface in the display window. (Refer to ifType in RFC 1156.) For example, even though there may only by one physical network interface on a monitored device, it may concurrently support both ethernet-csmacd and iso88023-csmacd. Also note that the same type may be reported more than once where a logical interface includes several valid subtypes. For example, ethernet-802.2 and ethernet-II both fall under ethernet-csmacd.

This window provides a description and status information on each subnetwork connected to this system



Interface: 1, 2, 3, 4, 5, 6, 7

Description: RMON Port 1 on Unit 1

Type: ethernet-csmacd

Physcial Address: 001111111112

Interface State: up

Operational State: down

Specific: 1.3.6.1.2.1.10.7

Field Descriptions for Interface Admin Window

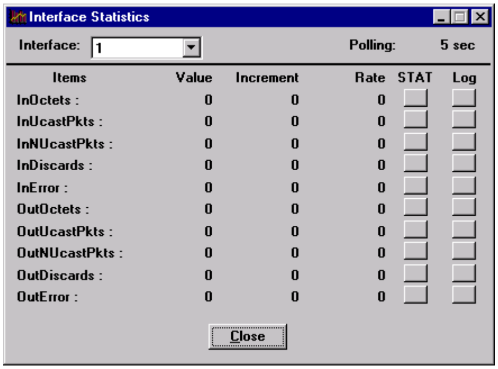
Field	Description
Interface (Index)	A unique index for each subnetwork connection.
Description	A textual description of the interface, which may include items such as the product name, manufacturer, or version number for the hardware interface.
Type	The interface type based on the physical/link protocols running immediately below the network layer, e.g. *ethernet-csmacd* (where csmacd indicates Carrier Sense Multiple Access / Collision Detection).
Physical Address	The interface address used at the protocol layer immediately below the network layer. This value will be zero for interface types that do not support such an address.

Field	Description
Interface State	The requested state of the interface including the following items. Note that when the interface is in *testing* mode, no operational packets can be passed. *up* - ready to pass packets *down* - not allowed to pass packets *testing* - operating in a test mode
Operational State	The current operational status of the interface, including the same states as defined above for *Interface State*, except where *down* indicates that the device is not capable of passing packets.
Specific	Reference to an MIB with definitions for the media type (e.g., Ethernet) used by the interface. If no information is available, this value will be zero.

Interface Statistics

This window provides statistical information on the traffic and associated errors for the selected interface.

Data for this window is extracted from the Interface Group in MIB II (RFC 1213). An entry is included for each subnetwork interface. This window provides information on the amount of traffic passing through this interface and the associated errors. This information can also be displayed as a graph by clicking STAT, or passed to the EliteView Log Manager by clicking LOG and defining the required event criteria. Refer to the sections on Viewing Statistics or Adding a Log Process if these functions are required.



Interface: 1 Polling: 5 sec

Items	Value	Increment	Rate	STAT	Log
InOctets :	0	0	0	☐	☐
InUcastPkts :	0	0	0	☐	☐
InNUcastPkts :	0	0	0	☐	☐
InDiscards :	0	0	0	☐	☐
InError :	0	0	0	☐	☐
OutOctets :	0	0	0	☐	☐
OutUcastPkts :	0	0	0	☐	☐
OutNUcastPkts :	0	0	0	☐	☐
OutDiscards :	0	0	0	☐	☐
OutError :	0	0	0	☐	☐

Field Description for Interface Statistics Window

Field	Description
Interface (Index)	A unique index for each subnetwork connection.
InOctets	Total number of bytes received on the interface, including framing characters.
InUcastPkts	Number of subnetwork unicast packets delivered to a higher-layer protocol.
InNUcastPkts	Number of non-unicast packets (i.e., broadcast or multicast) delivered to a higher-layer protocol.
InDiscards	Number of inbound packets that were discarded even though no errors were detected. One reason for discarding such packets is lack of buffer space.
InError	The number of inbound packets containing errors that prevented them from being delivered to a higher-layer protocol.
OutOctets	Total number of bytes transmitted from the interface, including framing characters.
OutUcastPkts	Total number of packets requested by higher-level protocols for retransmission to a unicast address, including those either discarded or not sent.
OutNUcastPkts	Total number of packets requested by higher-level protocols for retransmission to a broadcast or multicast address, including those either discarded or not sent.
OutDiscards	The number of outbound packets containing errors that prevented them from being delivered to a higher-layer protocol. One reason for discarding such packets is to free up buffer space.
OutError	Number of outbound packets that could not be transmitted due to errors.

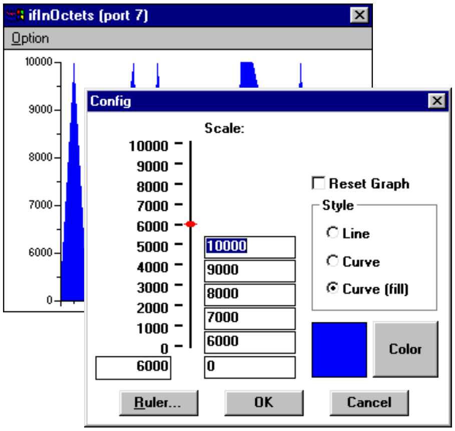
Viewing Statistics

The STAT button is used to display a real-time graph of the corresponding counter read during each polling interval.

Click Stat to display the corresponding graph.

Click on *Config* under the Option menu to change parameters for the graph. The Config Statistics dialog box appears. You can change the graph's scale, ruler, style (i.e., filled curve, curve (fill), and color, or reset the graph. Click *OK* to return to the graph, or *Cancel* to abort any changes.

Click on *Exit* under the Option menu to return to the invoking window.



Scale fields: 10000, 9000, 8000, 7000, 6000, 0; 6000

☐ Reset Graph

Style
○ Line
○ Curve
● Curve (fill)

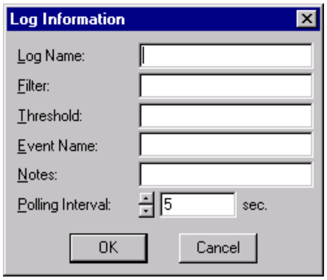
Adding a Log Process

The *Log* button is used to define a new log process for the Log Manager.

The Log Manager performs the following basic functions:

- Periodically records values for device variables.
- Sets thresholds to trigger events when conditions are met.

Click on *Log* to display the Log Information dialog box.



Log Information

Log Name:

Filter:

Threshold:

Event Name:

Notes:

Polling Interval: 5 sec.

Complete the entries for *Log Name*, *Filter*, *Threshold*, *Event Name*, *Notes* and *Polling Interval* to provide precise control over Logging operations. Refer to Chapter 7, “Collecting Data with the Log Manager,” for a detailed explanation of the Log Information dialog box.

MIB Browser

The MIB Browser is a generic SNMP management tool used to browse MIBs. By browsing through the MIB, you can send commands to get or set information defined in the MIB.

You can invoke the MIB Browser by selecting MIB Browser from the Tools menu in the main EliteView program, or by clicking on the MIB Browser icon in the EliteView program group. Once loaded it begins searching for definitions for the specified object in EliteView’s MIB database. After selecting a desired variable, you may issue SNMP commands to get or set various device parameters.

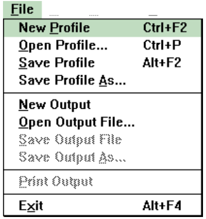
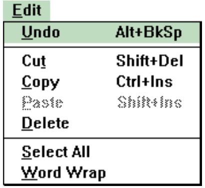
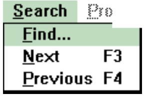
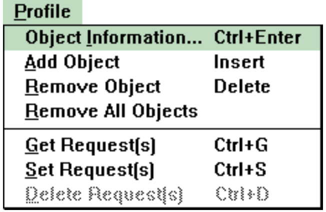
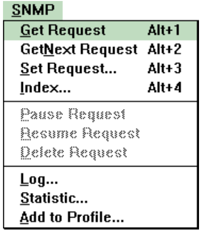
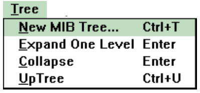
Basic Functions of MIB Browser

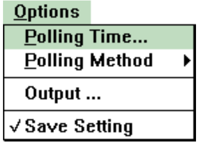
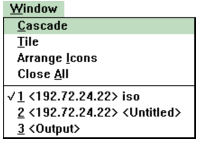
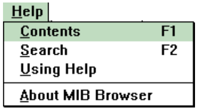
1. The MIB Browser window provides access to the EliteView MIB database. The definitions for all known MIB variables can be consulted here. This window also provides access to variables stored in managed devices via the SNMP *GET*, *SET* and *GETNEXT* commands. You can view any of these variables in either ASCII or binary format, set values for variables that provide write access, pass specified variables to the profile window, or specify a log process.
2. The output editor provides basic tools required to edit data retrieved from the target device. To open an output window, click on the output button in the main toolbar, then set the output mode and retrieved field entries using the Options menu.
3. The profile window is used to specify commonly referenced variables, which can be more easily accessed and managed as variable sets. Use profile functions to modify the selected variable set, and to fetch or store values as a set.
4. The log utility allows you to define new processes for the Log Manager, and quickly paste selected variables into the filter and threshold fields.

Menu Description

The menus provided for the MIB Browser are briefly introduced below. Menus are only enabled for applicable windows as indicated in the Window field.

Menu Definitions

Menu	Label	Window*	Description
 File New Profile Ctrl+F2 Open Profile... Ctrl+P Save Profile Alt+F2 Save Profile As... New Output Open Output File... Save Output File Save Output As... Print Output Exit Alt+F4	File	Tree Profile Output	**File and Printer Access -** Provides standard file access and printer functions for the Output Editor window, open and save functions for Profile files, and exit from the MIB Browser.
 Edit Undo Alt+BkSp Cut Shift+Del Copy Ctrl+Ins Paste Shift+Ins Delete Select All Word Wrap	Edit	Output	**Editing Tool -** Provides standard editing functions including cut and paste, item selection, and word-wrap display.
	Search	Output	**Search Functions -** Provides search related functions.
 Profile Object Information... Ctrl+Enter Add Object Insert Remove Object Delete Remove All Objects Get Request(s) Ctrl+G Set Request(s) Ctrl+S Delete Request(s) Ctrl+D		Profile	**Profile Functions -** Provides batch access to MIB variables.
 SNMP Get Request Alt+1 GetNext Request Alt+2 Set Request... Alt+3 Index... Alt+4 Pause Request Resume Request Delete Request Log... Statistic... Add to Profile...	SNMP	Tree	**Access MIB Variables -** Provides standard access commands for MIB variables, along with utilities to add a new log process, display a graph for a selected variable, or add the current variable into the Profile window.
 Tree New MIB Tree... Ctrl+T Expand One Level Enter Collapse Enter UpTree Ctrl+U	Tree	Tree *UpTree*	**Navigate Tree -** Provides options to open a new tree (where the default sets the root at the currently selected node), expand or collapse the current node (i.e., same as double-clicking on a parent node), or move up the tree to the parent node (i.e., same as using the UpTree button).

Menu	Label	Window*	Description
 Options Polling Time... Polling Method Output ... √ Save Setting	Options	Tree Profile Output	**Polling -** Adjust timing for data requests, including polling interval, timeout, and retries. Set polling to comply with the retries defined in the *Polling Time* dialog box, or opt to continue polling until the queried device responds. **Output -** Set output to display values in ASCII or binary, and select value fields to pass to the Output Editor.
 Window Cascade Tile Arrange Icons Close All √1 <192.72.24.22> iso 2 <192.72.24.22> <Untitled> 3 <Output>	Window	Tree Profile Output	**Manage Windows -** Arrange windows or icons, or activate an existing window.
 Help Contents F1 Search F2 Using Help About MIB Browser	Help	Tree Profile Output	**Help Facility -** Access detailed help information about the MIB Browser.

** Window - indicates the windows for which this menu is active.*

Accessing Device Values

To fetch device values using the MIB Browser:

1. Start EliteView.
2. Open your network map. (Refer to Chapter 4 if you have not yet created your network map.)
3. Select the required device by clicking on it with the mouse.
4. Select MIB Browser from the Tools Menu. This will bring up the MIB Browser window.
5. If this is the first time you have used the MIB Browser, the New MIB Tree dialog box will display. Open a new MIB tree by specifying the root variable :
   a) Indicate whether the object type is a Label or numeric Object Identifier (e.g. the object identifier for internet is 1.3.6.1).
   b) Select the MIB containing the root from the scroll list.
   c) Specify the tree root (or the name of the required object) in the Object edit box.
   d) To use the first entry matching the specified prefix, clear the Find exactly check box. To set the tree root at the exact variable as specified, mark the Find exactly check box. Click OK to continue or Cancel to start the browser without a tree window.
6. If the MIB Browser is opened from within the EliteView Platform program, the protocol and address of the selected device will be used. However, if the MIB Browser is activated directly from the EliteView program group, the Program Information dialog box will be displayed for a new tree. Specify the network protocol used for this device as UDP/IP or IPX. Verify the target address, and then press OK to open the new tree window or Cancel to start the browser without a specified tree.

Select MIB Browser from the Tools Menu of the main EliteView program, specify the root for a new tree in the New MIB Tree dialog box, and indicate the network protocol used for the selected device.

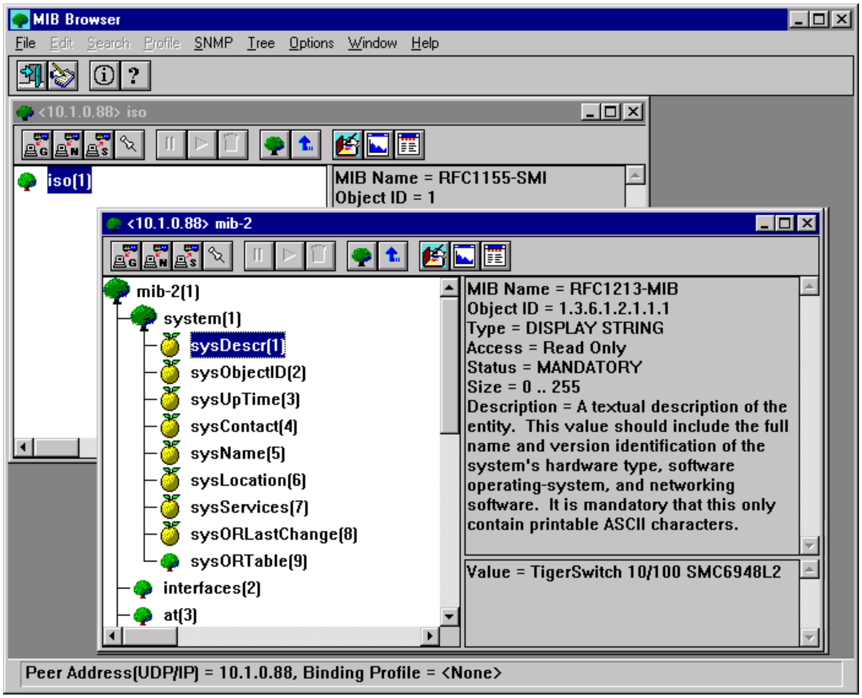
If this is not the first time you have used the SNMP MIB Browser, EliteView will automatically open the window(s) last used.

- press for new subtree

- collapsable node

- expandable node

- leaf node



MIB Name = RFC1155-SMI
Object ID = 1

MIB Name = RFC1213-MIB
Object ID = 1.3.6.1.2.1.1.1
Type = DISPLAY STRING
Access = Read Only
Status = MANDATORY
Size = 0 .. 255
Description = A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contain printable ASCII characters.

Value = TigerSwitch 10/100 SMC6948L2

Peer Address(UDP/IP) = 10.1.0.88, Binding Profile = <None>

SNMP window showing MIB description as it appears in the MIB database and value for the highlighted item. Each subwindow may be resized by dragging on the inner frame with your mouse.

7. Locate the MIB variable you want to browse. Use the scroll bar to move the tree display up or down, and double-click on any intermediate nodes in the path to the required variable to open the map for a lower-level hierarchy. After you have found the variable, press the Get Request button to fetch the required information. The data display options are binary or ASCII. To change the output mode, use the *Output* selection under the *Options* menu.

Tip: Object names may be duplicated in different MIB modules. For accurate results, be sure to enter the correct node.

8. When you select any MIB variable, the textual description (as listed in the database) is automatically displayed to the right of the tree. Standard entries in the list box include the MIB Name, Object ID, Type, Access, Status, Range, Size, Description and more depending on the variable selected. The following table describes each entry.

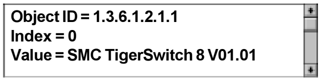
Field Description of Object Parameters

Item	Description
Label	Standard name for MIB variable (as appearing in the MIB tree).
MIB Name	Name of the MIB module to which the variable belongs.
Object ID	Dotted-decimal identifier for current variable, indicating its exact location in the database structure.
Type	Data type of the object. This is NULL if the object is a structure (i.e., not a real variable).
Access	Refers to the way the data can be accessed. This item is only meaningful for real variables. Acceptable values include "Read Only," "Read/Write," or "No Access." Note that a Read-Only object does not support the "set" operation.
Status	Can be MANDATORY, OPTIONAL or DEPRECATED. In general, a mandatory object must be implemented, an optional object may be omitted, and a depreciated object may be taken out of a definition. However, according to grouping conventions as defined in the standards, objects may be grouped such that all of them are implemented, or omitted altogether. In such case these objects may have the status of MANDATORY but not be implemented without violating the rules.
Index	Index to current table entry. The IP address of the target device is commonly used as an index.
Value	The value of the current variable. Value type depends on the specific variable.
Range	Range of the variable in (x, y) format.
Size	Size of the string data in number of bytes.
Description	Text that briefly describes the use of the corresponding variable.

9. When you execute a Get request (or double-click on an item in the MIB tree), the value of the selected variable is retrieved from the managed device and displayed beneath the textual description. However, note that when the GetNext request is executed, the next variable actually retrieved may be several nodes away, so the ObjectID and Index entries are also displayed.

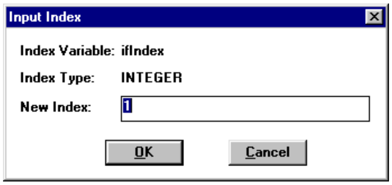
Get Request

Get Next Request


Object ID = 1.3.6.1.2.1.1
Index = 0
Value = SMC TigerSwitch 8 V01.01

Index

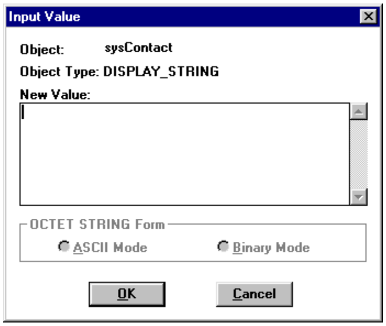
10. The MIB database contains both simple variables and tables. For example, the *atEntry* uses the IP address as a pointer to specific network devices. When you expand a table by double-clicking on the associated node, the Input Index dialog box will open to query for the required table index. You can change the table index for the current variable (provided it is a table) using the *Index* button.


Input Index

Index Variable: ifIndex
Index Type: INTEGER
New Index: 1

Set Request

11. You can modify the value of variables for which you have write access (as indicated in MIB definition's Access field). Once you have located the required variable, click the Set Request button to open the Input Value dialog box. The object and object type, as defined in the MIB, are listed in this box. Input the new value, ensuring that you use the correct type (as indicated in MIB definition's Type field). Then click OK to write this value into the managed device, or Cancel to abort the change.


Input Value

Object: sysContact
Object Type: DISPLAY_STRING
New Value:

OCTET STRING Form
○ ASCII Mode ○ Binary Mode

12. If you need to pause/resume or discontinue a data request, click on the appropriate buttons in the toolbar. To exit the MIB Browser, click on the exit button.

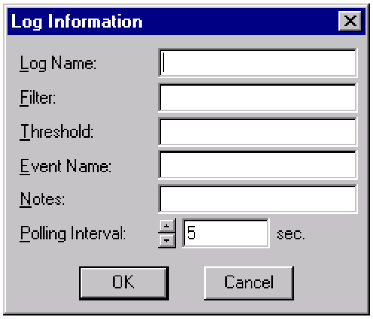
Pause *Resume* *Abort* *Exit*

13. You may display the real-time value for suitable variables (i.e., types including counter, integer, gauge or time ticks) by clicking on the Graphical Statistics button. (Refer to the section on Viewing Statistics on page 14.)

Graphical Statistics

14. Use the Logging Data button to have the Log Manager periodically record values for device variables or set thresholds to trigger events when conditions are met. To pass a request to the Log Manager, select the MIB variable from the SNMP tree you want to log by highlighting it with your mouse. Then click on the log button to open the LOG Information dialog box. Input entries for Log Name, Filter, Threshold, Event Name, Notes and Polling Interval to provide precise control over logging operations. (See Chapter 7, "Collecting Data with the Log Manager" for more detailed information on logging data.)

Logging Data



Log Information

Log Name:
Filter:
Threshold:
Event Name:
Notes:
Polling Interval: 5 sec.

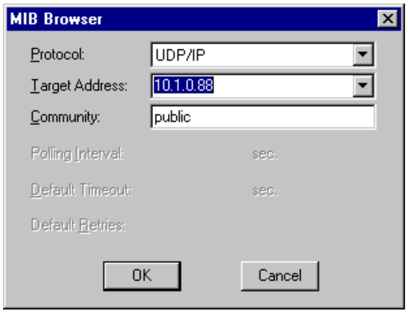
Using Profile Windows

If you frequently access a fixed set of variables, place them in a profile for easy access and management. The profile window is used to define or access commonly referenced variables. The functions provided by this window allow you to modify the selected variable set, and fetch or store values as a set.

The Profile Window provides a subset of MIB access commands, of which *Object Information* and *Remove Object* apply specifically to the current item in the Profile window. In additon to adding and removing objects, you can also execute get and set requests from this menu. Object Information opens a dialog box that displays object values, including the MIB name, along with object label, ID, range, index and type. The dialog box allows you to modify the value of the current variable or to change the index for a table variable.

To create a profile:

1. Select New Profile from the File menu.
2. Verify the default entries in the MIB Browser dialog box. Select the network protocol used for the target device as UDP or IPX. Change the target address and community string if required.
3. Press OK to continue or Cancel to abort the operation. The Profile window will open as shown below.



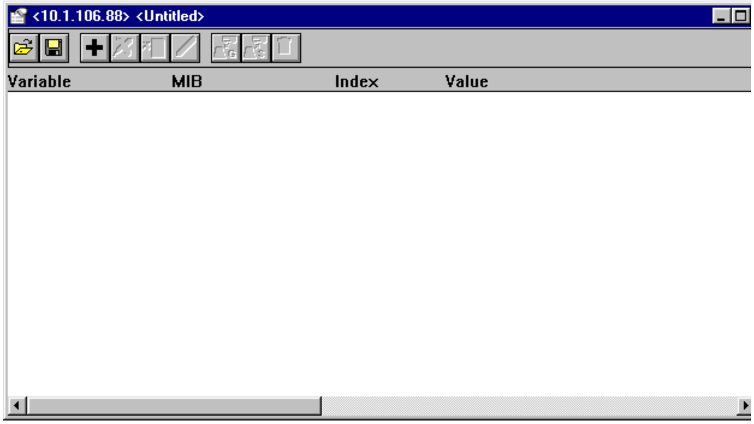
MIB Browser

Protocol: UDP/IP
Target Address: 10.1.0.88
Community: public
Polling Interval: sec.
Default Timeout: sec.
Default Retries:

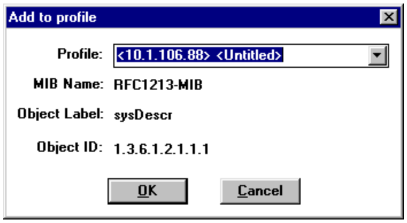
Note: To open an existing profile, select Open Profile from the File menu.

To open the profile window, select New Profile from the File menu and enter the target parameters in the dialog box.

Select the required variables from the MIB Tree window and press the Add to Profile button.

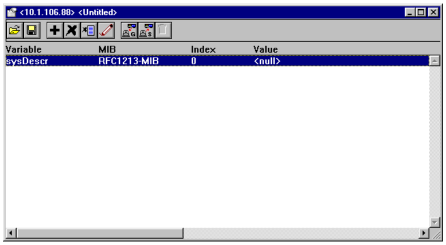


4. Select the required variable from the MIB Tree window by highlighting it with your mouse and then pressing the *Add to Profile* button.
5. The *Add to profile* dialog box will open, allowing you to verify the selected variable and destination profile. Press OK to include the selected variable in your profile or Cancel to abort the selection.



Profile: <10.1.106.88> <Untitled>
MIB Name: RFC1213-MIB
Object Label: sysDescr
Object ID: 1.3.6.1.2.1.1.1

6. MIB name, object label, table index and value fields for the selected variable will be added to the Profile window. Note that the table index will be 0 if the selected item is not a table variable, and the initial value displayed is <null>.
7. Use the editing buttons provided in the Profile window toolbar to insert or remove variables, or to display the MIB definition for any variable.
8. Use the data request buttons to get or set the values for the profile set as a group. To cancel a request, use the delete button.



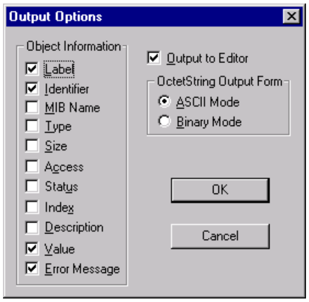
Variable	MIB	Index	Value
sysDescr	RFC1213-MIB	0	<null>

9. After defining the required profile list, use the save file button to write the profile to disk, or the open file button to start a new profile.

Add Object
Remove Object
Remove All
Modify Object
Get
Set
Delete
Open
Save

Using the Output Editor

The output editor is designed for creating system status reports based on information you fetch from the MIB database. You can insert a wide range of object information into the output editor using the data request functions provided in the MIB Tree and Profile windows. The output editor provides basic tools required to edit data retrieved from the MIB.


Output Options

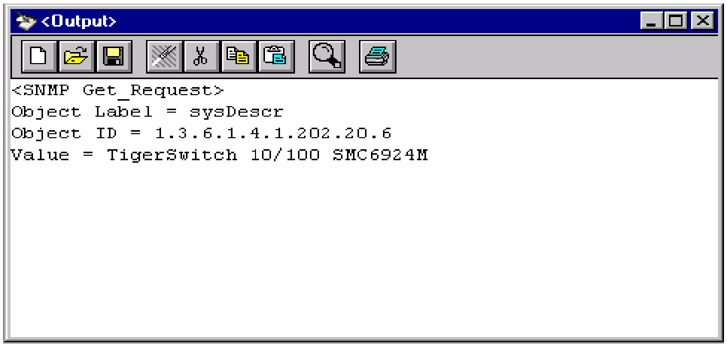
Object Information
☑ Label
☑ Identifier
☐ MIB Name
☐ Type
☐ Size
☐ Access
☐ Status
☐ Index
☐ Description
☑ Value
☑ Error Message

☑ Output to Editor

OctetString Output Form
● ASCII Mode
○ Binary Mode

To open an output window:

1. Click on the output button in the main toolbar to open the output window.
2. Select Output from the Options menu to open the Output Options dialog box. Mark the Output to Editor checkbox, select the output mode as ASCII or binary, and select the object information to display. Then press OK to continue or Cancel to abort the selected output options.
3. Select the required variables from the MIB Tree, and then use Get, GetNext or Set requests to insert information into the output window. Get or Set requests executed in the Profile window will also insert returned data into the Editor window.



```
<SNMP Get_Request>
Object Label = sysDescr
Object ID = 1.3.6.1.4.1.202.20.6
Value = TigerSwitch 10/100 SMC6924M
```

4. Use the editing functions provided in the output editor toolbar. (Refer to the description of how to display information on toolbar buttons using the Status Bar in Chapter 3.) Note that edited reports can be printed directly from the editor window or saved to disk for later reference.

CHAPTER 7

COLLECTING DATA WITH LOG MANAGER

The Log Manager is a powerful tool for the network manager. By collecting relevant network statistics periodically from all SNMP-compliant network devices, the Log Manager can:

- Record network characteristics (e.g., utilization, error rate)
- Set thresholds to generate events when values are out of range
- Provide the basis upon which you can predict future network load based on current usage and plan for future requirements

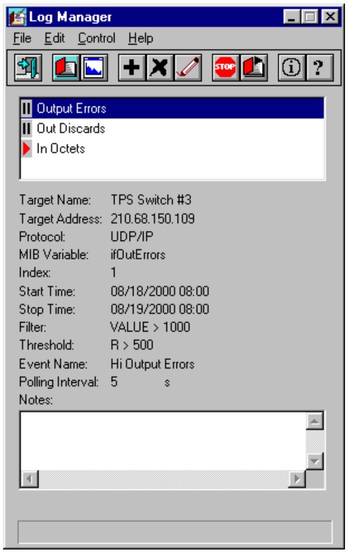
The Log Manager is designed with a filtering mechanism that logs only the data you indicate. The Log Manager works with other EliteView applications such as the Event Manager, Log Database Manager, and the MIB Browser. Thresholds can be set to generate specific events to warn the network manager of certain unique conditions. All information can be logged in a database and easily retrieved in numeric or graphic form. You can pause logging at any time for a selected process or for the entire system, if necessary.

Events specified in the Log Manager are passed to the Event Manager. In response to an event, an audible alarm, on-screen message, or a user-defined application can be executed.

Overview

To view the Log Manager window, select *Log Manager* from the Utilities menu of the main EliteView program, or from the EliteView program group. If needed, the Log Manager can be started automatically along with the EliteView main program; see “Customizing EliteView,” Appendix B. The Log Manager can also be invoked from other management modules, like the MIB Browser.

The Log Manager dialog box consists of a menu bar, a toolbar, a list of all user-defined log processes, and a summary of log parameters for the currently selected process. Note that the same functions are provided in both the menu bar and toolbar.



‖ Output Errors
‖ Out Discards
▶ In Octets

Target Name: TPS Switch #3
Target Address: 210.68.150.109
Protocol: UDP/IP
MIB Variable: ifOutErrors
Index: 1
Start Time: 08/18/2000 08:00
Stop Time: 08/19/2000 08:00
Filter: VALUE > 1000
Threshold: R > 500
Event Name: Hi Output Errors
Polling Interval: 5 s
Notes:

The first step in using the Log Manager is to decide which MIB variables you should log to solve a specific problem or just to maintain a record of system performance. If you are not sure about the MIB variables, refer to their description in the MIB Browser or in the relevant RFC documents.

To display the current log parameters for any process, click on the required process with your mouse. (The þ and | | markers to left of each process entry indicate whether this process is currently active or paused.)

The toolbar provides access to all functions in the Log Manager. The three key function groups include data display, process editing, and process management. To delete, edit, activate or pause a process, first identify the concerned process by clicking on it with your mouse, and then select the appropriate function.

Numeric/Graphic Data Display

Add/Delete/Edit Process

Activate/Pause System

Activate/Pause Process

For a description of log parameters listed in the main window, refer to the “Field Description for Log Manager/Information Dialog Boxes” in the following section.

Editing a Log Process

EliteView’s powerful Log Manager gathers a wide variety of network statistics based upon rules (i.e., log processes) you define. Using the Log Manager, you can monitor any MIB variable for a target SNMP device. Each log process defines the parameters under which data related to the specified MIB variable is collected and placed in the central database. The Log Manager controls the data collection activity, as well as allowing you to display and manipulate the data gathered as described in the section on “Viewing Log Data.”

Add/Delete/Edit a Log Process

To add a new log process:

1. Click the “Add a new record” toolbar button. ✚

2. Fill in the parameters in the Log Information dialog box. Process parameters are defined in the table on the next page.

   Not all parameters are required. Only *Protocol*, *Target Address* and *Variable* are mandatory.

Protocol - Click the down arrow to choose from UDP (default), IPX or Ethernet.

Target Address - Use appropriate notation for selected protocol.

Community - Community strings control access rights to network resources. Define your own community string to prevent unauthorized access to critical resources. However, if you do not have any special security concerns, then retain the default community string of public. Refer to the description for object editing in “Defining the Network Configuration,” Chapter 4.

MIB Module - Includes standard MIBs based on the RFCs, device-specific MIBs, and private MIBs. For variable names not duplicated in other MIB modules, the default of *Any* is sufficient

3. Choose *OK* to add the new log process or *Cancel* to abort your selections.

Field Description for Log Manager/Information Dialog Boxes

Parameters	Description	Example
Community[2]	Community string used to access the target device.	public
Log Name[2]	Process identifier displayed in process list.	
Target Name[1]	A user-defined name for this device stored in the Name Database.	MIS Server
Target Address	Network address of the target device.	192.72.24.232
Protocol	Network transport protocol used to request data, i.e., UDP/IP, IPX or Ethernet.	UDP
MIB Module[2]	Module to search for variable. Select a specific module or *Any* module.	RFC1213-MIB
MIB Variable	Name of the variable being polled, (as defined in the MIB database).	hubTotalBytes
Index[3]	Index to entry in a table variable.	1
Start Time	Time to start log process. (YYYYMMDDHHMM)	199908260800
Stop Time	Time to stop log process. (YYYYMMDDHHMM)	199908260800
Filter	Formula used to filter information.	(VALUE > 100) AND (TIME < 120000)
Threshold	When true, this formula will generate an event and pass it to the Event Manager.	R>100
Event Name	Name of event enabled by Event Manager when threshold condition is met.	CRITICAL
Polling Interval	Elapsed time between data requests (sec).	30
Notes[1]	Short comment associated with this log process.	If excess CRC errors occur, check port 9

1 - These variables only appear in the Log Manager dialog box.
2 - These variables only appear in the Log Information dialog box.
3 - If you are unsure if an index is required for a variable, first examine the specification for that variable under the MIB Browser.

Modifying a Log Process

You can easily “fine tune” the parameters for any log process.

To modify a log process:

1. Highlight the relevant process in the Log Manager screen.
2. Open the Log Information by pressing the edit button.
3. Enter a new value for any parameter.
4. Click *OK* to accept the new value or click *Cancel* to abort any changes.

To delete a log process:

1. Highlight the relevant log process in the Log Manager.
2. Click the "Delete record" toolbar button.

   Note that deleting a process also deletes all log files associated with that process.

Log Controls

The Log Manager has two basic toggle switches for logging:

- System Activate/Pause toggles all logging activities on/off.
- Log Activate/Pause toggles logging for a selected process.

Activate/Pause System

Activate/Pause Process

Viewing Log Data

Log data is saved in dBASE-compatible files. EliteView offers several ways to view logged data. To view a log file:

- Open Log Manager and double-click on the required process.
- Highlight a process with your mouse, and then press the numeric or graphic data display icon. (Note that the numeric data display icon is just another entry point for the Log Database Manager.)
- Open the Log Database Manager from the Program Manager.

Log Manager

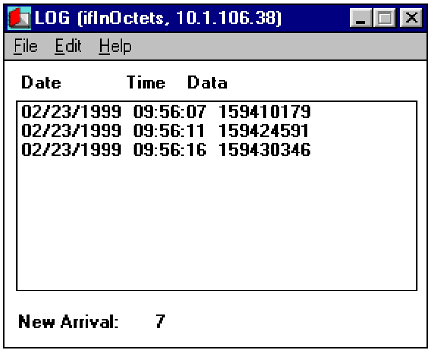
Numeric/Graphic Data Display

Log Database Manager

The Log Database Manager opens a numeric display for the selected file.

Using the Log Database Manager

The Log Database Manager displays data from the different log processes listed in the Log Manager window. Information in the log database may be readily copied and shared with other applications. Open the Log Database Manager as described above. The numeric display posts information for data matching the filter criteria up to the current polling interval, including the date, time and specific data for the selected variable. New Arrival shows the count for filtered data that have not yet been included in the display. Refer to the diagram on page 3-4 for an illustration of how the Log Manager works.



```
Date          Time      Data
02/23/1999  09:56:07  159410179
02/23/1999  09:56:11  159424591
02/23/1999  09:56:16  159430346

New Arrival:      7
```

File Menu

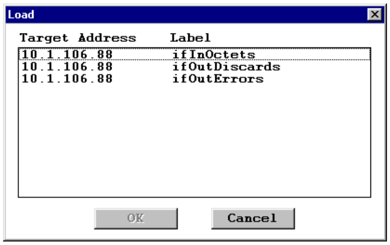
The file menu provides functions for retrieving log files, copying selected data to a specified file, or moving specified data to another file.

To copy selected data to another file, use the *Copy To..* command. This data is saved in standard dBASE format (with a **dbf** extension) and may be accessed using a database program for further processing.

To move selected data to another file, using the *Move To..* command. Moved data disappears from the database window. This data is saved in standard dBASE format (*.dbf) and may be accessed using a database program for further processing.

To load a data file, select *Load* from the File menu, select the required process from the Load list and press *OK*.

Note: The Load option is only enabled when the Log Database Manager is opened from outside the Log Manager (i.e., from the main EliteView program or from the EliteView group window). When using the Log Manager, the Log Database Manager will only load the process selected from the Log Manager dialog box.



Target Address	Label
10.1.106.88	ifInOctets
10.1.106.88	ifOutDiscards
10.1.106.88	ifOutErrors

Edit Menu

The edit menu provides functions for deleting selected entries, copying data to the clipboard, and refreshing the display.

To delete entries from the database, select the required items with your mouse, and then choose *Delete* from the Edit menu. Note that deleting all entries will not remove the log file.

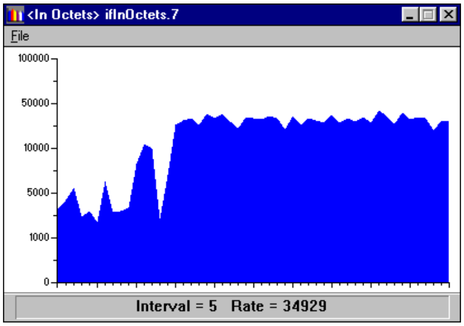
To copy entries from the log database to the clipboard, select the required items with your mouse, choose *Copy* or *Delete* from the Edit menu, and then choose Paste from the target application.

Data is logged directly into a log file associated with each process. The Log Database Manager only displays the data stored this file. The *New Arrival* line at the bottom of the dialog box indicates the number of events recorded into the log file since the last time data was retrieved by the Log Database Manager. To update the display, use the *Refresh* command.

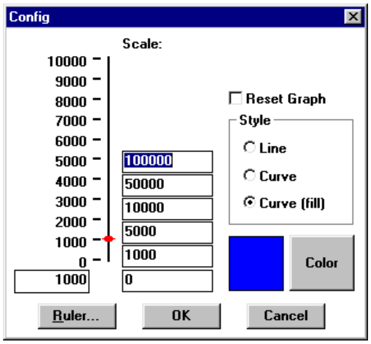
Using the Graphic Display Dialog Box

Open the graphic display from the Log Manager by selecting a process and clicking on the Graphic Data Display button in the toolbar. This dialog box displays a real-time graph of the count for data matched up to the current polling interval.

Click *Config* to change the display parameters. The Config Statistics dialog box appears. You can change the scale, style (i.e., filled curve, curve, line) and color, or reset the graph.



Interval = 5 Rate = 34929



☐ Reset Graph

Style: ○ Line ○ Curve ● Curve (fill)

100000
50000
10000
5000
1000
0

1000

Click *Config* to change the display parameters. The Config Statistics dialog box appears. You can change the scale, style (i.e., filled curve, curve, line) and color, or reset the graph.

To change the scale, you can adjust the display interval by sliding the red slider up or down on the ruler, by defining a specific interval in the edit box (located in the lower left corner of the dialog box), or by editing the value for each tick mark in the edit boxes next to the ruler. You can also change the ruler displayed for the Configuration dialog box by clicking on *Ruler...* and entering a new *Base* value and/or *Unit* of interval.

Be sure to reset the graph if necessary. Then click OK to return to the graph, or Cancel to abort any changes.

Defining Filter Formulas

Filters may be defined for any log process. A filter sets the conditions that determine if data received by the Log Manager will be saved into the log database. A filter is defined in the Filter field of the Log Information dialog box (i.e., the dialog box opened when you create or edit a log process). If no filter is defined, then all the data received is automatically logged.

Filter Formula

The filter formula uses Backus-Naur Form (BNF) as follows:

```
Filter::=                 SimpleExpression|
                          ComplexExpression|
                          <NULL>  -- Nothing

SimpleExpression::=Variable rel_op Value

Variable::=   "VALUE"|    --Value of the data
              "DATE" |    --Date the data arrives
              "TIME"      --Time the data arrives

rel_op::=     ">"  |      --Greater than
              "<"  |      --Less than
              ">="|       --Greater than or Equal to
              "<="|       --Less than or Equal to
              "=="|       --Equal
              "!="|       --Unequal
```

```
Value::=<INTEGER VALUE>|  --Number represented in decimal digits, within the range
                            of a 4-byte unsigned integer.
             <yymmdd>|    --Eight digits representing a date.
             <hhmmss>     --Eight digits representing a time, in 24 hour format

ComplexExpression::=
  "(" SimpleExpression ")" |
  "(" ComplexExpression ")"|
  ComplexExpression logic_op ComplexExpression

logic_op::="AND"|   "OR"
```

Filter Formula Syntax

The filter formula can be a simple or complex expression.

Syntax for Simple Expressions

A simple expression conforms to the following syntax:

`Variable Relation Value`

`Variable` - Legal variables include:

`VALUE` -- Value of the data
`DATE` -- Date the data arrives
`TIME` -- Time the data arrives

`Relation - Legal relations include:`

> -- Greater than
< -- Less than
>= -- Greater than or Equal to
<= -- Less than or Equal to
== -- Equal
!= -- Unequal

Value - Legal values include:

<INTEGER VALUE> -- An unsigned integer, 0 ~ 4 bytes long.
<yyyymmdd> -- Eight digits representing a date, where yyyy stands for the year, mm the month and dd the day
<hhmmss> -- Six digits representing time in 24-hour format, where hh stands for hour, mm minutes and ss seconds

Syntax for Complex Expressions

A complex expression combines several simple expressions using logical operators. Each expression must be enclosed in parentheses. The syntax for a complex expression follows:

(Simple Expression) Logical_Operator (Simple Expression)

As described in the previous section a simple expressions should have three basic elements, namely, a Variable, Relation and Value. Legal values for each element are described under the syntax for simple expressions.

Logical_Operator - Legal values include:

AND	-- Both expressions must be true to meet a given condition
OR	-- One true expression is enough to meet a given condition

Elements of Filter Formulas

Parameters	Example	Description
Variable	VALUE	**Value** of the data **Date** when the value is reached **Time** when the value is reached
Relation	> < >= <= == !=	Greater than Less than Greater than or equal to Less than or equal to Equal Not equal
Value	256 19990229 135501	Positive number (4 byte unsigned integer) Date format (YYYYMMDD) for 2 Feb 1999 Time format (HHMMSS) for 01:55:01pm
Logical_Operator	AND OR	Both statements **must** be true Either statement **may** be true

Notes: 1. The equal to (==) and unequal (!=) expressions follow C language syntax.

2. If more than one expression is used to define a filter, first enclose each expression in parentheses and then combine them with `AND` or `OR`. Parentheses are used to maintain the order of evaluation. Otherwise conditions are evaluated from left to right.
3. If you want to test a log process without saving data into the database, set the filter formula to an impossible condition; for example, (TIME<000000).

Example: Filter Formulas

Some possible filter formulas include:

- `VALUE > 10000`
- `(VALUE > 100) AND (VALUE < 10000)`
- `((DATE < 19990701) AND (TIME>120000)) OR`
- `((DATE >=19990701) AND (TIME<120000))`

In the last formula, data is filtered on (before 1 July 1994 after 12 noon) or (after 1 July 1994 before 12 noon).

A formula follows this basic syntax:

```
(Variable Relation Value Logical_Operator
(Variable Relation Value)
```

Defining Threshold Formulas

Thresholds are used to trigger events (which are defined in the Event Manager). EliteView's powerful Event Manager allows you to define an unlimited number of events corresponding to specific actions. See the chapter 8, Managing Events for more information on defining event response procedures.

EliteView's Log Manager uses thresholds to trigger an event. For example, a critical event can be handled whenever CRC errors exceed 5 per minute.

To set thresholds triggering certain events,

1. Define an event using the Event Manager, specifying an Event Name and Event Action.
2. Define threshold limits using the Log Manager. Fill in the Threshold field with the appropriate formula. Also fill in other necessary fields in the Log Information dialog box.

Threshold vs. Filter Formula

A threshold formula is very similar to the filter formula. In a threshold formula, the value of the data or the data rate can be used. For example, a threshold formula can monitor value fluctuations based on rates per second, per minute, or per hour.

Accuracy

Data rates greater than one per second are accurate. In computing data rates per second, the Log Manager calculates an average between each two consecutive data units.

Threshold Formula

The threshold formula uses Backus-Naur Form (BNF) as follows:

```
Threshold::=  SimpleExpression|
              ComplexExpression|
              <NULL>-- Nothing
```

```
SimpleExpression::=Variable rel_op Value

Variable::=   "R"|          -- Value of data
              "H"|                 – Changes in data per hour
              "M"|                 – Changes in data per minute
              "S"                  -- Changes in data per second

rel_op::=     ">" |         – Greater than
              "<" |         -- Less than
              ">="|         – Greater than or Equal to
              "<="|         – Less than or Equal to
              "=="|         -- Equal
              "!="|         – Unequal
```

`Value::=<INTEGER VALUE>|` -- Number represented in decimal digits, within the range of a 4-byte unsigned integer.

```
ComplexExpression::=
  "(" SimpleExpression ")"|
  "(" ComplexExpression ")"|
  ComplexExpression logic_op ComplexExpression

logic_op::="AND"|"OR"
```

Threshold Formula Syntax

The threshold formula can be a simple or complex expression.

Syntax for Simple Expressions

A simple expression follows the following syntax:

`Variable Relation Value`

`Variable` - Legal variables include:

`R` -- Value of data
`H` -- Changes in data per hour
`M` -- Changes in data per minute
`S` -- Changes in data per second

`Relation` - Legal relations include:

`>` – Greater than
`<` -- Less than

`>=` -- Greater than or Equal to
`<=` -- Less than or Equal to
`==` -- Equal
`!=` -- Unequal

`Value` - Legal values include:

`<INTEGER VALUE>` -- An unsigned integer, 0 ~ 4 bytes long.

Syntax for Complex Expressions

Refer to the same section under Filter Formulas Syntax.

Elements of Threshold Formulas

Parameters	Example	Description
Variable	R	Actual value of the data
	H	Data rate per **hour**
	M	Data rate per **minute**
	S	Data rate per **second**
Relation	>	Greater than
	<	Less than
	>=	Greater than or equal to
	<=	Less than or equal to
	==	Equal
	!=	Not equal
Value	256	Positive number (4 byte unsigned integer)
	19990229	Date format (YYYYMMDD) for 2 Feb 1999
	135501	Time format (HHMMSS) for 1:55:01pm
Logical_Operator	AND	Both statements **must** be true
	OR	Either statement **may** be true

Example: Threshold Formulas

Some possible threshold formulas include:

- `R > 1000`
- `(M > 5) OR (H > 500)`
- `(S > 100) AND (R < 10000)`

In the last formula, an event is triggered if (the data is changing at the rate greater than 100 per second) or (the value is less than 10,000).

A formula follows this basic syntax:

(Variable Relation Value) Logical_Operator (Variable Relation Value)

CHAPTER 8

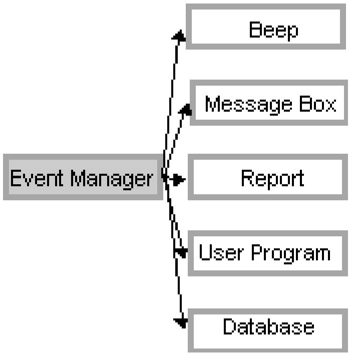
MANAGING EVENTS

Understanding the Event Manager

Events are special conditions which occur during normal operation of network devices, and normally require the network manager's attention. Events are generated by the main EliteView program, the Log Manager, and the Trap Manager.

In response to messages from the main EliteView program, the Trap Manager or Log Manager, the Event Manager can dispatch actions in any of five different ways.

- Sound audible signal
- Display on-screen message box
- Log event into an on-screen report window
- Run any user-specified program, e.g., BEEPER, FAX, EMAIL, etc.
- Log events into a database for later analysis



Unlike simplistic network monitoring, EliteView's Event Manager can accept input and process the event according to pre-defined rules.

Starting the Event Manager

The Event Manager automatically starts when the main EliteView program is invoked, and when any system or user-defined event occurs. If the Event Manager is closed or hidden, simply select Event Manager from the Utilities menu of main EliteView program or click on the Event Manager icon in the EliteView program group to bring it up.

Defining Events

The Event Manager supports system and user events, as indicated in the Type field for the selected event.

Pre-Defined "System" Events

Pre-defined system events include the following:

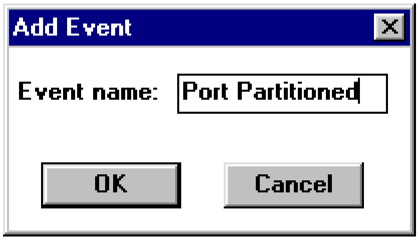
- *Connection Lost* - a device fails to respond after the specified number of retires.
- *Device Up* - a device which has previously been "down" is now responding.
- *TFTP* - a device is currently performing file transfer or download.
- *Trap* - a device has issued a trap message and it has been received by the Trap Manager. The Trap Manager translates the trap message into a readable text string.

Defining "User" Events

In addition to the pre-defined system events, the Event Manager supports user-definable events.

To add an event to the Event Manager,

1. From the EliteView main program window, choose *Event Manager* from the Utilities menu.
2. Click on the Add button.
3. In the Add Event dialog box, enter a descriptive name of the event.



Event name: Port Partitioned

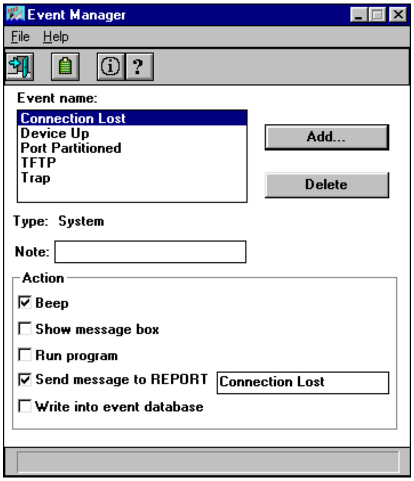
4. Click on the *OK* button to accept the new event or click on Cancel to abort this action.

Defining Event Actions

In response to any system or user events, the Event Manager can be used to take specific actions. The five basic actions are:

- Sound audible signal
- Display on-screen message box
- Log event into an on-screen report window
- Run any user-specified program, e.g., BEEPER, FAX, EMAIL.
- Log event into a database for later analysis

Press this button to view the event database



Event name:
Connection Lost
Device Up
Port Partitioned
TFTP
Trap

Type: System

Note:

☑ Beep
☐ Show message box
☐ Run program
☑ Send message to REPORT Connection Lost
☐ Write into event database

To define an action:

1. Click on the event name
2. Enter annotation in the Note field (optional)
3. Click on one or more Actions (Beep, Show, Run, Send, Write)

Actions	Description	Example
Beep	Sound an audible signal on the network management station (i.e., your local PC).	
Show message box	Display a user-definded message in a text box on the NMS's screen. The message box appears on top of all the other windows. When the message appears, just click on the OK button to dismiss the message. This action is recommended for critical events.	CRITICAL: Hub overheating
Run program	Execute any Windows-based application. This is ideal for sending a message to a pager, email or FAX.	PAGER 408-555-4742
Send message to REPORT	Pass a message to the EliteView REPORT module. Report messages are time stamped and shown in chronological order. In addition, Report messages are shown in context with one another. This action is recommended for routine events.	WARNING: Server disk > 90% full
Write into database	Put a message into the event database	Excessive CRC errors on device

Example: Displaying a text file

When an important event occurs, you may display instructions for others to follow. For example, a text file named URGENT.TXT might include information on how to contact key personnel who can fix the problem (telephone number, pager number, etc.).

To display information with Windows Notepad:

1. Open the *Event Manager*.
2. Choose an event.
3. Click on the Run Program action and type

```
NOTEPAD [PATH]URGENT.TXT
```

4. Use the Windows Notepad program to create the file URGENT.TXT in the indicated directory.

When the specified event occurs, the text file URGENT.TXT will be automatically displayed using Notepad.

Using Special Text

The Event Manager supports special text messages using the $$ and ## symbols in the text boxes.

- $$ is substituted with text provided by the source of the event. For example, Trap Manager will pass a text string translated from the trap message provided by the device. If no trap messages are available, then $$ will be null.
- ## is substituted with the network address of the device.

Example: Logging detailed messages

To log a detailed message in the database (or message box or report window), use the special text substitutions (i.e., $$ or ##).

To make an entry in the database like
`"192.75.255.32 TEMPERATURE EXCEEDS 85 DEGREES",`

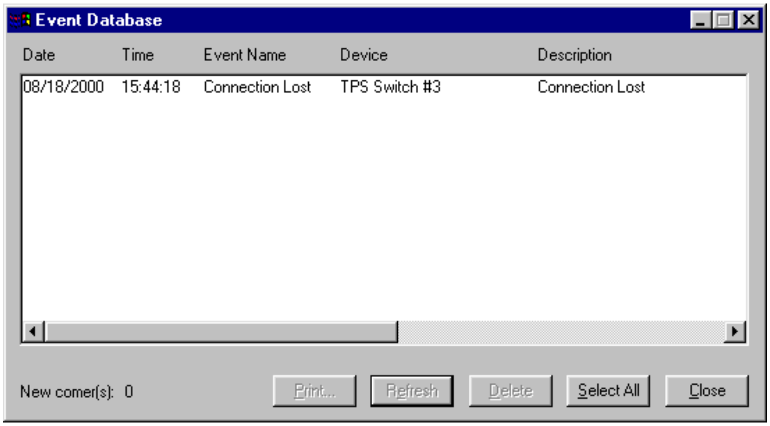
1. Choose an event name.
2. Check the box for Write into database.
3. In the adjacent text box, enter - **`## $$`**

Event Data

Event data is stored in a dBASE-compatible file (EVENT_DT.DBF).

To view event data from the Event Manager:

1. Double-click on an event name, or click on the *View Data* button.
2. The Event Data window will show a listing of events.



Date	Time	Event Name	Device	Description
08/18/2000	15:44:18	Connection Lost	TPS Switch #3	Connection Lost

New comer(s): 0

Parameters	Description	Example
Date	Date when the event occurred	07/16/1999
Time	Time when the event occurred	13:33:48 (1:33 pm)
Device	IP network address of the device	192.9.211.52
Description	Text from "Write into database"	Dropped server connection

3. Click the Close button to close the Event Data window.

To erase some or all event information:

1. From the Event Data window, select the target event line. To delete all event information, click the *Select All* button. The selected entries are highlighted.
2. Click on the *Delete* button to delete highlighted events.

To print information from the event database:

1. From the Event Database window, click on the *Print* button.
2. Select the required events.
3. To set the print pointer at the beginning of the database, mark the radio button for *From Beginning*. To set the pointer at the location where the last print operation terminated, mark the radio button for *From Last Printed Data*.
4. Edit the *From* and *To* times if required.
5. Use the *Print Setup* button to verify your printer settings.
6. Press the *Print* button and then press *Close*.

Note: Event logs may be viewed by any application that can import a dBASE (.DBF) file, such as FoxPro.

Receiving SNMP Traps with the Trap Manager

Trap is a protocol mechanism defined in SNMP by which managed devices report unique events to the network management station. Devices can be set up to report specified conditions to EliteView using Trap messages.

Limitations of Trap Messages

Trap messages are designed to report information that requires immediate attention. However, the value of Traps is limited in SNMP because there is no guarantee these messages will be delivered. Trap messages may be lost en route or ignored by the network manager.

Trap Types

There are two types of traps defined in the SNMP standard.

- *Generic trap* is supported by all SNMP-compliant devices. The most common events include cold start, warm start, link down, link up, SNMP authentication failure, and EGP neighbor loss. (EGP is the Exterior Gateway Protocol used to exchang routing information.) Generic traps, numbered 0 to 5 in the “generic trap” field of the trap message, are defined in TRAP.INI.
- *Specific trap* is supported according to the characteristics of the device. Specific traps are numbered 6 in the “generic trap” field. A “specific trap” field identifies the trap type.

Trap Manager

EliteView's Trap Manager collects trap messages and converts them into events. The Trap Manager generates a "trap" event and outputs a text message according to the pattern specified in TRAP.INI. These events are then handled by the Event Manager.

If the TRAP.INI file is changed, re-boot the Trap Manager by closing it and then starting it again.

The Trap Manager has no tangible user interface. When the main EliteView program is started, if the Trap Manager is not loaded automatically (as defined in NETMGR.INI), then you can load it from the Utilities menu, or from the Window's Program Manager (by clicking on the Trap Manager icon in the EliteView program group). When EliteView terminates, the Trap Manager is also closed.

Posting Messages to the Report Window

Both predefined and user-defined system activity may be specified in the Event Manager to be posted to EliteView's Report window. If any event criteria have been met, then this window will display a chronological list of pertinent messages stamped by time and date.

To view the Report window, select Report from the Utilities menu of the main EliteView program, or click on the Report icon in the EliteView program group.

The Report window shows user-definable messages in chronological order. Entries placed in the Report window by the Event Manager include connection lost, device up, tftp, and trap messages.

Field Description for Report Menus

Menu	Function
File	**Open** - Opens any previously saved Report file. **Save** - Saves the current Report file. **Save As** - Save the current Report file under a new name. **Print** - Prints the current file. **Exit** - Closes the Report window.
Edit	**Copy** - Copies the selected data into the Windows clipboard. **Clear** - Deletes all information from the Report window.
Search	**Find** - Searches for the specified data. **Find Next** - Searches the next occurance of the specified data. **Find Previous** - Searches the previous occurance of the specified data.

To copy from Report window to another Windows application,

1. Drag the mouse over the target text.
2. From the Edit menu, choose *Copy*.
3. Switch to another Windows application.
4. From the target application's main menu, choose *Paste* from the Edit menu. The highlighted text from EliteView Report will now appear in the target application.

CHAPTER 9
USING RMON

Introduction

This chapter describes how to use Remote Monitoring (RMON) to more effectively monitor your network. RMON provides a cost-effective way to monitor large networks by placing embeded or external probes on distributed network equipment (i.e., hubs, switches or routers). EliteView can access the probes embedded in recent SMC network products to perform traffic analysis, troubleshoot network problems, evaluate historical trends, or implement proactive management policies. RMON has already become a valuable tool for network managers faced with a quickly changing network landscape that contains dozens or hundreds of separate segments. RMON is the only way to retain control of the network and analyze applications running at multi-megabit speeds. It provides the tools you need to implement either reactive or proactive policies that can keep your network running based on real-time access to key statistical information.

RMON can be used to perform a wide range of mangement tasks, including:

- Troubleshoot problems
- Track down intermittent problems
- Locate bottlenecks
- Plan for network expansion

A Brief Description of RMON

Remote Monitoring allows you to instruct a remote device to collect information or respond to specified events on an independent basis. An RMON-capable device can independently perform a wide range of tasks, significantly reducing network management traffic. It can continuously run diagnostics and log network performance. If an event is triggered, the remote device can automatically notify the network administrator of a failure and provide historical information about the event. If the remote device cannot connect to the management agent, it will continue to perform any specified tasks and pass data back to the management station the next time it contacts the remote device.

RMON is designed to limit the amount of traffic required by management applications. It consists of an independent agent that resides on the managed device, and is charged with monitoring and collecting information about network traffic or the status of the host device. The agent gradually builds up information about the attached segment or VLAN, storing this information in the relevant RMON database group. The client (or mangement agent) then periodically communicates with the various RMON probes using SNMP protocol. It can then present a numeric or graphic summary of the data collected from a large number of probes. However, if the probe encounters a critical event (as defined by the management agent), it can automatically send a trap to the management agent which will then respond to the event as determined by the Event Manager (see Chapter 8).

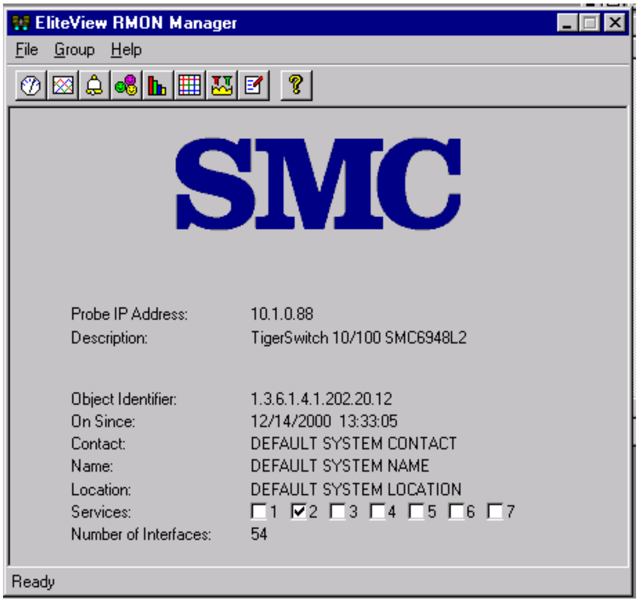
Starting the RMON Manager

To use the RMON Manager, open any network map and select the RMON program from the menu bar . You can also run the RMON Manager directly from the Start Menu by selecting the RMON Manager icon directly from the EliteView program group.

When you start the RMON Manager, a Probe Information window will pop up requesting target information. You must provide the following information:

1. Enter the Target Address of the device.
2. Define a Community name describing the administrative relationship (i.e., access rights) between SNMP entities.

If you open the main application for the RMON Manager, the screen will display a detailed description of the managed device as shown in the following example for a TigerSwitch SMC6948L2



Probe IP Address: 10.1.0.88
Description: TigerSwitch 10/100 SMC6948L2

Object Identifier: 1.3.6.1.4.1.202.20.12
On Since: 12/14/2000 13:33:05
Contact: DEFAULT SYSTEM CONTACT
Name: DEFAULT SYSTEM NAME
Location: DEFAULT SYSTEM LOCATION
Services: ☐1 ☑2 ☐3 ☐4 ☐5 ☐6 ☐7
Number of Interfaces: 54

Field	Description
Probe IP Address	The IP address of the RMON probe.
Description	Description of the device, including manufacturer and model name.
Object Identifier	The object identifier used to identify this device in the MIB tree.
On Since	The time at which the device was turned on.
Contact	The person responsible for managing this device.
Name	Name used for this device; such as a hierarchical network name.
Location	Physical location of the device.
Services	Network services provided by the device, as specified in the seven-layer network protocol of the Open Systems Interconnection.
Number of Interfaces	Number of port interfaces provided by the device.

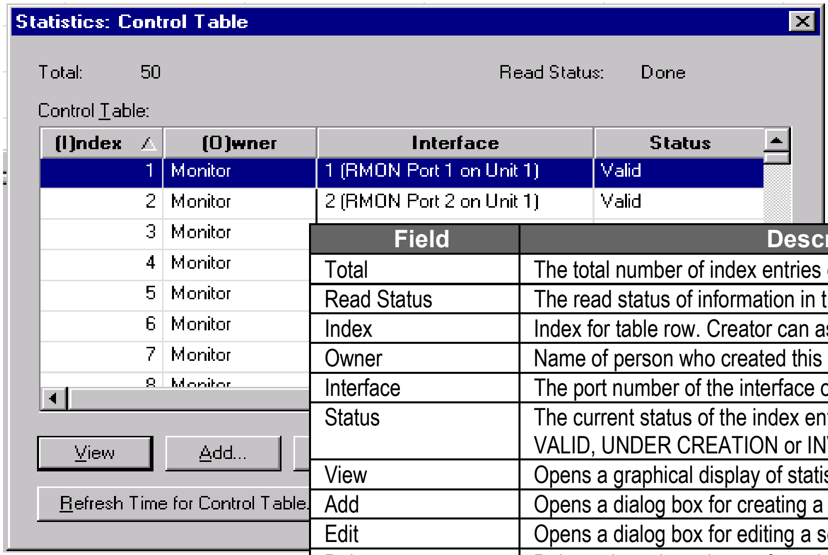
RMON Utilities

The RMON Manager currently provides access to all nine RMON groups, as shown in the following table. Most of SMC's intelligent products provide support for mini-RMON (which include Statistics, History, Alarms and Events). This selection covers key information required to manage your network on a regular basis, and also switches, and especially on all backbone switches. An external RMON probe that supports all nine RMON groups can then be used for extended troubleshooting when needed.

Group	Description
Statistics	Shows bandwidth utilization, counters for network traffic, errors and collisions, as well as packet size distribution
History	Periodically samples and saves information from the statistics group.
Hosts	Maintains statistics on each host attached to the network device monitored by the probe.
Host Top N	Displays a specified subset of statistics for a selected number of top users.
Matrix	Maintains statistics on traffic passing between node pairs.
Alarms	Sets thresholds for RMON variables, which can subsequently trigger response events.
Events	Defines the action to take when an alarm is triggered, including logging the alarm, generating a trap message, or triggering a capture channel.
Filters	Filters a stream of packets that match the specified criteria.
Capture	Configures the buffers used to store packets generated from the Filter group.

Statistics Group

The Statistics Group includes all the tools you need to monitor your network for common errors and overall traffic rates. When you open the Statistics Group the Control Table screen is displayed as shown below. This table allows you to add, edit and delete items to be monitored, or to select a specific index entry and then view the statistical data in numeric or graphic form.



Total: 50

Read Status: Done

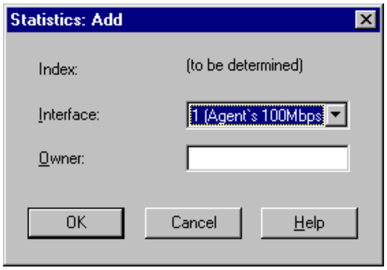
Control Table:

(I)ndex	(O)wner	Interface	Status
1	Monitor	1 (RMON Port 1 on Unit 1)	Valid
2	Monitor	2 (RMON Port 2 on Unit 1)	Valid
3	Monitor		
4	Monitor		
5	Monitor		
6	Monitor		
7	Monitor		
8	Monitor		

Field	Description
Total	The total number of index entries contained in the Control Table.
Read Status	The read status of information in the Statistics Control Table.
Index	Index for table row. Creator can assign a value from 1 to 65,535.
Owner	Name of person who created this entry in the Control Table.
Interface	The port number of the interface on the device.
Status	The current status of the index entry in the Control Table: VALID, UNDER CREATION or INVALID
View	Opens a graphical display of statistics for the selected index entry.
Add	Opens a dialog box for creating a new entry in the Control Table.
Edit	Opens a dialog box for editing a selected entry in the Control Table.
Delete	Deletes the selected entry from the Control Table.

Adding or Editing an Entry in the Control Table

Click on the Add (Edit) button in the Statistics Control Table to add (edit) an index entry. The dialog box that opens includes three fields (1) entry index number, (2) system interface number, and (3) owner of this entry. The system automatically generates an index number, but you can enter any number from 1 to 65,535 that is not currently in use. Each interface number equates to a physical media on the device being monitored. This information can be found under MIB2 as shown to the right. (See MIB Browser in Chapter 6.)



Index: (to be determined)
Interface: 1 (Agent's 100Mbps
Owner:

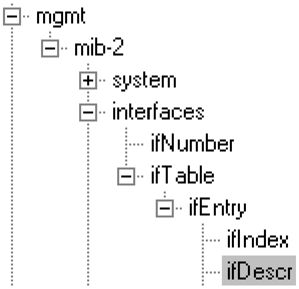
The interfaces to the SMC6508 switch and SMC5912/24 dual-speed hub are listed in the tables below.

SMC6508 Interface Description

Interface	Description
1 - 8	Ports 1 - 8 (10 Mbps port)
9	Port 9 (10/100 Mbps port)
10	Port 10 (100BASE-TX or 100BASE-FX port)

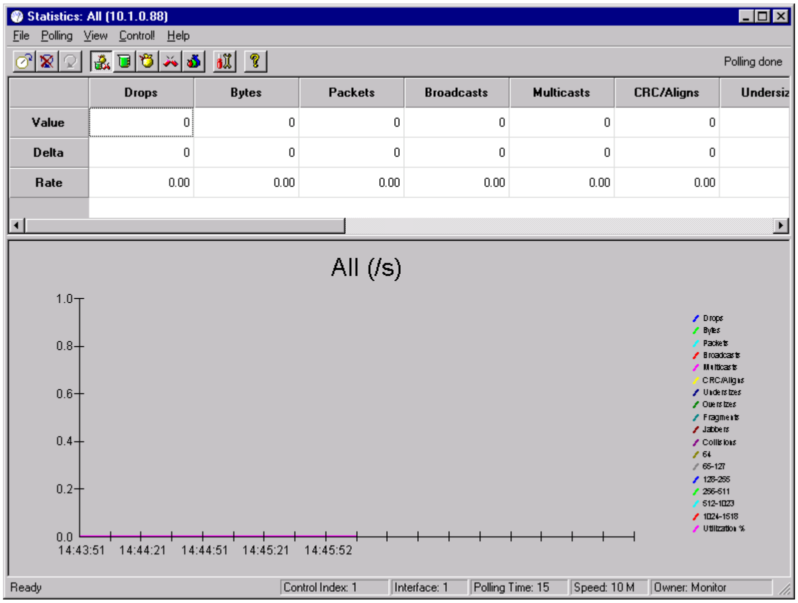
SMC5912/5924 Interface Description

Interface	Description
1	Agent board 100 Mbps interface
2	Agent board 10 Mbps interface
3	Serial port used for out-of-band connection
4	100 Mbps stack backplane
5	10 Mbps stack backplane
6	Stack Unit 1 100 Mbps segment
7	Stack Unit 1 10 Mbps segment
⋮	
16	Stack Unit 6 100 Mbps segment
17	Stack Unit 6 10 Mbps segment



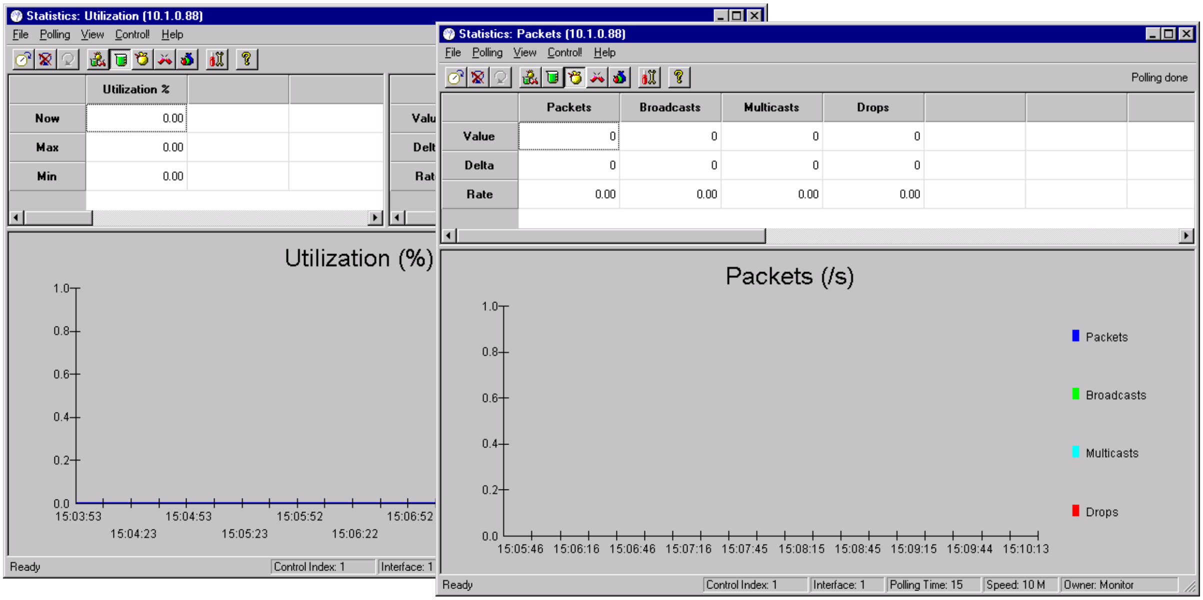
Viewing Statistics

The RMON Manager collects statistics that allow you to quickly determine how the network is performing. Information is provided on bandwidth utilization, packet types, errors and collisions, as well as the distribution of packet sizes. Information is also included on peak utilization. Statistics are displayed in both a numeric and graphic format that can be easily interpreted. To display statistics for a specific entry, click on the View button in the Control Table. The various statistic screens are shown below. The scale for the graphic displays are automatically adjusted to present the best view possible. (However, note that the smallest increment on the vertical axis is limited to 0.05.). If necessary, you can reduce the polling interval to focus in a specific problem area, or increase it to reduce the overall management traffic running on the network.



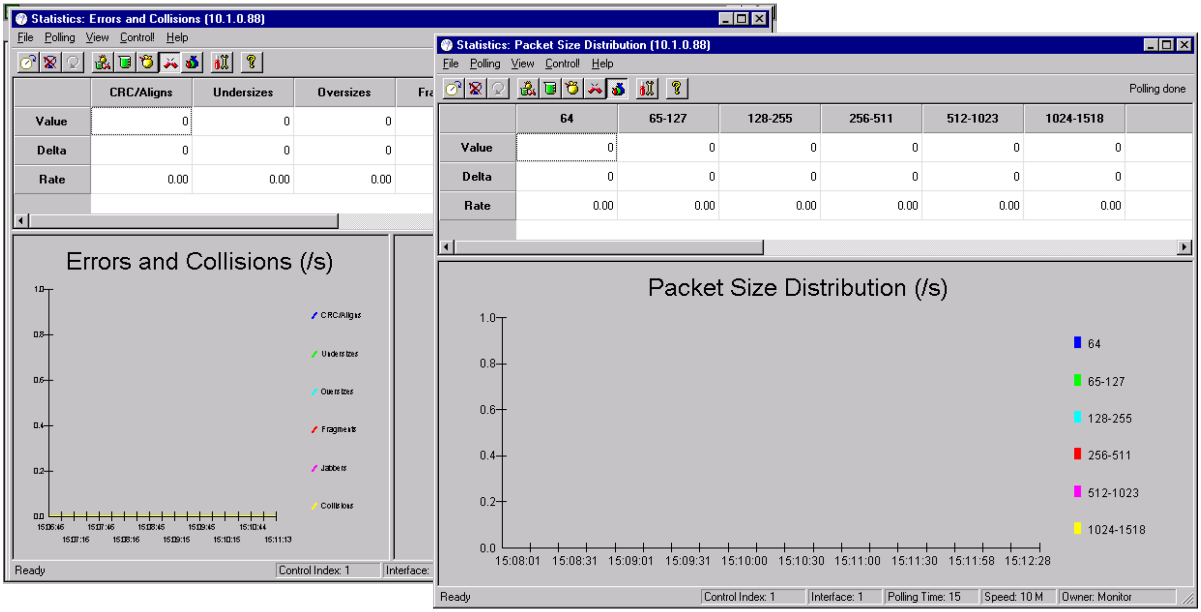
	Drops	Bytes	Packets	Broadcasts	Multicasts	CRC/Aligns	Undersiz
Value	0	0	0	0	0	0	
Delta	0	0	0	0	0	0	
Rate	0.00	0.00	0.00	0.00	0.00	0.00	

Control Index: 1 | Interface: 1 | Polling Time: 15 | Speed: 10 M | Owner: Monitor



	Utilization %
Now	0.00
Max	0.00
Min	0.00

	Packets	Broadcasts	Multicasts	Drops
Value	0	0	0	0
Delta	0	0	0	0
Rate	0.00	0.00	0.00	0.00



	CRC/Aligns	Undersizes	Oversizes
Value	0	0	0
Delta	0	0	0
Rate	0.00	0.00	0.00

	64	65-127	128-255	256-511	512-1023	1024-1518
Value	0	0	0	0	0	0
Delta	0	0	0	0	0	0
Rate	0.00	0.00	0.00	0.00	0.00	0.00

Statistics are provided for the following areas

Field	Description
Utilization	Displays the percentage of bandwidth utilized over the sample period. It also shows the total count, the rate, and the rate of change (delta) for packets and bytes seen on the interface.
Packets	Displays the total count, the rate, and the rate of change (delta) for all packets, broadcasts, multicasts, and dropped packets.
Errors and Collisions	Displays the total count, the rate, and the rate of change for CRC/alignment errors, undersize packets, oversize packets, fragments, jabbers, and collisions.
Packet Size	Displays the distribution of packet sizes, including the total count, the rate, and the rate of change for each packet size.

The statistics for each area are further described in the following table.

Field	Description
Utilization	
Util% (Now, Max, Min)	This table displays the current bandwidth utilization, plus the maximum and minimum utilization since the statistics window was opened.
Packets/Bytes[1]	This table displays packets and bytes.
Packets[1]	
Packets	Packets (including bad, broadcast and multicast packets).
Broadcasts	Broadcast packets.
Multicasts	Multicast packets.
Drops	The number of events detected when packets were dropped (due to a lack of resources condition in the probe.
Errors and Collisions[1]	
CRC/Alignment	Packets with a CRC or alignment error.
Undersizes	Undersize packets.
Oversizes	Oversize packets.
Fragments	Packet fragments.
Jabbers	Jabber errors.
Collsions	Packet collisions.
Packet Size Distribution[1]	
Size	Packet sizes are divided into six groups, 64 byte packets, 65 to 127, 128 to 255, 256 to 511, 512 to 1023, and 1024 to 1518 bytes. Each size group is indicated as color bar for each polling time.

Note: Value - Total since the Control Table entry was created.
Delta - The difference in the count since the last poll.
Rate - The rate per second over the last polling interval.

Each of these windows includes a Menu bar and Tool bar with the following items:

Field	Description
File	Exit
Polling	Polling Time (5 - 3600 seconds), Pause, Resume
View	All, Utilization, Packets, Errors and Collisions, or Packet Size Distribution
Control!	Control Table
Help	On-line help

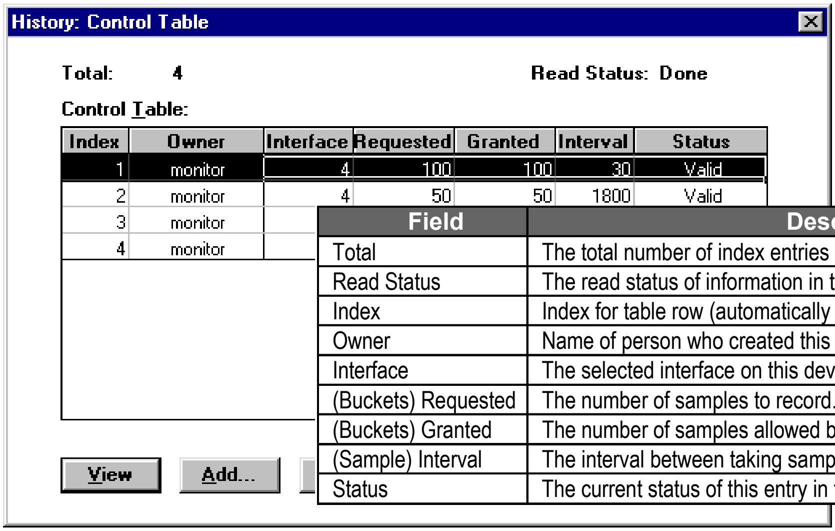
They also include a status bar at the bottom of the window that includes the following items:

Field	Description
Control Index	The index for the entry defined in the Statistics Control Table.
Interface	The media interface of the device being monitored.
Polling Time	The interval between polling samples.
Speed	The speed of the media interface being monitored.
Owner	The person who created this entry.

History Group

The History Group can be used to create a record of network utilization, packet types, errors and collisions. You need a historical record of activity to be able to track down intermittent problems. Historical data can also be used to establish normal baseline activity, which may reveal problems associated with high traffic levels, broadcast storms, or other unusual events. Historical information can also be used to predict network growth and plan for expansion before your network becomes too overloaded.

The History Group includes all the tools you need to monitor your network for common errors and overall traffic rates. When you open the History Group the Control Table screen is displayed as shown below. This table allows you to add, edit and delete collection entries, or to select a specific index entry and then view the historical data in numeric or graphic form.



Total: 4

Read Status: Done

Control Table:

Index	Owner	Interface	Requested	Granted	Interval	Status
1	monitor	4	100	100	30	Valid
2	monitor	4	50	50	1800	Valid
3	monitor					
4	monitor					

Field	Description
Total	The total number of index entries contained in the Control Table.
Read Status	The read status of information in the Statistics Control Table.
Index	Index for table row (automatically assigned).
Owner	Name of person who created this entry in the Control Table.
Interface	The selected interface on this device (as defined in MIB2).
(Buckets) Requested	The number of samples to record. (Default: 50)
(Buckets) Granted	The number of samples allowed by the system.
(Sample) Interval	The interval between taking samples. (Default: 1800 seconds)
Status	The current status of this entry in the Control Table.

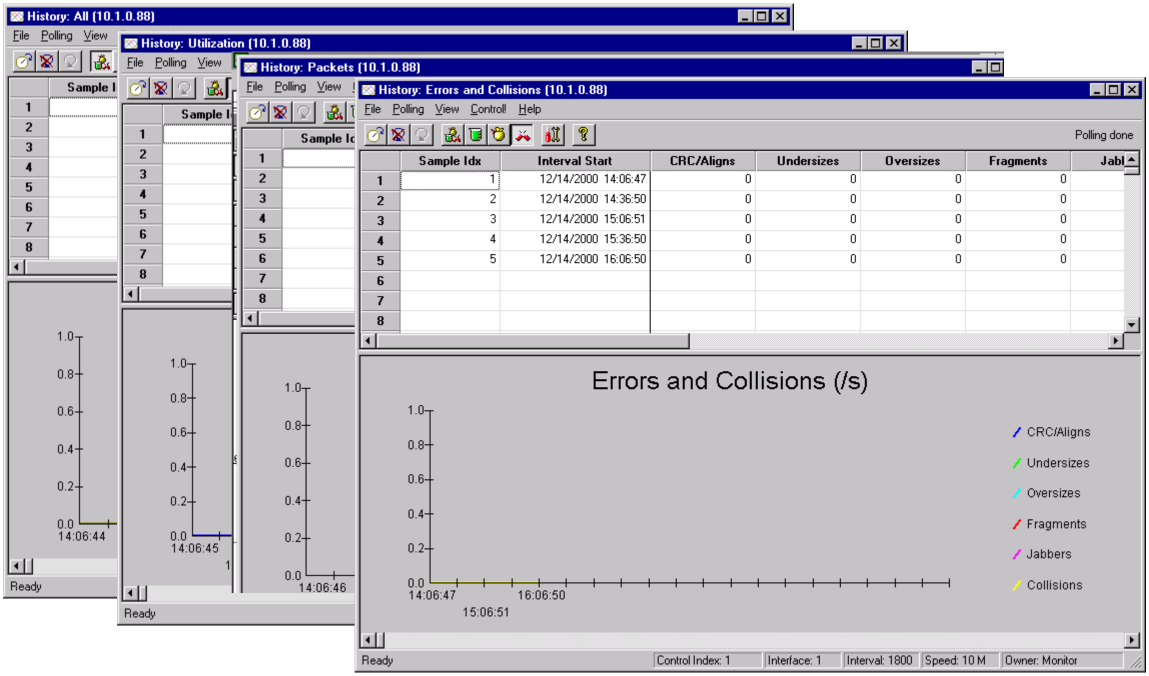
Adding or Editing an Entry in the Control Table

Click on the Add (Edit) button in the History Control Table to add (edit) an index entry as shown on the preceding page. Each interface equates to a physical media on the device being monitored. This information can be found under MIB2 (see page 6-5). The number of buckets indicates the number of samples to record. You should always be able to record the default of 50 buckets without overflowing the agent's memory. However, if more than 50 buckets are requested, the number actually granted will depend on the amount of memory currently available. Once all the buckets are filled, older data will be discarded. You also need to specify the interval at which to take samples. For example, using a 30 second interval with 120 buckets will provide one hour of historical samples.

Viewing History

The RMON Manager collects historical information on bandwidth utilization, packet types, errors and collisions, as well as the distribution of packet sizes for each entry you define. Each sample shows the interval's start time. The information collected during the interval is displayed in both numeric and graphic format for easy interpretation.

To display the history for a specific entry, highlight that entry in the Control Table and click on the View button. The various history screens are shown below. The scale for the graphic displays are automatically adjusted to present the best view possible. If necessary, you can reduce the polling interval to focus in a specific problem, or increase it to reduce overall management traffic running on the network.



	Sample Idx	Interval Start	CRC/Aligns	Undersizes	Oversizes	Fragments	Jabl
1	1	12/14/2000 14:06:47	0	0	0	0	
2	2	12/14/2000 14:36:50	0	0	0	0	
3	3	12/14/2000 15:06:51	0	0	0	0	
4	4	12/14/2000 15:36:50	0	0	0	0	
5	5	12/14/2000 16:06:50	0	0	0	0	
6							
7							
8							

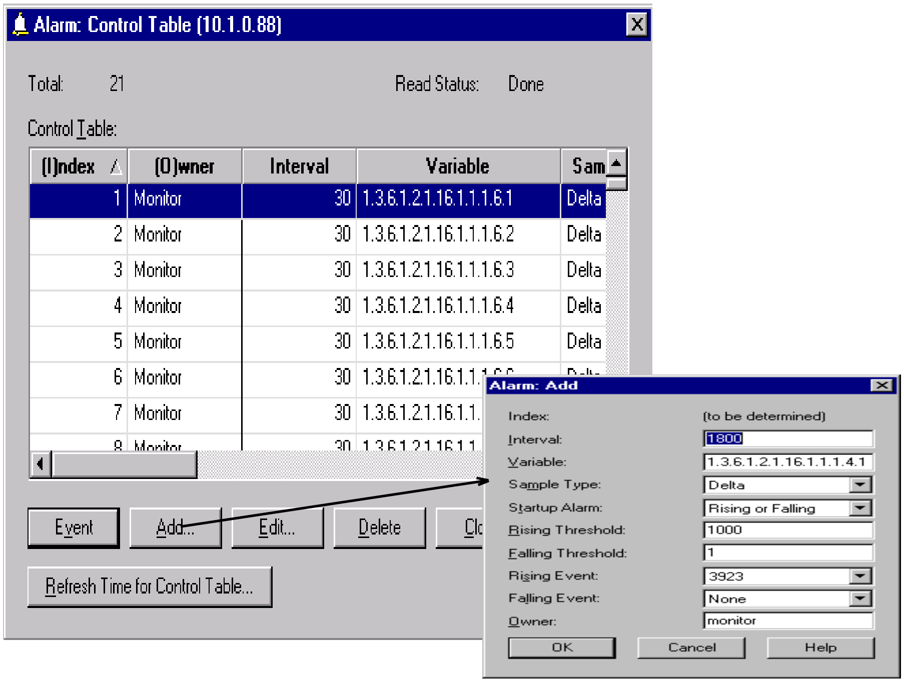
Control Index: 1 | Interface: 1 | Interval: 1800 | Speed: 10 M | Owner: Monitor

Alarm and Event Groups

The Alarm and Event Groups allow you to record important events or immediately respond to critical network problems. The Alarm and Event Control Tables (shown below) are used together to define specific criteria that will generate response events. (Note that you must use the scroll bar to display all the columns in the tables.) These tables allow you to add, edit and delete items, or to select a specific index entry and then view the corresponding response event (from the Alarm Table) or triggered events (from the Event table).

Adding or Editing an Entry in the Control Table

Alarm Control Table - Alarms can be set to test data over any specified time interval, and can monitor absolute or changing values (such as a statistical counter reaching a specific value, or a statistic changing by a certain amount over the set interval). Alarms can be set to respond to either rising or falling thresholds. (However, note that after an alarm is triggered it will not be triggered again until the statistical value crosses the opposite bounding threshold as described in the following table.) Click on the Add (Edit) button in the Alarm Control Table to add (edit) an index entry. The dialog box that opens is described in the following table.



Alarm: Control Table (10.1.0.88)
Event
Add...
Edit...
Delete
Cl
Refresh Time for Control Table...
Alarm: Add
OK
Cancel
Help

Total: 21

Read Status: Done

Control Table:

(I)ndex	(O)wner	Interval	Variable	Sam
1	Monitor	30	1.3.6.1.2.1.16.1.1.1.6.1	Delta
2	Monitor	30	1.3.6.1.2.1.16.1.1.1.6.2	Delta
3	Monitor	30	1.3.6.1.2.1.16.1.1.1.6.3	Delta
4	Monitor	30	1.3.6.1.2.1.16.1.1.1.6.4	Delta
5	Monitor	30	1.3.6.1.2.1.16.1.1.1.6.5	Delta
6	Monitor	30	1.3.6.1.2.1.16.1.1.	
7	Monitor	30	1.3.6.1.2.1.16.1.1.	

Index: (to be determined)

Interval: 1800

Variable: 1.3.6.1.2.1.16.1.1.1.4.1

Sample Type: Delta

Startup Alarm: Rising or Falling

Rising Threshold: 1000

Falling Threshold: 1

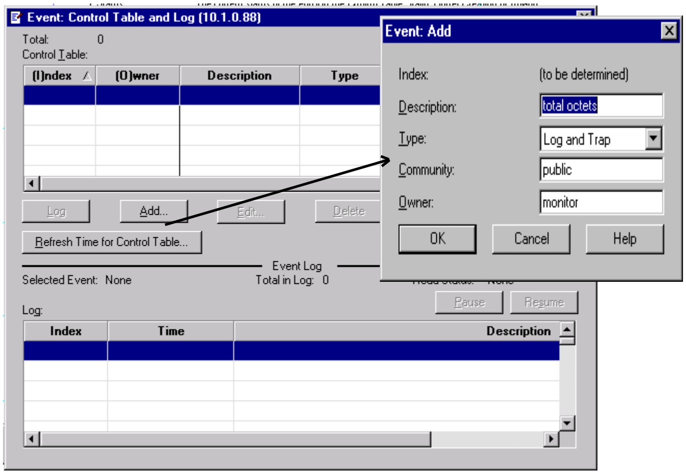
Rising Event: 3923

Falling Event: None

Owner: monitor

Field	Description
Index	The system automatically generates an index number.
Owner	The name of the person who configured the entry in the Control Table.
Interval	The time interval in seconds over which data is sampled and compared with the rising or falling threshold.
Variable	The object identifier of the MIB variable to be sampled. Only variables that resolve to an integer value may be sampled.
Sample Type	The method of sampling data, either Absolute or Delta. For an absolute sample the variable will be compared directly to the thresholds. For a delta sample the last sample is subtracted from the current value and the difference is then compared to the thresholds.
Startup Alarm	How the alarm is activated when the variable is compared to the thresholds. This can be set to Rising, Falling, or Rising or Falling.
Rising Threshold	An alarm threshold for the sampled variable. If the current value is greater than or equal to the threshold, and the last sample value was less than the threshold, then an alarm will be generated. (After a rising event has been generated, another such event will not be generated until the sampled value has fallen below the Rising Threshold and reaches the Falling Threshold.)
Falling Threshold	An alarm threshold for the sampled variable. If the current value is less than or equal to the threshold, and the last sample value was greater than the threshold, then an alarm will be generated. (After a falling event has been generated, another such event will not be generated until the sampled value has risen above the Falling Threshold and reaches the Rising Threshold.)
Rising Event	The index of the Event that will be used if a rising alarm is triggered. If there is no corresponding entry in the Event Control Table, or if this number is zero, then no event will be generated.
Falling Event	The index of the Event that will be used if a falling alarm is triggered. If there is no corresponding entry in the Event Control Table, or if this number is zero, then no event will be generated.
Status	The current status of the entry in the Control Table: Valid, Under Creation or Invalid

Event Control Table - If the response corresponding to the alarm has not yet been defined, click on the Event button to open the Event Control table. Click on the Add (Edit) button in the Event Control Table to add (edit) an index entry. The dialog box that opens is described in the following table.



Total: 0

Selected Event: None

Total in Log: 0

Index: (to be determined)

Description: total octets

Type: Log and Trap

Community: public

Owner: monitor

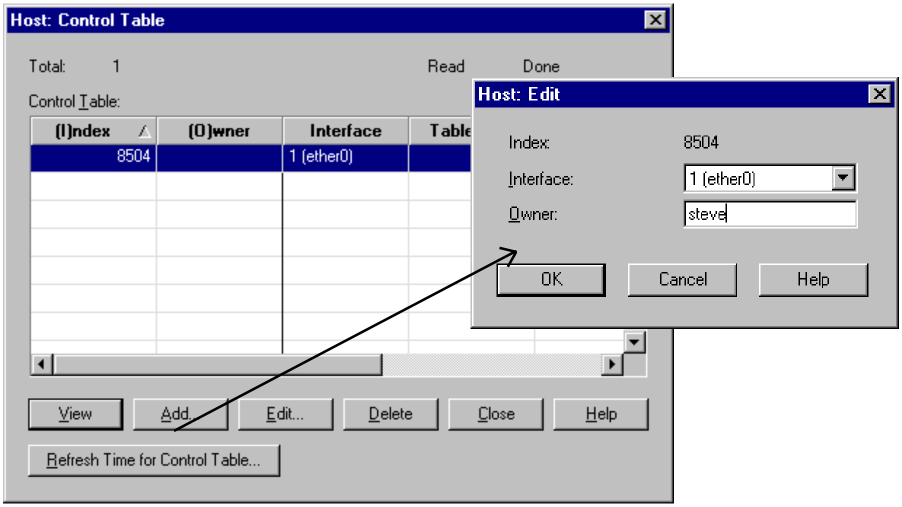
Field	Description
Index	A number that identifies the row in the table.
Description	A text comment that describes the entry in the Control Table.
Type	The type of action that is taken for the alarm. This can be None, Log, Trap, or Log and Trap.
Community	The SNMP community name that a trap manager must use to receive trap messages.

Displaying Events in the RMON Manager - The Event determines the action to take when an alarm is triggered. The response to an alarm can include logging the alarm or sending a message to a trap manager. To display each time an event was triggered by an alarm, first highlight an entry in the upper half of the Event Control Table, and then click on the Log Button . The Log Table at the bottom half of the Event Control Table window will display each time this event was triggered. It shows the log index number, the time of an event, and the description of the event that activated this entry. (Note that there are no display windows associated with the Alarm and Event groups other than the control tables.)

Host Group

The Host Group can maintain statistics on all devices found on the network, with the only limitation being the amount of available buffer space. A full set of statistics, as defined in the Statistics Group, can be maintained for each unique address. This group is generally used as one of the last steps in troubleshooting. For example, if a network device has triggered a predefined event, you can configure the RMON probe to collect host information for the media interface where the problem occurred. After you have collected the pertinent data, it can be sorted for analysis based on MAC address, creation order, selected statistic, or errors.

When you open the Host Group the Control Table is displayed as shown below. Use this table to select an interface on the monitored device, such as a hub's repeater bus, a port on a switch, or a SLIP interface used for device management. Click on the Add (Edit) button to add (edit) an index entry. The dialog box that opens is described in the following table.



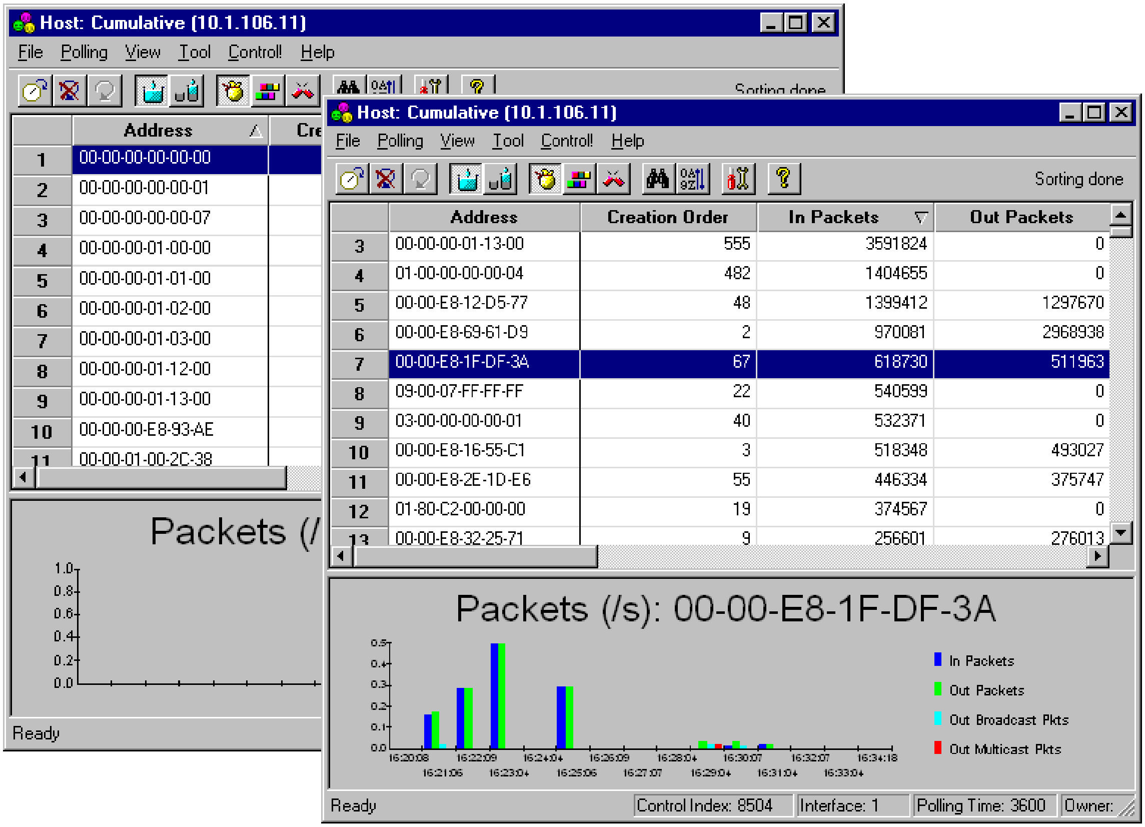
Total: 1

(I)ndex	(O)wner	Interface	Table
8504		1 (ether0)	

Index: 8504
Interface: 1 (ether0)
Owner: steve

Field	Description
Index	A number that identifies the row in the table.
Owner	The person who created this entry.
Interface	A media interface on the monitored device. (MIB-2: 1.3.6.1.2.1.2.2.1.2)
Table Size	The number of host entries added to this table by the RMON probe.
Last Delete Time	The last time data was deleted from this table due to lack of space.
Status	Possible states include “under creation,” “valid,” and “invalid.”
Refresh Time for Control Table	The refresh interval for this control table. Range: 5-600 seconds

To view the data collected for a specific interface, highlight the concerned entry in the control table and press the View button. When you open the host table, the RMON Manager will poll the RMON probe for the most current information. Polling may take a while to complete depending on the number of entries included in the table. Note that the polling status is displayed in the status bar at the bottom of the screen.



	Address	Cre
1	00-00-00-00-00-00	
2	00-00-00-00-00-01	
3	00-00-00-00-00-07	
4	00-00-00-01-00-00	
5	00-00-00-01-01-00	
6	00-00-00-01-02-00	
7	00-00-00-01-03-00	
8	00-00-00-01-12-00	
9	00-00-00-01-13-00	
10	00-00-00-E8-93-AE	
11	00-00-01-00-2C-38	

	Address	Creation Order	In Packets	Out Packets
3	00-00-00-01-13-00	555	3591824	0
4	01-00-00-00-00-04	482	1404655	0
5	00-00-E8-12-D5-77	48	1399412	1297670
6	00-00-E8-69-61-D9	2	970081	2968938
7	00-00-E8-1F-DF-3A	67	618730	511963
8	09-00-07-FF-FF-FF	22	540599	0
9	03-00-00-00-00-01	40	532371	0
10	00-00-E8-16-55-C1	3	518348	493027
11	00-00-E8-2E-1D-E6	55	446334	375747
12	01-80-C2-00-00-00	19	374567	0
13	00-00-E8-32-25-71	9	256601	276013

Control Index: 8504
Interface: 1
Polling Time: 3600
Owner:

By default, the entries are sorted according to address, cumulative values are listed in the numeric table at the top of the screen, and packets/second are displayed in the graph at the bottom of the screen. The configuration and display options are lised below.

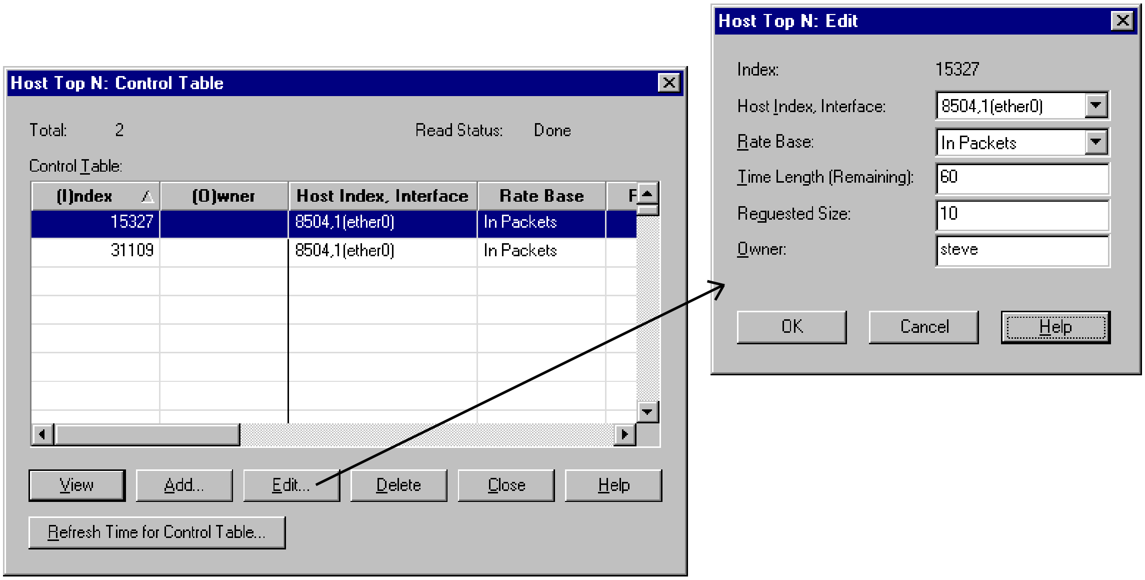
Field	Description
File	Exit
Polling	Polling Time (5 - 3600 seconds), Pause, Resume
View	Table - Cumulative, Delta Graph - Packets, Bytes, Errors
Tool	Find - Locates a specific MAC address in the host table. Sort - Sorts the table based on any of the displayed columns, in ascending or descending order. (You can also sort the table by clicking on any column header.)
Control!	Control Table
Help	On-line help

Note: If an entry displays “Lost Track” when you click on it in the table, the record cannot be found either because polling was stopped before completion, or the entry was deleted due to lack of space.

Host Top N Group

The Host Top N Group can display the hosts with the highest recorded value for a selected statistic. This group can be used to identify the most active hosts on the network in regard to certain statistics. The values are displayed as an overall rate for a specified interval for a specific number of top hosts. The group can be used to quickly identify the most active hosts based on a certain statistic, such as those that are transmitting the most broadcast messages or those that are reporting the largest number of errors.

Use the control table shown below to configure entries for the Host Top N group, including the device interface, the statistic to monitor, the duration to monitor, and the number of top hosts to list. Click on the Add (Edit) button to add (edit) an index entry. The dialog box that opens is described in the following table.



Total: 2

Read Status: Done

Control Table:

(I)ndex	(O)wner	Host Index, Interface	Rate Base	F
15327		8504,1(ether0)	In Packets	
31109		8504,1(ether0)	In Packets	

Index: 15327

Host Index, Interface: 8504,1(ether0)

Rate Base: In Packets

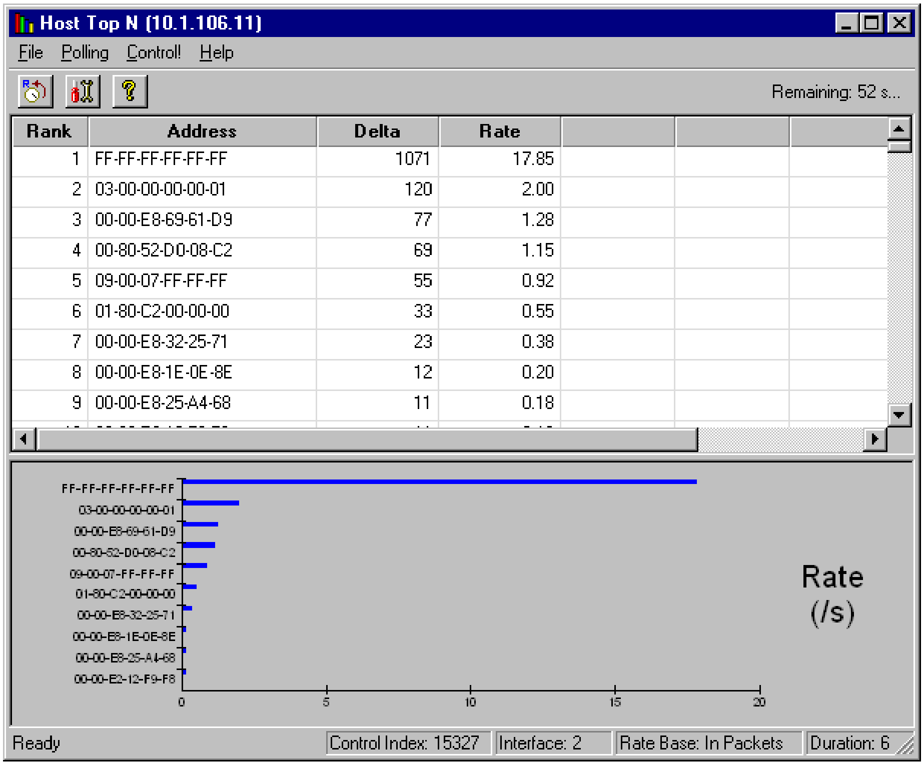
Time Length (Remaining): 60

Requested Size: 10

Owner: steve

Field	Description
Index	A number that identifies the row in the table.
Owner	The person who created this entry.
Host Index, Interface	A media interface on the monitored device. (MIB-2: 1.3.6.1.2.1.2.2.1.2)
Rate Base	The statistic to monitor (In Packets, Out Packets, In Bytes, Out Bytes, Errors, Broadcasts, or Multicasts).
Remaining	The monitoring time remaining (in seconds).
Duration	The length of time to monitor the interface.
Requested Size	The number of top hosts to list.
Granted Size	The number of top hosts the probe is able to list.
Start Time	The value of SysUpTime when this entry was last started.
Status	Possible states include “under creation,” “valid,” and “invalid.”
Refresh Time for Control Table	The refresh interval for this control table. Range: 5-600 seconds

To view the data collected for a specific entry, highlight it in the control table and press the View button. When you open the Host Top N table, the RMON probe start monitoring the interface. The time remaining to monitor the interface is listed at the top of table.



Rank	Address	Delta	Rate
1	FF-FF-FF-FF-FF-FF	1071	17.85
2	03-00-00-00-00-01	120	2.00
3	00-00-E8-69-61-D9	77	1.28
4	00-80-52-D0-08-C2	69	1.15
5	09-00-07-FF-FF-FF	55	0.92
6	01-80-C2-00-00-00	33	0.55
7	00-00-E8-32-25-71	23	0.38
8	00-00-E8-1E-0E-8E	12	0.20
9	00-00-E8-25-A4-68	11	0.18

Ready | Control Index: 15327 | Interface: 2 | Rate Base: In Packets | Duration: 6

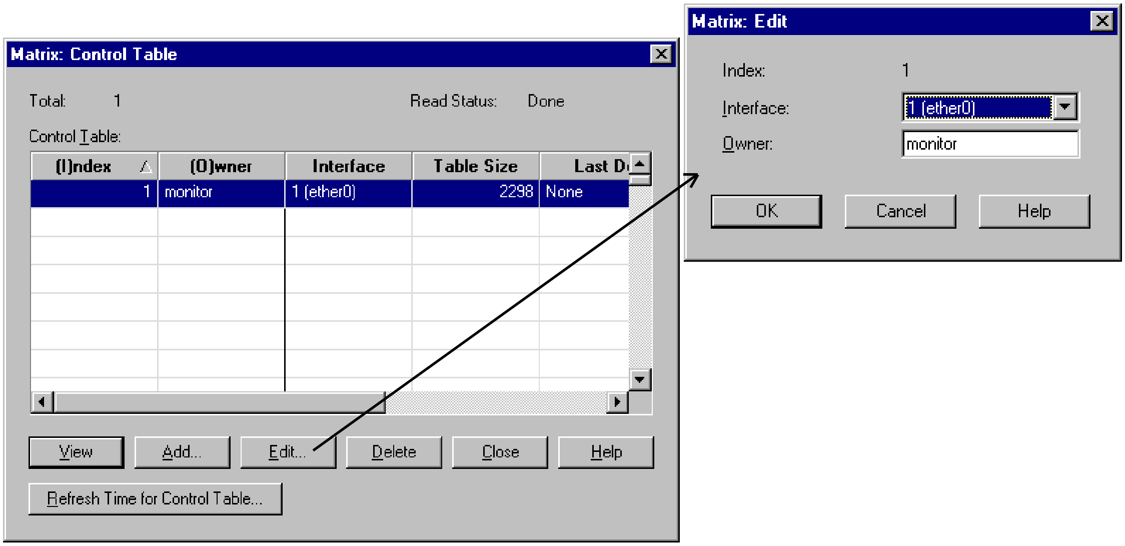
The entries are sorted according to the most active hosts, with the change in value (delta) and rate of change shown by the table at the top of the screen, and the rate shown by the graph at the bottom of the screen. The configuration and display options are listed below.

Field	Description
File	Exit
Polling	Automatically resets polling time after each period is completed.
Control!	Control Table
Help	On-line help

Matrix Group

The Matrix group can maintain statistics on conversations that occur between each pair of hosts on the network. This group can display statistics for traffic transmitted from any source address, traffic received by any destination address, or traffic passing between any host pair. For example, if an alarm is set off for a high watermark on traffic loading, you can use the Host Top N group to identify the hosts with the heaviest load, and then use the Matrix group to analyze the conversations taking place. If a host is transmitting a lot of packets, but not receiving many responses, this may indicate a faulty device. On the other hand, if a host is receiving a lot of traffic but is not responding, either the host is overloaded and cannot keep up with the requests, or the network is overloaded and should be segmented.

Use the control table shown below to configure entries for the Matrix group, including the device interface and owner. Click on the Add (Edit) button to add (edit) an index entry. The dialog box that opens is described in the following table.



Total: 1

Read Status: Done

(I)ndex	(O)wner	Interface	Table Size	Last D
1	monitor	1 (ether0)	2298	None

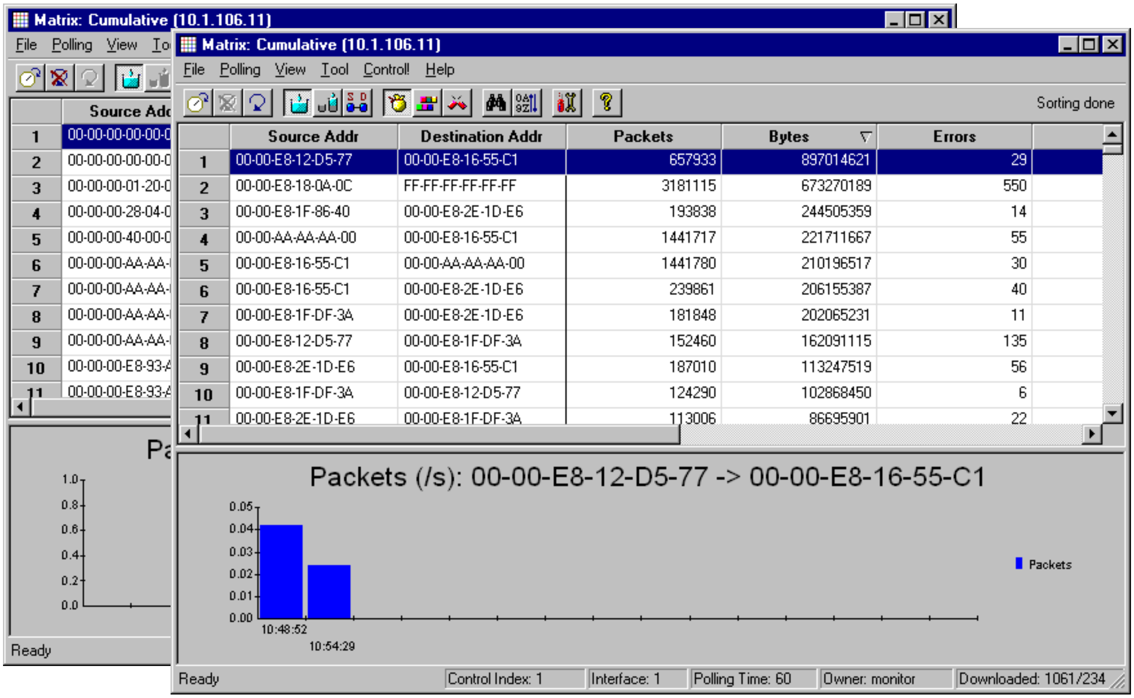
Index: 1

Interface: 1 (ether0)

Owner: monitor

Field	Description
Index	A number that identifies the row in the table.
Owner	The person who created this entry.
Interface	A media interface on the monitored device. (MIB-2: 1.3.6.1.2.1.2.2.1.2)
Table Size	The number of host entries added to this table by the RMON probe.
Last Delete Time	The last time data was deleted from this table due to lack of space.
Status	Possible states include “under creation,” “valid,” and “invalid.”
Refresh Time for Control Table	The refresh interval for this control table. Range: 5-600 seconds

To view the matrix data collected for a specific interface, highlight it in the control table and press the View button. When you open the Matrix table, the RMON probe starts monitoring the interface. The number of entries downloaded is listed in the status bar at the bottom of the table.


Matrix: Cumulative (10.1.106.11)
File Polling View Tool Control! Help
Sorting done
Packets (/s): 00-00-E8-12-D5-77 -> 00-00-E8-16-55-C1
0.05
0.04
0.03
0.02
0.01
0.00
10:48:52
10:54:29
Packets
Ready

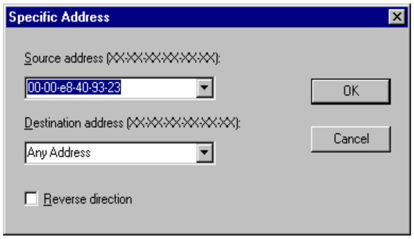
	Source Addr	Destination Addr	Packets	Bytes	Errors
1	00-00-E8-12-D5-77	00-00-E8-16-55-C1	657933	897014621	29
2	00-00-E8-18-0A-0C	FF-FF-FF-FF-FF-FF	3181115	673270189	550
3	00-00-E8-1F-86-40	00-00-E8-2E-1D-E6	193838	244505359	14
4	00-00-AA-AA-AA-00	00-00-E8-16-55-C1	1441717	221711667	55
5	00-00-E8-16-55-C1	00-00-AA-AA-AA-00	1441780	210196517	30
6	00-00-E8-16-55-C1	00-00-E8-2E-1D-E6	239861	206155387	40
7	00-00-E8-1F-DF-3A	00-00-E8-2E-1D-E6	181848	202065231	11
8	00-00-E8-12-D5-77	00-00-E8-1F-DF-3A	152460	162091115	135
9	00-00-E8-2E-1D-E6	00-00-E8-16-55-C1	187010	113247519	56
10	00-00-E8-1F-DF-3A	00-00-E8-12-D5-77	124290	102868450	6
11	00-00-E8-2E-1D-E6	00-00-E8-1F-DF-3A	113006	86695901	22

Control Index: 1 | Interface: 1 | Polling Time: 60 | Owner: monitor | Downloaded: 1061/234

By default, the entries are sorted according to the source address, cumulative values are listed in the numeric table at the top of the screen, and packets/second are displayed in the graph at the bottom of the screen. Since the graph displays rate, nothing will be displayed for the highlighted entry if no activity was seen during the last polling interval. The configuration and display options are listed below.

Field	Description
File	Exit
Polling	Polling Time (5 - 3600 seconds), Pause, Resume
View	Table - Cumulative, Delta, Specific Source/Destination Graph - Packets, Bytes, Errors
Tool	Find - Locates a specific MAC address in the matrix table. Sort - Sorts the table based on any of the displayed columns, in ascending or descending order. (You can also sort the table by clicking on any column header.)
Control!	Control Table
Help	On-line help

Note: When specifying the view for a specific source-destination pair, you can specify both the source and destination, just the source or destination (using Any Address as shown in this example),
or all the transmitted and received traffic for a specific address pair (using Reverse direction).



Specific Address

Source address (XX-XX-XX-XX-XX-XX): 00-00-e8-40-93-23

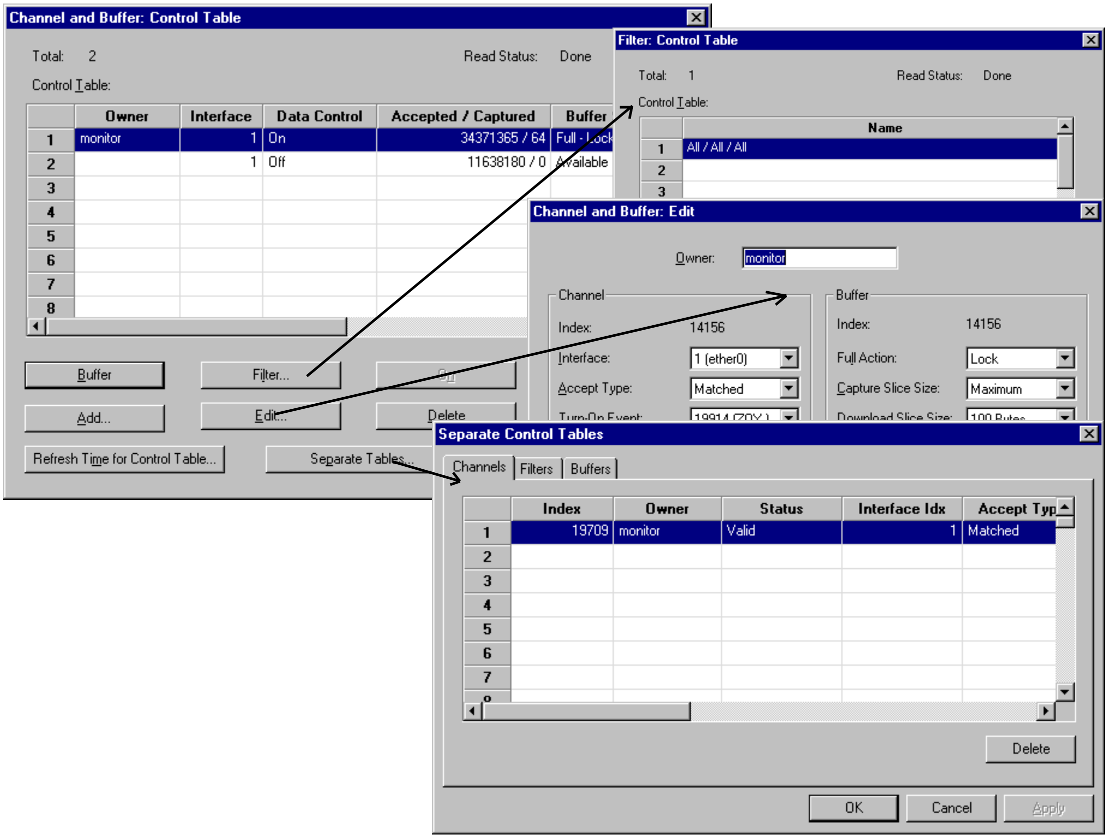
Destination address (XX-XX-XX-XX-XX-XX): Any Address

☐ Reverse direction

Filter and Capture Groups

The Filter Group is used to generate a packet stream from the frames that match a specified pattern, while the Capture Group is used to manage the storage buffers for packets captured by the Filter Group. These groups can be used to capture network traffic associated with precisely defined events. The captured data or trigger events can then be used to debug application problems or fine-tune network performance. From captured data, you can view the associated network protocol, summary information for each packet, or a detailed hexadecimal and ASCII breakdown of all traffic (page 5-24).

Use the control table to configure and activate channels as described below.



Channel and Buffer: Control Table

Total: 2 Read Status: Done

Control Table:

	Owner	Interface	Data Control	Accepted / Captured	Buffer
1	monitor	1	On	34371365 / 64	Full - Lock
2		1	Off	11638180 / 0	Available
3					
4					
5					
6					
7					
8					

Filter: Control Table

Total: 1 Read Status: Done

Control Table:

	Name
1	All / All / All
2	
3	

Channel and Buffer: Edit

Owner: monitor

Channel

Index: 14156

Interface: 1 (ether0)

Accept Type: Matched

Turn-On Event:

Buffer

Index: 14156

Full Action: Lock

Capture Slice Size: Maximum

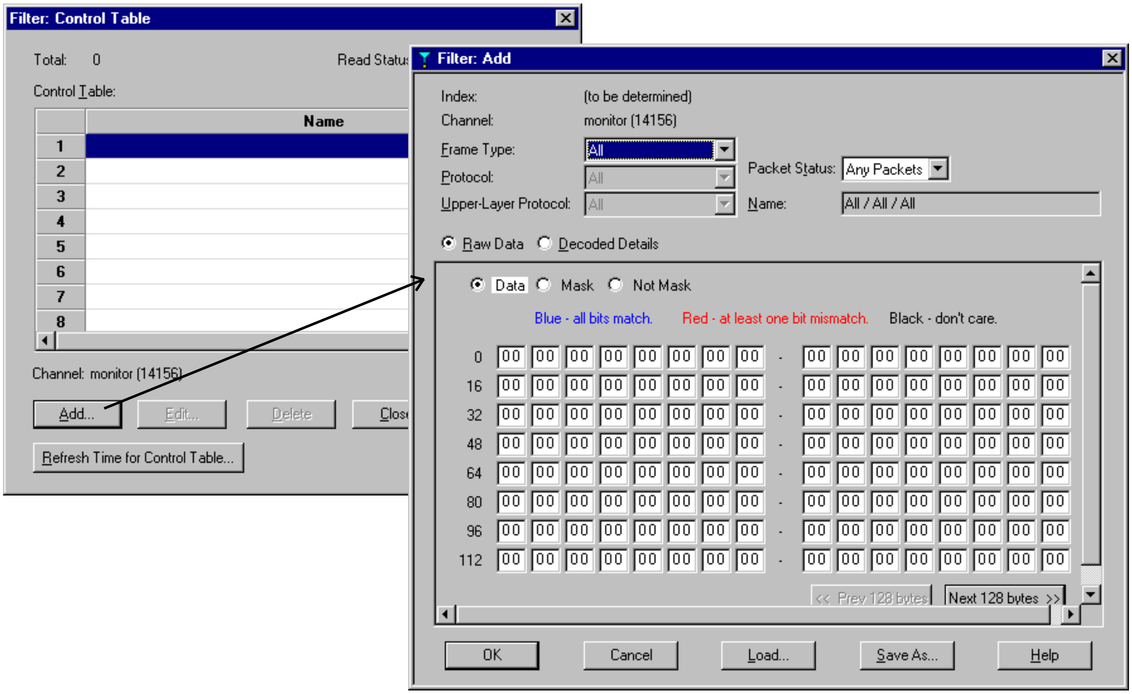
Download Slice Size:

Separate Control Tables

	Index	Owner	Status	Interface Idx	Accept Typ
1	19709	monitor	Valid	1	Matched
2					
3					
4					
5					
6					
7					

Field	Description
Owner	The person who created this entry.
Interface	A media interface on the monitored device. (MIB-2: 1.3.6.1.2.1.2.2.1.2)
Data Control	Indicates if the capture channel is enabled or not. The capture channel may be manually enabled or disabled using the On/Off button. Or a Turn-On Event defined for this channel can be used to enable it. Once enabled, the channel will start capturing packets that pass the filter.
Accepted/Captured	Indicates the number of times this channel has accepted a packet, and the number of packets currently in this capture buffer.
Buffer Status	Indicates whether the buffer space is available or full (wrapped or locked).
Channel Index	A number that identifies the channel in the channel table.
Buffer Control Index	A number that identifies this buffer in the bufferControl table.
Description	A comment provided by the user describing this channel.
Buffer	Displays the buffer for the selected control entry.
Filter	Displays the filter control table.
On/Off	This button is used to manually enable or disable the capture channel.
Refresh Time for Control Table	The refresh interval for this control table. Range: 5-600 seconds
Separate Tables	Displays a tabbed window with the channel, filter and buffer tables.

Configuring Filters - You can set up to 20 filters for each channel. Just highlight the concerned channel and then press the Filter button. If you define more than one filter, they will be OR'ed together and compared against data crossing the specified interface. To configure filters, click on the Add (Edit) button to add (edit) an entry. The dialog box that opens is described in the following table.



Filter: Control Table

Total: 0

Read Statu

Control Table:

	Name
1	
2	
3	
4	
5	
6	
7	
8	

Channel: monitor (14156)

Filter: Add

Index: (to be determined)

Channel: monitor (14156)

Frame Type: All

Protocol: All

Upper-Layer Protocol: All

Packet Status: Any Packets

Name: All / All / All

● Raw Data ○ Decoded Details

● Data ○ Mask ○ Not Mask

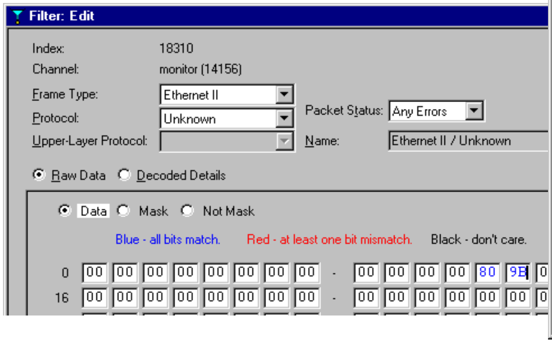
Blue - all bits match. Red - at least one bit mismatch. Black - don't care.

```
  0  00 00 00 00 00 00 00 00 . 00 00 00 00 00 00 00 00
 16  00 00 00 00 00 00 00 00 . 00 00 00 00 00 00 00 00
 32  00 00 00 00 00 00 00 00 . 00 00 00 00 00 00 00 00
 48  00 00 00 00 00 00 00 00 . 00 00 00 00 00 00 00 00
 64  00 00 00 00 00 00 00 00 . 00 00 00 00 00 00 00 00
 80  00 00 00 00 00 00 00 00 . 00 00 00 00 00 00 00 00
 96  00 00 00 00 00 00 00 00 . 00 00 00 00 00 00 00 00
112  00 00 00 00 00 00 00 00 . 00 00 00 00 00 00 00 00
```

You can filter raw data for any frame type. Or you can filter detailed information from the header fields if you indicate frame type. When filtering Ethernet II, Ethernet 802.2 or Ethernet SNAP, you can specify the network protocol (IP, IPX, Unknown), as well as the transport protocol (IP: All, TCP, UDP, Unknown; or IPX: All, RIP, SAP, Unknown). Also regardless of the frame type, you can choose to filter all packets or just certain error packets (such as fragments or jabber). Detailed configuration options for filters are shown in the following table.

Field	Description
Index	An index that uniquely identifies an entry in the filter table.
Channel	The owner who created the corresponding channel entry, and the channel index.
Frame Type	The frame type to filter. Values: All, Ethernet II, Ethernet 802.2, Ethernet SNAP, Ethernet 802.3 Raw, Unknown
Protocol	The network protocol to filter. This entry is only available if Ethernet II, Ethernet 802.2, or Ethernet SNAP is the selected frame type. Values: IP, IPX, Unknown
Upper-Layer Protocol	The transport protocol to filter. The options available depend on the selected network protocol. IP: All, TCP, UDP, Unknown IPX: All, RIP, SAP, Unknown
Packet Status	The status of packets to filter: Any Packets, No Errors, Any Errors, CRC/Alignment, Packets < 64 Bytes, Undersize (packets < 64 bytes, with no CRC/alignment errors), Packets > 1518 Bytes, Oversize (packets >1518 bytes, with no CRC/alignment errors), Fragments, Jabber, Other
Name	A name for this filter expression, consisting of the selected – Frame Type / Protocol / Upper-Layer Protocol.
Raw Data	Allows you to enter a specific data pattern to filter. Select Data and input the filter pattern, select Mask to indicate the relevant bits, then select Not Mask to indicate the bits that should match or not match. You can check for a pattern anywhere within the first 256 bytes of a frame. Data: The bit pattern to filter. Mask: The relevant bits within the data. Not Mask: Those bits in the mask to match (0) or not match (1).
Decoded Details	If the frame and protocol type are specified, then you can filter specific fields directly from the frame and protocol headers.
Load/Save As	Loads a predefined filter into the filter table, or saves the current filter as a file.

The following example shows a filter expression designed to capture any errors occurring in AppleTalk packets. While the other example shows filtering for decoded details that is designed to capture any Ethernet II packets directed to the IP address 10.1.0.23 that have the Don't Fragment flag set.


Filter: Edit

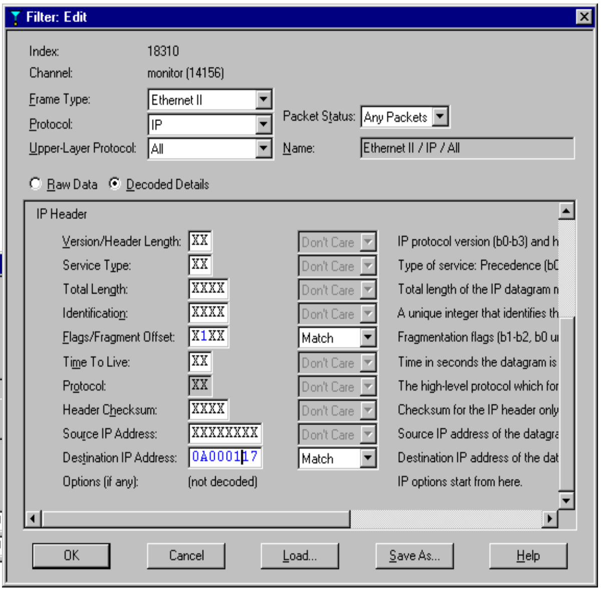
Index: 18310
Channel: monitor (14156)
Frame Type: Ethernet II
Protocol: Unknown
Upper-Layer Protocol:
Packet Status: Any Errors
Name: Ethernet II / Unknown

● Raw Data ○ Decoded Details

● Data ○ Mask ○ Not Mask

Blue - all bits match. Red - at least one bit mismatch. Black - don't care.

```
 0 00 00 00 00 00 00 00 00 - 00 00 00 00 80 9B 0
16 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 0
```


Filter: Edit

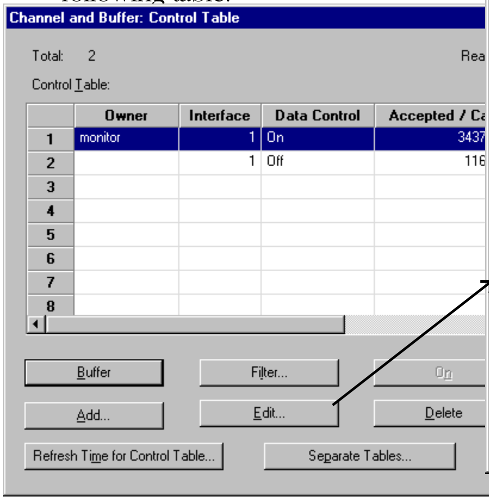
Index: 18310
Channel: monitor (14156)
Frame Type: Ethernet II
Protocol: IP
Upper-Layer Protocol: All
Packet Status: Any Packets
Name: Ethernet II / IP / All

○ Raw Data ● Decoded Details

IP Header

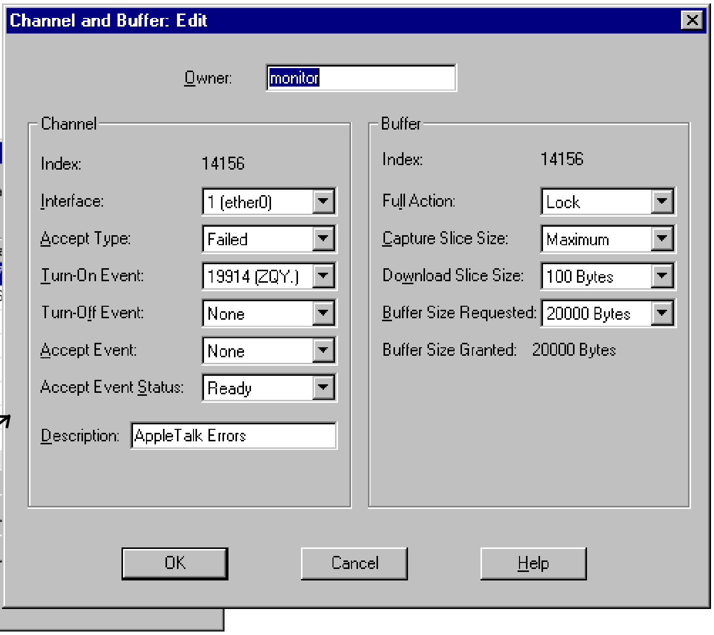
Version/Header Length:	XX	Don't Care	IP protocol version (b0-b3) and h
Service Type:	XX	Don't Care	Type of service: Precedence (b0
Total Length:	XXXX	Don't Care	Total length of the IP datagram n
Identification:	XXXX	Don't Care	A unique integer that identifies th
Flags/Fragment Offset:	X1XX	Match	Fragmentation flags (b1-b2, b0 u
Time To Live:	XX	Don't Care	Time in seconds the datagram is
Protocol:	XX	Don't Care	The high-level protocol which for
Header Checksum:	XXXX	Don't Care	Checksum for the IP header only
Source IP Address:	XXXXXXXX	Don't Care	Source IP address of the datagra
Destination IP Address:	0A000117	Match	Destination IP address of the dat
Options (if any):	(not decoded)		IP options start from here.

Configuring Channels - The data and event stream formed by the packets that match the filter (or a group of combined filters) is referred to as a "channel." A channel can be based on a single filter or on multiple filter expressions which are OR'ed together. The channel can be configured to pass packets through if they match or fail to match the stated expression(s). Events can then be defined to turn the channel on or off, or can also be triggered when the packets are accepted. From the control table, click on the Add (Edit) button to add (edit) a channel entry. The dialog box that opens is described in the following table.


Channel and Buffer: Control Table

Total: 2
Rea
Control Table:

	Owner	Interface	Data Control	Accepted / Ca
1	monitor	1	On	3437
2		1	Off	116
3				
4				
5				
6				
7				
8				


Channel and Buffer: Edit

Owner: monitor

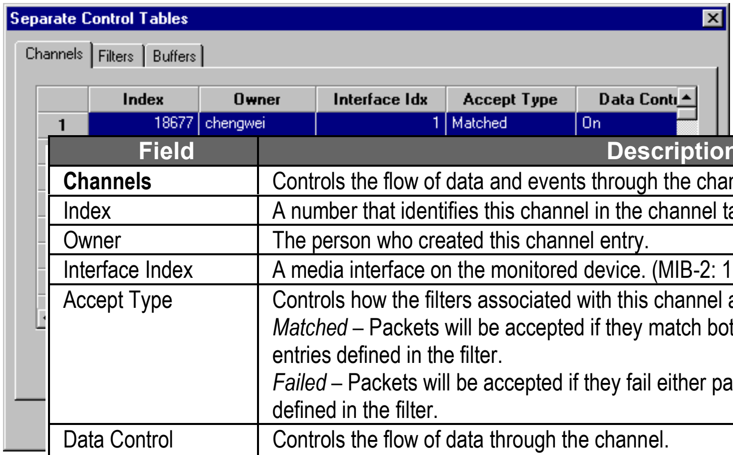
Channel
Index: 14156
Interface: 1 (ether0)
Accept Type: Failed
Turn-On Event: 19914 (ZQY.)
Turn-Off Event: None
Accept Event: None
Accept Event Status: Ready
Description: AppleTalk Errors

Buffer
Index: 14156
Full Action: Lock
Capture Slice Size: Maximum
Download Slice Size: 100 Bytes
Buffer Size Requested: 20000 Bytes
Buffer Size Granted: 20000 Bytes

Field	Description
Channel	Controls the flow of data through the channel.
Index	A number that identifies this channel in the channel table.
Interface	A media interface on the monitored device. (MIB-2: 1.3.6.1.2.1.2.2.1.2)
Accept Type	Controls how the filters associated with this channel are implemented. *Matched* - Packets will be accepted if they match both packet data and packet status entries defined in the filter. *Failed* - Packets will be accepted if they fail either packet data or packet status entries defined in the filter.
Turn-On Event	Specifies an event that will turn on the channel.
Turn-Off Event	Specifies an event that will turn off the channel.
Accept Event	Specifies an event that will be generated when the channel is on and a packet is accepted.
Accept Event Status	Controls the flow of events. *Ready* - A single event will be generated, after which the status will be set by the RMON probe to “Fired.” While in the Fired state, no events will be generated until the status is reset to Ready or Always Ready. *Always Ready* – Disables flow control and allows events to generated at will. Using this setting runs the risk of generating a high volume of traffic and affecting network performance.
Description	A comment provided by the user describing this channel.

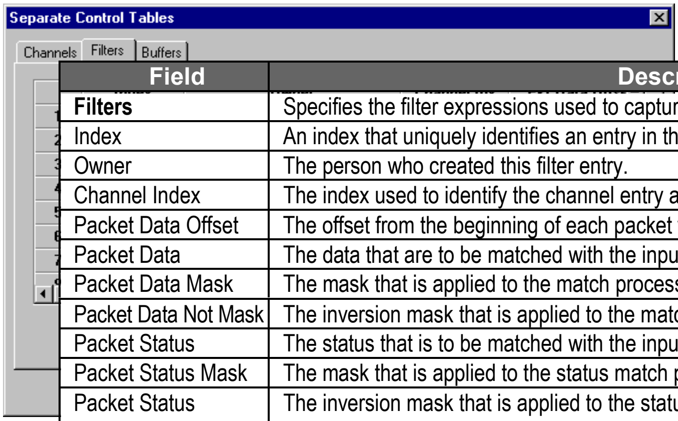
Buffer	Configures the buffer used to store packets matched for this control entry.
Index	A number that identifies this buffer in the buffer table.
Full Action	Controls the action of the buffer when it reaches full status. *Lock* - The buffer will be locked as soon as it fills. *Wrap* - Old data will be overwritten when the buffer fills.
Capture Slice Size	The maximum number of bytes for each packet that will be saved in this capture buffer. Values include 100, 200, 500, 1000 bytes and Maximum. If set to Maximum, the capture buffer will save as many bytes as possible.
Download Slice Size	The maximum number of bytes for each packet that will be returned to the management station in a single retrieve operation. Values include 100, 200 and 500 bytes.
Buffer Size Requested	The number of bytes requested for this capture buffer. Values include 10000, 20000, 50000, 100000, 200000, 500000 bytes and Maximum. If set to Maximum, the capture buffer will save as many bytes as possible.
Buffer Size Granted	The number of bytes granted for this capture buffer.

Viewing Separate Tables - You can quickly display the configuration for channels, filters and buffers using the tabbed window shown below. The information provided is described in the following tables.



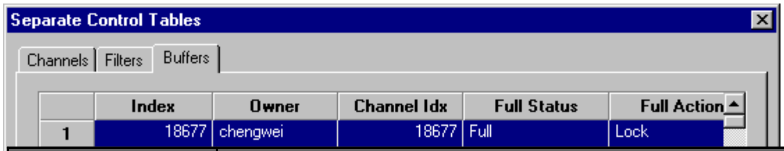
	Index	Owner	Interface Idx	Accept Type	Data Conti
1	18677	chengwei	1	Matched	On

Field	Description
Channels	Controls the flow of data and events through the channel.
Index	A number that identifies this channel in the channel table.
Owner	The person who created this channel entry.
Interface Index	A media interface on the monitored device. (MIB-2: 1.3.6.1.2.1.2.2.1.2)
Accept Type	Controls how the filters associated with this channel are implemented. *Matched* – Packets will be accepted if they match both packet data and packet status entries defined in the filter. *Failed* – Packets will be accepted if they fail either packet data or packet status entries defined in the filter.
Data Control	Controls the flow of data through the channel. *On* - Data, Status and Events flow through the channel. *Off* - Nothing flows through the channel.
Turn-On Event	Specifies an event that will turn on the channel.
Turn-Off Event	Specifies an event that will turn off the channel.
Event Index	Specifies an event to be generated when the channel is on and a packet is accepted.
Event Status	Controls the flow of events. *Ready* - A single event will be generated, after which the status will be set by the RMON probe to “Fired.” While in the Fired state, no events will be generated until the status is reset to Ready or Always Ready. *Always Ready* – Disables flow control and allows events to generated at will. Using this setting runs the risk of generating a high volume of traffic and affecting network performance.
Matches	The number of times this channel has accepted a packet.
Description	A comment provided by the user describing this channel.
Status	The current status of the index entry in the Control Table: Valid, Under Creation or Invalid



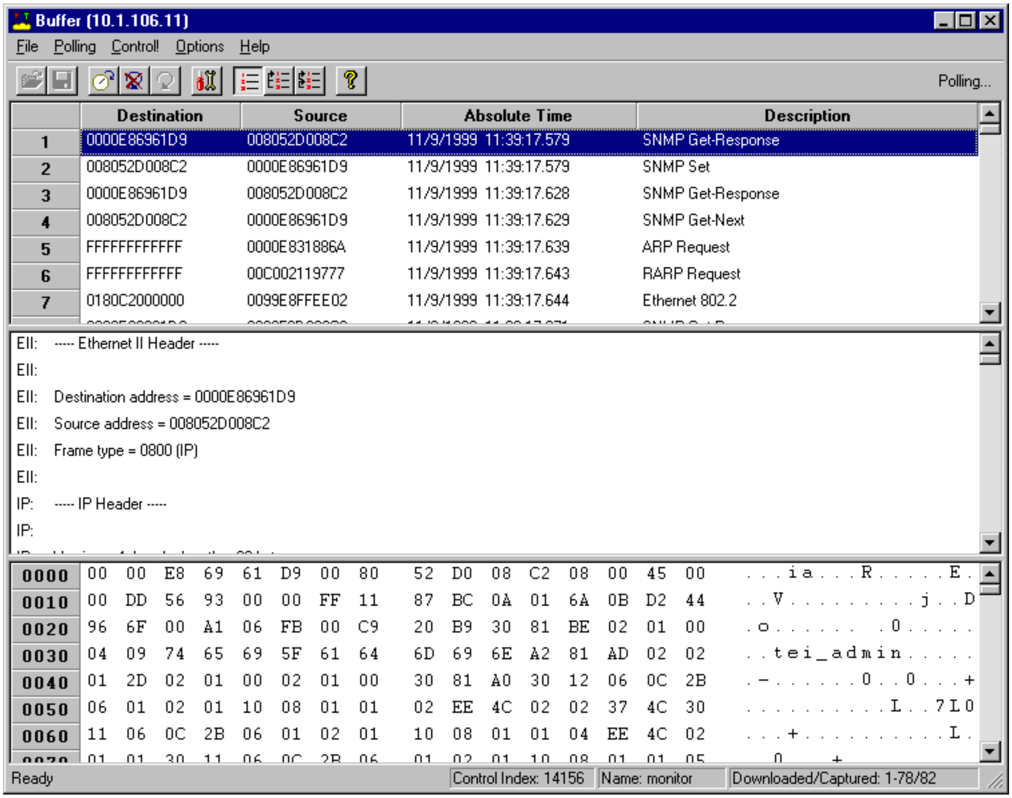
Field	Description
Filters	Specifies the filter expressions used to capture data from the interface.
Index	An index that uniquely identifies an entry in the filter table.
Owner	The person who created this filter entry.
Channel Index	The index used to identify the channel entry associated with this filter.
Packet Data Offset	The offset from the beginning of each packet where a match of packet data will be attempted.
Packet Data	The data that are to be matched with the input packet.
Packet Data Mask	The mask that is applied to the match process.
Packet Data Not Mask	The inversion mask that is applied to the match process.
Packet Status	The status that is to be matched with the input packet.
Packet Status Mask	The mask that is applied to the status match process.
Packet Status Not Mask	The inversion mask that is applied to the status match process.
Status	The current status of the index entry in the Control Table: Valid, Under Creation or Invalid

Displaying Buffer Contents - The capture buffer displays information on packets in three different windows. The upper window shows summary information on each packet, including source, destination, timestamp, and protocol type. The middle window decodes information for each layer in the protocol stack. The bottom window provides raw data in hexadecimal and ASCII format. The configuration options for the buffer display are described in the following table.



	Index	Owner	Channel Idx	Full Status	Full Action
1	18677	chengwei	18677	Full	Lock

Field	Description
Buffers	The buffer used to store packets matched for this control entry.
Index	A number that identifies this buffer in the buffer table.
Owner	The person who created this buffer entry.
Channel Index	The index for the channel entry associated with this buffer.
Full Status	Indicates whether space is still available in the buffer, or if it is full.
Full Action	Controls the action of the buffer when it reaches full status. *Lock* - The buffer will be locked as soon as it fills. *Wrap* - Old data will be overwritten when the buffer fills.
Capture Slice Size	The maximum number of bytes for each packet that will be saved in this capture buffer.
Download Slice Size	The maximum number of bytes for each packet that will be returned to the management station in a single retrieve operation.
Bytes Requested	The number of bytes requested for this capture buffer.
Bytes Granted	The number of bytes granted for this capture buffer.
Captured Packets	The number of packets currently in this capture buffer.
Turn On Time	The value of sysUpTime when this capture buffer was first tuned on.
Status	The current status of the index entry in the Control Table: Valid, Under Creation or Invalid



	Destination	Source	Absolute Time	Description
1	0000E86961D9	008052D008C2	11/9/1999 11:39:17.579	SNMP Get-Response
2	008052D008C2	0000E86961D9	11/9/1999 11:39:17.579	SNMP Set
3	0000E86961D9	008052D008C2	11/9/1999 11:39:17.628	SNMP Get-Response
4	008052D008C2	0000E86961D9	11/9/1999 11:39:17.629	SNMP Get-Next
5	FFFFFFFFFFFF	0000E831886A	11/9/1999 11:39:17.639	ARP Request
6	FFFFFFFFFFFF	00C002119777	11/9/1999 11:39:17.643	RARP Request
7	0180C2000000	0099E8FFEE02	11/9/1999 11:39:17.644	Ethernet 802.2

```
EII:   ----- Ethernet II Header -----
EII:
EII:   Destination address = 0000E86961D9
EII:   Source address = 008052D008C2
EII:   Frame type = 0800 (IP)
EII:
IP:    ----- IP Header -----
IP:
```

```
0000  00 00 E8 69 61 D9 00 80  52 D0 08 C2 08 00 45 00   . . . i a . . . R . . . . . E .
0010  00 DD 56 93 00 00 FF 11  87 BC 0A 01 6A 0B D2 44   . . V . . . . . . . . . j . . D
0020  96 6F 00 A1 06 FB 00 C9  20 B9 30 81 BE 02 01 00   . o . . . . . .   . 0 . . . . .
0030  04 09 74 65 69 5F 61 64  6D 69 6E A2 81 AD 02 02   . . t e i _ a d m i n . . . . .
0040  01 2D 02 01 00 02 01 00  30 81 A0 30 12 06 0C 2B   . - . . . . . . 0 . . 0 . . . +
0050  06 01 02 01 10 08 01 01  02 EE 4C 02 02 37 4C 30   . . . . . . . . . . L . . 7 L 0
0060  11 06 0C 2B 06 01 02 01  10 08 01 01 04 EE 4C 02   . . . + . . . . . . . . . . L .
```

Ready | Control Index: 14156 | Name: monitor | Downloaded/Captured: 1-78/82

Notes: 1. To display decoded and raw data for a summary item, highlight it with your cursor.

2. To display raw data corresponding to a decoded entry, highlight it with your cursor.

Field	Description
File	Load, Save As, Exit Note that you must pause polling before you can load or save a buffer file.
Polling	Polling Time (5 - 3600 seconds), Pause, Resume Also includes options to download Next 2000 Packets or Previous 2000 Packets.
Control!	Control Table
Options	Time Format includes: Absolute - Date and local time. Relative - Time relative to when the first packet was added to the buffer. Difference - Time difference between capturing last packet and current packet.
Help	On-line help

APPENDIX A

TYPICAL ELITEVIEW APPLICATIONS

EliteView is a versatile network management software package that supports an open platform architecture. Users with enough technical knowhow can customize it according to their needs in order to monitor and control SMC or other third-party SNMP devices. This chapter describes a few useful applications you can adopt to your own network environment.

Adding a New MIB Using the MIB Compiler

Let's say that you want to add and compile a new MIB into the EliteView MIB database. The new MIB is from a third-party SNMP device.

To add a new MIB into the EliteView MIB database,

1. Get the MIB text file of the third-party SNMP device. Let's assume that the filename of the MIB text file is DEVICE.MIB.
2. Copy the third-party MIB into the EliteView directory. Type:

```
Copy d:DEVICE.MIB c:\ELITEVIEW
```

3. Type the full name of the MIB file in the *Filename* field.
4. Select *Load* from the MIB Compiler.

DEVICE.MIB is now included in the EliteView MIB database.

Tip: Now that the third-party MIB has been added to the EliteView MIB database you can do any of the following things:

1. Use Discovery to search for the third-party device.
2. Drag-and-drop the third-party device icon onto any EliteView submap. Third-party devices are labeled as an SNMP node on the EliteView map.
3. Double-click on the SNMP node to bring up the MIB Browser. A full knowledge of your third-party MIB will allow you to input the necessary parameter in each box prompt. This will allow you to view the status of any MIB variable you wish to monitor.
4. Use Log Manager to gather data from the SNMP device.
5. Use Log and Event Manager to monitor the SNMP device.
6. Customize EliteView to receive third-party traps.
7. Import Logged data into other software for further processing.

Managing a Third-Party Device Using the MIB Browser

Before you can manage any third-party device, you should have a clear understanding of definitions for the variables in this MIB. Please consult your vendor for details.

Assumption: With the absence of a sample third-party MIB and device, and for purposes of this example, let us assume that SMC's EZ Stack-10 system is a third-party SNMP device.

Let's say that you want to monitor the total number of packets the third-party SNMP device receives.

To manage a third-party SNMP device (i.e., monitor the number of packets received):

1. Compile the third-party MIB into the EliteView MIB database. Refer to the section on "Adding a New MIB Using the MIB Compiler" for more information. If you have completed this operation, skip to the next step.
2. Create a map that includes the third-party SNMP device (i.e., a device labeled as "SNMP node") that you wish to monitor. You can use Discovery to find it; then simply drag and drop the corresponding SNMP node onto any submap.
3. Determine the MIB variable that represents the total number of packets received by the system. Say `ezRptrTotalFrames` is the MIB variable that represents this count.
4. Double-click on the SNMP node (i.e., third-party device) to bring up the MIB Browser.

Tip: You can also bring up the MIB Browser simply by clicking on its corresponding icon from the EliteView program group.

5. In the "*Node Label*" field, type "*ezRptrTotalFrames,"* and press <*Enter*>.

Tip: In the Node Label field, you will find the "iso" entry. This is the root MIB object. Double-clicking on this entry allows you to move through the MIB tree. Each time you double-click an entry, the list box will display the corresponding "branch" database.

6. An "*Input Index*" dialog box will appear on screen, asking you input the index of the variable you have specified (i.e., `ezRptrTotalFrames`). For example, in this field type the target *SNMP device's IP address*, and then press <*Enter*>.
7. Click the *Get* button and view the result in the Status window.

Let's say that you want to set a variable called hubNMS in a third-party device.

To set a variable in a third party device:

1. Compile the third-party MIB into the EliteView MIB database. Refer to the section on "Adding a New MIB Using the MIB Compiler" for more information. If you have completed this operation, then skip to the next step.
2. Create a map that includes the third-party SNMP device (i.e., a device labeled as "SNMP node") you wish to monitor. You can use Discovery to find it; then simply drag and drop the corresponding SNMP node onto any submap.
3. Double-click on the SNMP node (third-party device) to bring up the MIB Browser.
4. Locate the required MIB variable, "`hubNMS`."
5. Click the *Set Request* button.
6. An "Input Value" dialog box will appear on-screen, prompting you for the value you wish to set. Type in the IP address of your EliteView network management station. Press <*Enter*> or click <*OK*> to confirm your choice. Or click <*Cancel*> to abort this operation.
7. View the result in the Status window.

Using the Log and Event Managers to Monitor the Network

The Log Manager and the Event Manager are two powerful network monitoring modules. Regardless of whether you are managing SMC or other third-party manageable systems, these two modules work in conjuction with each other to allow flexible network management.

To use the Log Manager and Event Manager:

1. Fill in the table on the next page. It contains information you should know in order to use the Log Manager and Event Manager effectively.
2. Bring up the Event Manager and add an event according to the Event Name and actions you have specified. Refer to Chapter 8, "Managing Events" for more information.
3. Bring up the Log Manager and add a log request. Refer to Chapter 7, "Collecting Data with the Log Manager" for more information on using the Log Manager and for setting filters and threshold formulas.

The Log Manager will continuously monitor the target SNMP device after you have set the log request.

Tip: Delete or pause unnecessary log requests to prevent wasting system resources (i.e., disk space, CPU time, and network traffic). You can use the Log Manager to back up or delete logged data frequently and save precious disk space.

Definition of Key Log & Event Paramters

Parameter	Description	Example
Target Address	IP address of target SNMP device you wish to monitor.	192.72.24.05
MIB Module	A collection of managed objects.	SMC-MIB
Variable	Name of variable as defined in the MIB.	sysDescr
Index	Value of index as defined in MIB which is used to access a table.	1
Filter	Formula used to filter information.	(VALUE>100) AND (TIME<120000)
Threshold	Criteria used to generate events when the filter conditions are satisfied.	R>100
Event Name	Name of the event used by the Event Manager to handle the threshold condition	CRITICAL
Event Action	***Beep*** - an audible alarm issued by the network management station	
	Show message box - display a user-defined message in a box on the network management station's screen. The message box appears on top of all other windows. When the message box appears, you must click on the <OK> button to dismiss the message. **This action is recommended for critical events.**	CRITICAL: Hub overheating
	Run program - execute any Windows-based application. **This action is ideal for sending a message via email or FAX.**	PAGER 408-555-4742
	Send message to REPORT - Insert a message into the Report window. Alarm messages are time stamped and shown in chronological order. **This action is recommended for routine events.**	WARNING: SERVER DISK > 90% full
	Write into database - Put a message into the event database	CAUTION: Excessive CRC errors on device

For more detailed information on Log and Event parameters refer to chapters 7 and 8 respectively.

Let's say that you want to monitor the total packets received by an SNMP device whose IP address is 192.72.24.1. Then you would like to save this value every 60 seconds between 9:00 AM and 5:00 PM. And if the rate is greater than 1000 per second, you want to save the value and at the same time be informed of the situation through an audible alarm, such as a beep. Suppose you consider this as a critical condition and would like to call it "Hot 1."

By checking the target SNMP device's MIB, you find that the variable `ezRptrTotalFrames` contains the value that you need. Further, this variable belongs to the "`EZ Stack 10`" module. (The term MIB module is synonymous to the term "MIB Name" in the MIB variable information window of the MIB Browser.) Additional information you already know includes - index is equal to the IP address of the target SNMP device and the community is equal to public.

To monitor the total packets received by the target device:

1. Fill in the following table with all available information.

Parameter	Given Info
Target Address	192.72.24.1
Community	public
MIB Module	SMC3116-MIB
Variable	ifInOctets
Index	192.72.24.1
Polling Interval	60
Filter	(TIME>080000) AND (TIME<170000)
Threshold	S>1000000
Event Name	Hot_1
Event Actions	Beep, Write into database -- Value is $$

2. Bring up the *Event Manager*. It is automatically loaded when you run the main EliteView program. If it is not visible, bring it up by selecting *Event Manager* from the *Utilities Menu*. Add the new Event Name and Event Actions you have specified in the preceding table.
3. Select *Log Manager* from the *Utilities Menu*. Add new log processes according to the information specified in the preceding table into the Log Information dialog box.
4. Collect all the information you need to establish a threshold appropriate for your system. Then use the Log Data Manager to save or delete logged data into your hard disk.

Let's say that you want to monitor the collisions of an SNMP device whose IP address is 192.72.24.2. Then you would like to check the value for every 60 seconds, but do not want to save the data at all. And if the collision rate is greater than 100 per second, you need an audible alarm and a message box to warn you of the condition. You want to name this critical condition as "Hot_2."

By checking the target SNMP device's MIB, you find that the variable `hubTxCollisions` contains the value that you need. Further, this variable belongs to the "`EZ Stack 10`" module. Additional information you already know includes - index is equal to the IP address of the target SNMP device and the community is equal to abc.

To monitor the total packets received by the target device:

1. Fill in the following table with all available information.

Parameter	Given Info
Target Address	192.72.24.2
Community	abc
MIB Module	SMC3116-MIB
Variable	ifInUnknownProtos
Index	192.72.24.2
Polling Interval	60
Filter	TIME<0 (This is an impossible filter because you do not want to save any data.)
Threshold	S>10
Event Name	Security_2
Event Actions	Beep Show message box --- Security_2

2. Bring up the *Event Manager*. Add the new Event Name and Event Actions that you have specified in the preceding table.
3. Bring up the *Log Manager*. Add new log processes according to the information specified in the preceding table by filling in the Log Information dialog box.
4. Collect all the information needed. Then use the Log Data Manager to save or delete logged data on your hard disk.

Customizing EliteView to Receive Third-Party Traps

EliteView flexibly manages trap messages through the Trap Manager, which reads necessary information from TRAP.INI and “listens” to the network for traps.

Once traps are received, the Trap Manager takes the proper action in accordance with instructions in the TRAP.INI file.

Appendix B, “Customizing EliteView” describes the format of the TRAP.INI file. For an advanced user, this format is not difficult to understand. One important thing that you have to remember about is the arrangement of the events and messages for each trap. You should use the Event Manager to add all the events and responses you want. Additionally, the trap message specified in the TRAP.INI file is the “value” of the event that will be substituted for the $$ symbol.

To configure the Trap Manager to receive traps from any third-party SNMP device, you should find answers to the following questions from the vendor of your third-party device:

1. What is the enterprise of the device?
2. How many enterprise (private) traps does the device have?
3. What does each private trap mean?
4. What are the corresponding MIB variables used by each private trap?

Finally, use the answers you get for the above questions to edit the corresponding sections in the TRAP.INI file.

Exporting Logged Data to Other Software

All databases created by EliteView are in dBASE format. Thus any software supporting dBASE files can read EliteView data.

To export logged data into other software,

1. Bring up the Log Data Manager.
2. Highlight the logged data you wish to copy to another application. Do this by holding down the left button on the mouse and dragging the mouse over the required lines.
3. Select *Copy* from the *Edit Menu* of the Log Data Manager. This will copy the selected data into the Windows clipboard.
4. Open the application that you want to use. For example, load Microsoft Word if you want to copy the information to it.
5. Paste the information from the clipboard onto the Microsoft Word file. And then save it if necessary.

APPENDIX B

CUSTOMIZING ELITEVIEW

EliteView is a powerful network management platform that is designed to meet all your needs. To maximize system usability and functionality, EliteView takes full advantage of all user and programming interfaces available in the Microsoft Windows 95, 98, 2000 and NT environments.

You can easily customize EliteView by editing the NETMGR.INI, TRAP.INI and the configuration files found in the EliteView directory (**C:\EV60**). These files use the general format of the Windows initialization file. This chapter describes the contents of these **INI** files.

EliteView's Initialization Files

The initial settings for the EliteView program modules are found in several different initialization files. The default directory for initialization files is C:\EV60. To view these files, use any word processor or text editor (e.g., NOTEPAD.EXE). Initialization files are also included for the platform program and several other EliteView modules as described below. You may modify these files to meet your particular needs. (However, be sure to maintain backup copies of the original intialization files.) Detailed examples of how to modify these files are provided in the following sections.

Inside the NETMGR.INI File

The initial settings for the main EliteView program are found in NETMGR.INI. The topics included in this file are listed below.

Description of Sections in NETMGR.INI

Section	Description
device	Describes each device that can be managed by EliteView.
tools	Items that appear in the Tools menu or the main program.
util	Items that appear in the Utilities menu or the main program.
bitmaps	Contains the bitmap graphics used to display managed devices on the network map.
MIB	Directory where device MIBs are stored.
LOG	Directory for storing log files; also includes a flag which enables/disables logging upon startup.
tftp	Path for boot files; also includes setting for device connection.
startup	Indicates programs to run when EliteView is invoked.
system	Settings for desktop layout; also includes parameters for device connection.
discover	Includes default settings for Auto Discover; is updated each time Discovery is executed.
MESSAGE.DLL	Governs inter-module communication of messages.
Event Manager	Contains database pointers associated with the print option.
RMON	Contains several polling variables.

Changing Parameters in NETMGR.INI

You may edit any of the sections included in NETMGR.INI to meet the needs of your specific environment. However, a few of the more common changes include the following items.

- You may add any Windows-based application to the *tools* and *utilities* menus. Edit the [tools] section to add or delete tools; or edit the [util] section to add or delete utilities.
- You may change any of the bitmaps provided by EliteView or add additional bitmaps for unlisted devices. Bitmap graphic images are defined in the [bitmap] section. Each bitmap must be in Windows v3.1 BMP format. The first bitmap is used to represent the normal state of the device; the second bitmap the down state; and the third bitmap the "unmonitored" state.

The NETMGR.INI file follows the Windows initialization file format. It is divided into different sections; section names are indicated by square brackets for easy identification, e.g., [system].

These sections contain different parameters, occupying one line each. Parameters are presented in the following format:

```
keyword=value1,value2,...
```

Parameter format conventions:

1. Keywords and values may be alphanumeric characters.
2. No spaces are allowed before or after equal signs or commas.
3. Some values may use single spaces inside, however, no consecutive spaces are allowed.
4. Some parameters may mention "EliteView applications," providing that :
   - The Windows-based applications are written according to a set of rules required by EliteView
   - They accept a list of parameters provided by the EliteView main program.
   - They are able to communicate with each other through the EliteView message center.
   - All EliteView applications started from the main EliteView program are automatically terminated when the main EliteView program is terminated. However, non-EliteView applications are not affected when the main EliteView program is terminated, even if they were started within the main EliteView program.

The [system] Section

The [system] section contains parameters that are used by basic components of EliteView, such as the protocol stacks. There are two ways to change parameter settings in this section. They are:

- Using the Default Settings command in the Options Menu.
- Making changes directly inside the NETMGR.INI file. To do this you should know the meaning of each item.

Description of Parameters in NETMGR.INI

Parameter	Description	Example
Community=	The default community string that appears in the Add Object dialog box.	Community=public
Map=	Default map opened by the EliteView main map when it starts. If no default map is assigned, it opens with an empty map.	Map=SMC
Polling=	Default polling interval, measured in seconds. This appears in the Add Object dialog box.	Polling=1
Timeout=	Default timeout period that appears in the Add Object dialog box.	Timeout=1
Retries=	The number of default retries that appear in the Add Object dialog box	Retries=3

The [device] Section

The [device] section controls the number of devices EliteView can manage. The Add Object dialog box should display a complete list. Each device is refered to by a unique name. You can edit this section to add devices.

Tip: Adding a device will require editing not only the [device] section, but also other sections as well. For example, you may wish to define the tools that can be applied to the device by editing the [tools] section. You also need to add names of bitmaps that show the device onscreen in the [bitmap] section.

To add a new device,

Edit the `total=n` line to specify the number of devices you wish to manage. The variable n is a positive number, which specifies the number of devices that will appear in the list of objects (i.e., Add Object dialog box).

Add a line describing the device, using the following format:

```
Seq=Device_Name,Protocol,Device_Description,Device_Type,Object_ID,Device_Manager
```

Parameter Definitions for the [device] Section

Parameter	Description
Seq=	The sequence number of the device runs from 1 to n, where n is the total number of devices you can add to a map or monitor.
Device_Name	The device name, which is used by EliteView to identify the device.
Protocol	A number representing the transport protocol used for communicating with the target device. (0) SNMP/UDP/IP, (1) IP Node, (2) SNMP, (3) IPX Node
Device_Description	The name of the device that appears in the Add Object dialog box.
Device_Type	The identifier for the device, which is used by EliteView to identify the device. 5000-5499 - SMC device 1000- 2000 - third-party device
Object_ID	MIB variable “SysObjectID” of SNMP Agent.
Device_Manager	Set to DevMan if this device can be opened by the Device Manager, or NoDevMan if it cannot.

Example: Sample entry for [device] section

```
[device]
total=12
1=GenNode,0,SNMP Node,1000, NoDevMan
.
.
.
10=EZ Stack10,0,EZ Stack10,5002,1.3.6.1.4.1.202.10.2
.
.
.
```

The following information is listed in this sample:

- These items are included in the Add Object dialog box.
- You can monitor these device types, namely, a Generic Node (i.e., third-party SNMP device) also called an SNMP node, and SMC’s EZ Stack 10.
- Both of these devices use the UDP/IP protocol.

The [tools] Section

The [tools] NETMGR.INI section describes the menu items that appear in the Tools Menu of the main EliteView program. You can add, delete, or change items in the Tools Menu simply by editing this section in the NETMGR.INI file.

To edit the [tools] section:

1. Edit the `total=n` line to specify the number of items listed in the Tools Menu. The variable n specifies the number of items listed under this menu.
2. Add a line describing each item in the [tools] section using the following format:

Parameter Definitions for the [tools] Section

Parameter	Description
Seq=	The sequence number is from 1 to n, where n is the total number of items in the Tools Menu.
EV_Flag	Difference between EliteView and Non-EliteView Windows applications. Acceptable values include: 0 Designates the entry as a "Non-EliteView Windows application"; i.e., it cannot work closely with EliteView, but may be initiated from within some EliteView commands as an independent process. 1 Designates the entry as an "EliteView Application"; i.e., it follows the set of rules specified by SMC to allow interaction with other EliteView modules.
Menu_Item	This is the text that appears in the menu. An ampersand "&" before one of the letters in the text designates a short-cut key sequence (i.e., a single-letter used with the Alt key), which you can press to invoke the corresponding command. This is normally shown onscreen with an underline.
Executable	The name of the Windows application that is invoked when the corresponding item is selected. This application program is set as the default application, which means that it is invoked only if no other definition for individual devices exists.
Help_Message	The message that appears in the status bar when the specified menu item is browsed.
Toolbar_Bitmap	The toolbar bitmap shown in the EliteView platform program.

The above list may be followed by zero or more lines for definitions that apply to individual devices.

3. Identify particular tools for any device, if necessary. Do this by adding or editing the following command line:

```
Device_ID.Seq=Executable
```

The following describes each item:

Parameter	Description
Device_ID	The device identifier defined in the [device] section.
Seq	The menu item sequence number the device will override.
Executable	Filename of the EliteView application that is invoked when the device type and menu items are selected.

If no application is designated for a device, then the default application listed in the preceding table is invoked when the corresponding menu item is selected.

Example: Sample entry for [tools] section

```
[tools]
total=5
1=1,&Zoom,SNMPTREE.EXE,Zoomin the object,zoom
2=1,&Alive Test,ALTEST.EXE,Alive test,altest
3=1,&MIB Browser,SNMPTREE.EXE,MIB browser,tree
4=1,M&IB-2 Viewer,MIB2VIEW.EXE,MIB-2(RFC1213) Viewer, mib2view
5=1,&Telnet,TELNET.EXE,Telnet,telnet
```

From the [tools] section, one can read the following information:

- Five menu items are listed in the Tools Menu - `Zoom`, `Alive Test`, `MIB Browser`, `MIB-2 Viewer`, and `Telnet`.
- Run each item by invoking the corresponding executable file. For example, run `Zoom` by invoking `ZOOM.EXE`, etc.
- Invoking any command will display the corresponding help message in the status bar. For example, clicking *Zoom* will display *Zoom in the object.*
- For the SNMP node, there is no override. This means that when Zoom and MIB Browser are selected, SNMPTREE.EXE is executed.
- For SMC devices, the Zoom command invokes the corresponding EliteView management module; the Alive Test command invokes ALTEST.EXE; and the MIB Browser command invokes SNMPTREE.EXE.

The [bitmaps] Section

The [bitmaps] NETMGR.INI section lists the filenames of graphic bitmaps that are used to show devices on the EliteView maps. The graphic bitmaps are in standard *.BMP format. If you want to change any of the bitmaps EliteView provides, you can create you own and integrate them into EliteView. You can do this simply by copying a new bitmap into the bitmap directory and editing the [`bitmap`] section of the NETMGR.INI file.

Tip: If you are going to design your own graphic bitmap for a device, remember to design three kinds of graphic images -

- the device at normal operating condition
- the device when it is “down” or not operating
- the device in the “not monitored” mode

To edit the [bitmaps] section,

1. Identify the directory where your graphic bitmap file(s) are located by editing the first line in the [`bitmap`] section. This parameter identifies the directory where the bitmaps are found. It follows the following syntax: `path=directory`
2. Enumerate the graphic bitmaps to include using the following format:
   `Device_ID=bitmap1,bitmap2,bitmap3`

The following table describes each item:

Parameter	Description
Device_ID	The device identifier defined in the [device] section.
bitmap1	The bitmap used to represent the normal status of the device.
bitmap2	The bitmap used to represent "down" status of the device.
bitmap3	The bitmap used to represent the "not monitored" status of the device.

Any bitmap size is acceptable. When entering the bitmap filename, the BMP filename extension should not be included.

Example: Sample entry for [bitmap] section

```
[bitmap]
path=C:\EV60\BITMAP\
Ether1000=ether1,ether2,ether3
GenNode=gen1,gen2,gen3
Bridge1=bri1,bri2,bri3
CompRemote=remote1,remote2,remote3
PC=pc1,pc2,pc3
submap=submap
```

The following information is included in the [bitmap] section:

- Bitmap files are in the C:\EV60\BITMAP subdirectory.
- For example, the `Device_ID EZ Stack10` uses:

  ```
  ether1(bitmap1 file)to represent it at normal status
  ether2(bitmap2 file)to represent it at "down" status
  ether3(bitmap3 file) to indicate it is "not monitored"
  ```

- The same explanation follows for the other Device_IDs mentioned, e.g., GenNode, Bridge1, CompRemote and PC.

The [util] Section

The [`util`] section controls the menu items that appear in the Utility Menu of the EliteView platform program. You can add, delete or change items in the Utilities Menu by editing this section in the NETMGR.INI file.

To edit the [util] section,

1. Edit the `total=n` line to specify the number of items listed in the Utilities Menu. The variable n is a positive number that specifies the number of items listed under this menu.
2. Add a line describing each item in the [tools] section using the following format:

   `Seq=EV_Flag,Menu_Item,Executable,Help_Message`

Parameter Definitions for the [util] Section

Parameter	Description
Seq=	The sequence number from 1 to n, where n is the total number of items in the Tools Menu.
EV_Flag	Acceptable EV_Flag values include: 0 Designates entry as a "Non-EliteView Windows application." It cannot work closely with EliteView, but may be initiated within some EliteView commands as an independent process. 1 Designates the entry as an "EliteView Application;" and that it follows the set of rules specified by SMC to allow interaction with other EliteView modules.
Menu_Item	This is the text that appears in the menu. An ampersand "&" before one of the letters in the text designates an Alt-key short-cut command sequence, which you can press to invoke the corresponding command. This is normally shown on screen with an underline.
Executable	The name of the Windows application that is to be invoked when the corresponding item is selected. This application program is set as the default application, which means that it is invoked only if no other definition for an individual device exists.
Help_Message	The message that appears in the status bar when the specified menu item is browsed.
Toolbar_Bitmap	The toolbar bitmap shown in the EliteView platform program.

Example: Sample entry for [util] section

```
[util]
total=10
1=1,&Log Manager,LOGMAN.EXE,Log manager,log
2=1,Log &Database Manager,LOGDATA.EXE,Log database manager,data
3=1,&BOOTP Server,BOOTP_DB.EXE,BOOTP server,bootp
4=1,&TFTP Server,TFTPSVR.EXE,TFTP server,tftp
5=1,&Report,REPORT.EXE,Report window,report
6=1,Tra&p Manager,TRAPMAN.EXE,Trap manager,trap
7=1,&Event Manager,EVENT.EXE,Event manager,event
8=1,&Name Database Manager,NBMGR.EXE,Name database manager,nbmgr
9=1,D&iscovery,DISCOVER.EXE,Discovery,discover
10=1,&MIB Compiler,MIBCOMP.EXE,MIB compiler,mibcomp
```

From the [util] section, you can see the following information:

- There are 10 items or options in the Utilities Menu (total=10).
- The first item in the menu is the Log Manager. You can run it by selecting it from the Utilities Menu; or by simply typing <L> while the Utilities Menu (pull-down type) is displayed onscreen.
- When your cursor is positioned on the Log Manager option, the status bar at the bottom of your screen will display "Invoke Log Manager" to describe the Log Manager.
- When you run the program you invoke LOGMAN.EXE

The [tftp] Section

The [tftp] section controls the list and settings of the items that appear in the TFTP Server dialog box. You can add, delete or change these settings.

Example: Sample entry for [tftp] section

```
[tftp]
public=C:\EV50\PUBLIC
timeout=5
retry=3
```

From the `[tftp]` section, you can view the following data:

- Section name is `[tftp]`
- All TFTP files are stored in C:\EV50\PUBLIC
- The timeout value is 5; and retry value is 3

The [startup] Section

You can customize EliteView to run other programs before loading the EliteView platform program. For example, notice that when you click on the EliteView icon from the EliteView program group, the Event and the Trap Manager programs are automatically executed by default. Editing the [startup] section of the NETMGR.INI file will allow you to include additional startup processes.

To edit the [startup] section:

1. Edit the `total=n` line to specify the number of processes you wish to start.
2. Add a line describing each item in the [`startup`] section using the following format:

   `Seq=EV_Flag,Menu_Item,Executable,Optional parameter`

Parameter Definitions for the [startup] Section

Parameter	Description
Seq=	The sequence number from 1 to n, where n is a positive number equal to the total number of startup processes you wish to invoke before running the EliteView platform program.
EV_Flag	Distinguishes between EliteView and non-EliteView applications. Acceptable values include: 0 Designates the entry as a “Non-EliteView Windows application.” This means it cannot work closely with EliteView, but may be initiated within some EliteView commands as an independent process. 1 Designates an “EliteView Application;” i.e., it follows the set of rules specified by SMC to allow interaction with other EliteView modules.
Menu_Item	This is the text that appears in the menu. An ampersand “&” before one of the letters in the text designates an Alt-key short-cut command sequence, which you can press to invoke the corresponding command. This is normally shown on screen with an underline.
Executable	The name of the Windows application to be invoked when the corresponding item is selected. This application program is set as the default application, which means that it is invoked only if no other definition for an individual device exists.
Optional Parameters	Command line parameters of the program. Optional entry.

Example: Sample entry for [startup] section

```
[startup]
total=2
1=1,EVENT.EXE
2=1,TRAPMAN.EXE
```

From the [startup] section, you can view the following data:

- There are two applications that will be loaded with the main EliteView program.
- These are the Event Manager and the Trap Manager.

The [discover] Section

This section determines the settings for protocol selection and polling parameters under the Discovery module. These settings are determined by the choices you make during Configuration and within the Discovery Setup menu (page 4-3).

Parameter Definitions for the [discover] Section

Parameter	Description
protocol	Determines the initial protocol selected when the Discovery module is opened. Values: UDP, IPX, Ethernet
autosave[1]	Saves options including currently selected protocol type and other entries under the Discovery Setup menu. Refer to the last item in the Discovery Setup menu. If you choose autosave, then all the items in this table, other than broadcast, are updated.
scanrate[2]	The search rate at which query messages are transmitted.
repoll[2]	The number of times to query for device response.
ipnode[1]	Display IP nodes without an SNMP agent.
ipxnode[1]	Display IPX nodes without an SNMP agent.
nwserver[1]	Display Novell servers.
broadcast[1]	Used to disable the Broadcast button in the Discovery window. Note that if you select the *Distint* TCP/IP WINSOCKET platform under Network Setup, the broadcast button will be disabled.

1 - Values: on, off

2 - Values: 1~10

Inside the TRAP.INI File

EliteView's Trap Manager configuration is described in the TRAP.INI file. The sections included in the TRAP.INI file are listed below.

Description of Sections in TRAP.INI

Section	Description
generic	Provides the generic template for all traps.
enterprise	Provides trap information for specific devices.

The TRAP.INI file defines how the Trap Manager parses trap messages from SNMP devices and converts them into events, which are then sent to the Event Manager.

Like the NETMGR.INI file, the TRAP.INI file is divided into different sections, namely, the [`generic`] section, the [`enterprise`] section and the specific trap sections.

Elements of a Trap Message

Element	Description
enterprise	In a generic trap, this is the object identifier of the device that generates the trap (sysObjectID). In a specific trap, this field contains an identifier used to differentiate the definition domain of the trap.
agent-addr	The network address of the device that issues the trap. If IP protocol is used, the agent-addr is the IP address of the device.
generic-trap	An integer value that identifies the type of trap. Allowable values: 0 to 5 Identifies common conditions that occur in IP network operations; i.e., indicates that the trap is a device generic trap. 6 Indicates that the trap is a device-specific trap
specific-trap	An integer that identifies different device conditions; defined by the device vendor.
time-stamp	The sysUpTime variable. This is the time period from the initialization of the device to the moment the trap is issued. This time period is expressed in hundredths of a second.
variable-bindings	A list of MIB variables with values related to a particular event.

The [generic] Section

The first section of the TRAP.INI file is the [generic] section. It defines the number of generic traps available and identifies the events each trap triggers. Entries for this section use the following format: `n=Active_Flag,Event_Name,Message`

The following table describes each item:

Parameter	Description
n	Number of generic traps. Value ranges from 0 to 5
Active_Flag	Select either 0 or 1, where 1=active; 0=disabled (not processed)
Event_Name	The name of the event you wish to trigger
Message	The message displayed on the message box

Example: Sample entry for [generic] section

```
[generic]
0=1,Trap,Cold Start!!
1=1,Trap,Warm Start
2=1,Trap,Link Down
3=1,Trap,Link Up
4=1,Trap, Authentication Failure
5=1,Trap,egpNeighborLoss
```

From the [generic] section, you can view the following data:

- Trap number 0
- It is active (1)
- It triggers the “Trap” event
- When invoked the message “Cold Start” will be displayed in the event message box

The [enterprise] Section

This section lists and defines all enterprises used in the program. Enterprises are object identifiers used in generic and specific traps.

To edit the [enterprise] section,

1. Edit the total=n line to specify the total number of available enterprises. The variable n is a positive number that specifies the number of items listed under this menu.
2. Add a line describing each item in the [enterprise] section using the following format:

   ```
   i=Active_Flag,Enterprise_ID,Number_of_Traps
   ```

The following table describes each parameter:

Parameter	Description
i=	The sequence number is from 1 to n, where n is the total number of enterprises.
Active_Flag	Select either 0 or 1, where 1=Active and 0=Disabled.
Enterprise_ID	The object identifer of the enterprise.
Number_of_Traps	Number of specific traps defined for this enterprise.

Example: Sample entry for [enterprise] section

```
[enterprise]
total=1
1=1,1.3.6.1.4.1.259,15
```

From the [`enterprise`] section, one can read the following information:

- The entry defines a single enterprise (total=1)
- The enterprise is active with object identifier equal to 1.3.6.1.4.1.259
- This enterprise defines 15 traps

Specific Trap Sections

Each specific trap is defined in a section of its own. The section name takes the following format:

```
[Ent(n).(m)]
```

where:	`(n)`	is the sequence number of the enterprise
	`(m)`	is the specific trap number

To edit specific traps:

1. Define trap message using the following format:

   `message=string`

   The string variable can be any character string with some components generated from the information in the variable list. You can use variables in the form of %x, where x is any letter from a to z.

2. Define variables generated from trap messages using the following format:

   `x=Object_name,Symbol`

   The following table describes each item:

Parameter	Description
x=	Variable name, any letter from a to z.
Object_name	MIB object name defined in the MIB database.
Symbol	A symbol used to identify the use of a value or index in the object name. • The variable binding uses the format object.instance (where instance is the index of a table entry if a table is being referred to); and is equal to "0" for single value variables. • You can use the value of the variable, which is represented with "v" or one of the indices. • The indices are represented as a list of numbers. Long indices like the IP address take four numbers. The format for indices is **`in`**, where `n` is a positive number, e.g., `i2`.

3. Define the event to trigger using the following format:

   `event=Active_Flag,Event_Name`

   The following describes each item:

Parameter	Description
event=	The event to trigger.
Active_Flag	Select either 1 or 0; 1=Active, 0=Disabled.
Event_Name	Name of the event or trap to trigger.

Example: Sample of specific trap entry

```
[Ent1.1]
message=Temperature over 65 degrees
event=1,Trap

[Ent1.2]
message=Port auto-partitioned
event=1,Trap

[Ent1.3]
message=Port bad link
event=1,Trap
```

```
[Ent1.4]
message=Hub %b collision count over %a
event=1,Trap
a=groupTotalCollisions,v
b=groupTotalCollisions,i5

[Ent1.5]
message=Hub %b alignment error count over %a
event=1,Trap
a=groupFAEErrors,v
b=groupFAEErrors,i5

[Ent1.6]
message=Hub %b CRC error count over %a
event=1,Trap
a=groupCRCErrors,v
b=portCRCErrors,i5

[Ent1.7]
message=Port(%b,%c)length < 64 bits count over %a
event=1, Trap
a=portPygmys,v
b=portPygmys,i5
c=portPygmys,i6

[Ent1.8]
message=Port (%b, %c) CRC error count over %a
event=1, Trap
a=portCRCErrors, v
b=portCRCErrors,i5
c=portCRCErrors,i6
```

From the preceding example, you can view the following data:

- There are 8 specific traps Available for enterprise number 1.
- All traps are active (event=1).
- When a trap is activated, it will trigger a specified event called “Trap” and display the corresponding message on screen. Supposed trap number 4 is triggered [Ent1.4].
- Trap number 4 contains the group variable TotalCollisions in its variable-bindings, which has 5 numbers.
- Supposed that trap number 4 is triggered and the 5th number in the index of groupTotalCollisions is 2, and the value of the variable is 20. The Event Manager will display the following message on screen: Hub 2 collision count over 20.

APPENDIX C
SNMP ENVIRONMENT

EliteView uses Simple Network Management Protocol (SNMP), the most popular network management protocol. SNMP was developed by the Internet Engineering Task Force (IETF) using the Internet Protocol (IP). SNMP was originally designed to run on top of the UDP/IP transport protocol. EliteView currently supports transport protocols including IP and IPX.

SNMP Roles

There are two defined roles in SNMP: manager and managed. The *manager* (EliteView) sends request messages and the *managed* (device) sends responses and alarm messages.

SNMP uses a simple command-response style to request information or action. By sending a command to perform an action, the manager is able to control the managed device. After the action is performed, the managed device sends a response to indicate that the requested action has been completed.

Each command-response dialog is independent. The managed device does not maintain a “session” with the manager.

Managing Data

Data in monitored devices are defined using the Management Information Base (MIB) model. Database management functions are built into agent software using standard data structures. SNMP is based on the “Consice MIB Definitions,” which are defined in RFC 1212.

Here is a portion of the EZ Stack 10 MIB:

```
EZ Stack-10 DEFINITIONS ::= BEGIN
 IMPORTS
        OBJECT-TYPE, Counter, TimeTicks ,IpAddress
        FROM RFC1155-SMI;
        IMPORTS
      DisplayString
        FROM RFC1213-MIB-II;

        enterprises    OBJECT IDENTIFIER ::=
       { iso(1) org(3) dod(6) internet(1) private(4) 1}
        smc OBJECT IDENTIFIER ::=
      { enterprises 202 }
        hmBasicCapability
     OBJECT IDENTIFIER ::= { smc 1}
        hmSelfTestCapability
     OBJECT IDENTIFIER ::= { smc 2}
        hmPerfMonCapability
     OBJECT IDENTIFIER ::= { smc 3}
        hmAddrTrackCapability
     OBJECT IDENTIFIER ::= { smc 4}
```

The MIB itself has a hierarchical structure, defining objects in a tree-like structure. Using this model, the entire world can be defined from a single origin.

Objects

Real objects in the world of network management are variables that contain values. Data types may be numbers, character strings or structures. New data types may be created by renaming existing data types, limiting the range of values or structuring them using pre-defined methods. The SNMP MIB uses only a small portion of the data types and structuring methods defined in ASN.1. See IETF documents RFC 1212 and 1213 for more information.

table.index notation

Objects are organized into *tables*. Each table represents a group of *objects* that may have multiple instances; for example, the hrDeviceEntry table in the HR MIB contains values for device index, type, description, ID, status, and errors. Each object has a set of values representing its operating status. While the “dotted” notation for the hrDeviceEntry table is 1.2.6.1.36.2.1, the notation for the hrDeviceIndex field is 1.2.6.1.36.2.1.1. Multiple instances in a table are identified by an *index* number.

By definition, all single MIB variables have the index 0. Only numbers are used as indices; if other data types are used as indices, then they are represented by a list of numbers separated by dots (also referred to as dotted-decimal notation).

iso origin

All objects in the SNMP world begin with the object ISO, which has a unique identifier of 1. In other network implementations, objects may begin with CCITT (2) or joint-ISO CCITT (3). Each object can be represented by an object which can uniquely identify itself on the object tree. The notation starts with the root object and each level represents its descendants.

Example: MIB-2 Identifier

The MIB-2 identifier is known as:

- iso.org.dod.internet.mgmt.mib2, or
- 1.3.6.1.2.1

All common network variables are defined below this object.

Example: A DOS Filename Analogy

For example, MS-DOS uses a hierarchical file naming structure. Initialization information is stored in the EV60 subdirectory. The full path name is

```
C:\EV60\NETMGR.INI
```

which means that the file is:

- Located on drive C
- Directory is EV60
- File name NETMGR.INI

The `.INI` extension indicates that the file follows the standard Windows convention for program initialization. By comparison, the `EV60.EXE` file is an executable file and `EV60.HLP` is a help file for the EliteView program.

Branches

Some common branches to the iso origin include:

Object identifier	Numeric identifier
iso.org.dod.internet.mgmt.mib2	1.3.6.1.1.2.1
iso.org.dod.internet.mgmt.private.enterprises	1.3.6.1.4.1
iso.org.dod.internet.mgmt.private.enterprise.smc	1.3.6.1.4.1.202

For example, SMC is assigned an identifier of 202 under “enterprise.”

Appendix D

Technical References

Information about networking and protocols is available from RFC reports and industry-related documentation.

RFC Reports

The most comprehensive collection of networking information is a series of reports called Request for Comments (RFC). Each RFC has a title and an RFC number, such as Internet Protocol, RFC-791. RFCs are all listed in an index, titled rfc-index. The index is not definitive and titles do not always indicate the contents (for example, RFC-1155 and RFC-1156 define SNMP requirements although SNMP is not mentioned in the title). To obtain one or more RFCs on paper, contact the Network Information Center (NIC) at:

Government Systems, Inc.
Attn: Network Information Center
4200 Park Meadow Drive - Suite 200
Chantilly, VA 22021

Help desk: 1-800-365-3642 or 1-703-802-4535
(Hours: 7:00am to 7:00pm, Eastern Time)
Fax number: 1-703-802-8376
e-mail: nic@diis.ddn.mil

RFC Reports

RFC Number	Title	Publisher/Year
RFC-768	User Datagram Protocol	SRI International, 1980
RFC-783	Trivial File Transfer Protocol (TFTP)	SRI International, 1981
RFC-791	Internet Protocol	SRI International, 1982
RFC-792	Internet Control Message Protocol	SRI International, 1980
RFC-793	Transmission Control Protocol	SRI International, 1981
RFC-854	Telnet Protocol	SRI International, 1980
RFC-1060	Assigned Numbers	SRI International, 1980
RFC-1033/103	Domain Name Protocol	SRI International, 1987
RFC-1042	A Standard for Transmission of IP Datagrams over IEEE 802 Networks	SRI International, 1988
RFC-1155	Structure and Identification of Management Information for TCP/IP-based Internets	SRI International, 1990
RFC-1156	Management Information Base for Network Management of TCP/IP-based Internets	SRI International, 1990
RFC-1157	SNMP	SRI International, 1990
RFC-1166	Internet Numbers	SRI International, 1990
RFC-1213	MIB-II	SRI International, 1991
RFC-1286	Definitions of Managed Objects for Bridges	SRI International, 1991
RFC-1298	SNMP over IPX	SRI International, 1992
RFC-1350	The TFTP Protocol	SRI International, 1992
RFC-1368	Definitions of Managed Objects for IEEE 802.3 Repeater Devices	SRI International, 1992
RFC-1420	SNMP over IPX	SRI International, 1993
RFC-1493	Definitions of Managed Objects for Bridges	SRI International, 1993
RFC-1514	Host Resource MIB	SRI International, 1993
RFC-1516	Definitions of Managed Objects for IEEE 802.3 Repeater Devices	SRI International, 1993
RFC-1525	Definitions of Managed Objects for Source Routing Bridges	SRI International, 1993

Industry-Related Documentation

Also refer to these industry-related documents:

- The Ethernet: a Local Area Network, Data Link Layer and Physical Layer Specification Standard (Digital-Intel-Xerox; also known as the Blue Book)
  IEEE Std 802.2 - 1985 (ISO/DIS 8802.2)
  IEEE Std 802.3 - 1985 (ISO/DIS 8802.3)
- Internetworking with TCP/IP: Principles, Protocols, and Architecture, Vol. 1 (Douglas Comer, Prentice Hall, 1990)

APPENDIX F

SPECIFICATIONS

EliteView is designed to manage any workgroup using the SNMP network management protocol.

Product Overview

- Event-driven, scalable, modular architecture
- Heterogeneous device management with discovery
- Object-oriented, Microsoft® Windows® 95, 98 or NT application with menus-on-demand, drag-and-drop icons and MDI (multiple document interface)
- Share information with Windows Clipboard and dBASE®-compatible files

Features and Specifications

	EliteView
Architecture	
Open platform	Permits addition of user-designed management modules
Scalable, modular design	Over 20 modules
User Interface	
Microsoft Windows	Windows 95, 98 Windows NT 3.5x, 4.0
Menus-on-demand	Brings up floating menus with one click of (secondary) right mouse button
Drag-and-drop icons	Move icons to any map view (lockable)
Supports MDI (Multiple Document Interface)	Allows users to open several submaps simultaneously and view individual numeric or graphic display for processor or network interface statistics
Hot keys	Short cut keystrokes invoke certain actions; Press <INS> to add object; Press <Del> to delete object; <F1> for on-line help
User-definable icons	Choose any Windows metafile (BMP) graphic images

Performance Management	
Tune network for optimum performance	Selectable statistical polling intervals
Traffic filters	Data value, date, time with comparison operators (<, >, <=, >=, ==, !=) and logic operators (AND, OR)
Quantify and graph network throughput	Quantify any MIB value (SNMP or private). Real-time scalable graphics support for SMC products
Record network activity in database to plan future growth	View database directly or with third-party application
Fault Management	
Network status	Device Up, Connection Lost, and Trap
Detect errors	Isolates problem down to system device
Prioritized error log	User-definable
User-definable events and actions	Any one or more of these actions: Audible Signal, Show Message, Run Program, Report, and Write Into Database
Record network errors in database to anticipate future problems	View database directly or with third-party application
ICMP station response diagnostic	Interactive diagnostic
Inventory and Configuration Management	
Hierarchical network map	Unlimited levels
Manage heterogeneous devices	Manages all SMC adapters. Also manages non-SMC SNMP products with SNMP MIB
SNMP MIB tree browser	Interactive browser with SET, GET and GET NEXT commands
Database of network devices	Catalogs any SNMP variable for any device
Resource and Load Management	
Print detailed and summary reports	Interactive compilation of station reports with time stamp. Print or transfer to any word processor or spreadsheet.
Data Transfer	
Integrated, relational database	Number of records limited only by disk space*
File format compatibility	dBASE IV
Maximum number of nodes	Limited only to disk space*
Transfer with Windows Clipboard	Print or transfer to any word processor or spreadsheet
System Requirements	
Software	Microsoft Windows 95, 98 or NT 3.51 (or later)
Hardware (minimum)	80486DX2-66, 16 MB RAM, 30 MB hard disk*, VGA color monitor, and a mouse
Hardware (recommended)	Pentium-133, 32 MB RAM, 40 MB of free space on a hard disk, Super VGA color monitor, mouse

* *Disk space required to install EliteView is about 30 MB using a local or network hard disk. User files (network map, data logs, etc.) will vary depending on your network configuration and statistical tracking requirements.*

APPENDIX G

CODEBASE 6.0 DLL SUB-LICENSE AGREEMENT

This section contains a statement of agreement between Sequiter Software Inc. and the CodeBase 6.0 LICENSEE (SMC) concerning sub-licensing (specifically for software used in the file C4DLL.DLL). All the terms and conditions in this agreement are imposed upon the CodeBase 6.0 DLL SUB-LICENSEE (i.e., the party purchasing the software described in this manual).

This legal document is an agreement between you, the CodeBase 6.0 DLL SUB-LICENSEE, and the CodeBase 6.0 LICENSEE (hereinafter referred to as the "Agreement").

You are not being "sold" any Sequiter Software Inc. software. Instead, you are being granted the right to use Sequiter Software Inc. software through this license agreement. Sequiter Software retains all ownership of its software including all copies of its software.

1. Definitions

Software

This is the Sequiter Software Inc. computer programs contained in the CodeBase 6 software package or any computer programs containing parts of the computer programs in this package. These programs could be in any form: in print, as electronic source code, as compiled object modules, as a library file, a dynamic link library, or an executable program.

Executable Software

This is a form of the Software which can be executed by DOS, Microsoft Windows or OS/2 software packages.

Distributable Software

This is the Executable Software except for the CodeReporter executable program.

DLL Software

This is a dynamic link library form of the Software which is executed indirectly under DOS, Microsoft Windows or OS/2 software packages. It includes Microsoft Windows and OS/2 dynamic link library forms of the software. For the purpose of this license agreement, other forms of the Software which are executed indirectly, such as the an AutoCad ".EXP" form of the Software, are also considered to be DLL Software.

2. Sub-License

You may use the DLL Software with, and only with, the Distributable Software provided by the CodeBase 6 LICENSEE. You may not use the DLL Software for any other purpose. Specifically, you agree not to use the DLL Software for the purposes of developing or creating Executable Software.

3. Transfer Restrictions

The DLL Software is sub-licensed to you, and may not be transferred to anyone without the prior consent of the CodeBase 6.0 DLL LICENSEE. Any authorized transferee of the sub-license shall be bound by the terms of this agreement.

4. Disclaimer

The DLL Software is provided “as is” without any kind of warranty. It is your responsibility to determine whether the DLL Software is suitable for your purpose.

5. Miscellaneous

This agreement is governed by the laws of the Province of Alberta, Canada. The CodeBase DLL SUB-LICENSE consents to jurisdiction in the province of Alberta, Canada.

APPENDIX H

TROUBLESHOOTING

This section summarizes the most common error messages generated by EliteView.

EliteView Map: Icon Stays Red

Symptoms

1. When a map is opened, the icon(s) stay red.
2. When an object is added, the icon stays red.

Possible Causes

EliteView cannot communicate with the device. When a device does not respond within the Retries limit, the device is assumed to be off-line and the "Connection Lost" event is announced. The icon turns red; and EliteView continues polling for a response. EliteView will continue polling for responses (unless the Monitor option is turned off in the Add Object or Modify Object selections under the Edit menu).

Suggestions

1. There may be a problem with the device driver or network cabling used on the Network Management Station (NMS). Use a hardware or software network test utility to verify that the NMS can receive network traffic.
2. Verify the IP address of the object.
3. Probe the concerned device with the Alive Test. If it responds to this query but not to EliteView, check the community setting for the device.
4. Check the Retries and Timeout values of the object.

Discovery: No Nodes Found

Symptom

Discovery does not find any devices.

Possible Causes

1. EliteView network management station (NMS) may be using an IP address that is used by another device.
2. There may be a cabling problem.

Suggestions

1. Try changing the address of the EliteView NMS.
2. If EliteView is running, you should see a trap and an alarm indicating that a device has been restarted. From the Alarm Log, copy the IP address into the search criteria and attempt to discover this node. If the node still cannot be located, check for cabling or other logical problems.

Trap Manager: MIB Variable Not Found

Symptom

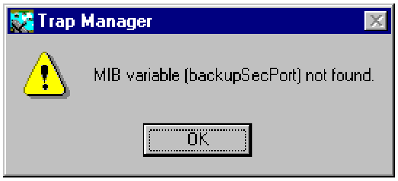
Running Trap Manager shows this error message.

Possible Causes

1. The specified variable does not exist in the MIB database.
2. The MIB database has been changed.

Suggestions

Verify that the specified variable does exist in the MIB database.



Trap Manager

MIB variable (backupSecPort) not found.

APPENDIX I

ERROR MESSAGES

The error messages related to the overall EliteView platform are described in this appendix. Error messages relating to specific network devices are listed in the corresponding EliteView manual. The following error messages are sorted by program module. First match the error message's label with the module name in this appendix, and then look up the error message in alphabetical order.

EliteView

A device should have three bitmap files in [bitmap] section of NETMGR.INI.
Cause: The number of bitmap files for a device is incorrect.
Action: Check the [bitmap] section of NETMGR.INI.

A submap should have two bitmap files in [bitmap] section of NETMGR.INI.
Cause: The number of bitmap files for a submap is incorrect.
Action: Draw two bitmap files for a submap object.

Bitmaps for a device should have the same size.
Cause: Some bitmap files for a device are not the same size.
Action: Redraw the bitmap and make its size equal to other bitmaps.

Cannot find toolbar bitmap.
Cause: Toolbar bitmap does not exist.
Action: Draw a new bitmap for the missing item.

Cannot load xxx file.
Cause: Cannot load xxx file as background bitmap.
Action: Choose another bitmap file.

Cannot move a submap to its child.
Cause: An attempt was made to move a submap to its child map.
Action: This operation not permitted.

Cannot open submap window.
Cause: File may be corrupt.
Action: Recreate a new map file.

Cannot read bitmap file.
Cause: The bitmap file defined in the [bitmap] section of NETMGR.INI may be corrupt.
Action: Recreate the bitmap file.

Cannot read bitmap file for submap.
Cause: The bitmap file of a submap object may be corrupt.
Action: Recreate the bitmap file for the submap.

Cannot run the program (xxx)
Cause: File xxx may be corrupt or does not exist.
Action: Reinstall the concerned file.

Create file error (xxx)
Cause: Create file xxx operation failed.
Action: Check disk space.

[device] section error in NETMGR.INI.
Cause: The format of a device entry in the [device] section of NETMGR.INI is invalid.
Action: Check the [device] section of NETMGR.INI.

Invalid filename.
Cause: The file containing default settings may be corrupt or does not exist.
Action: Change the default setting.

Invalid format for [tools] or [util] in NETMGR.INI.
Cause: Bad format in [tools] or [util] section.
Action: Check the [tools] or [util] section of NETMGR.INI.

Invalid old password.
Cause: Incorrect old password.
Action: Input correct old password.

Invalid or duplicate device name in NETMGR.INI.
Cause: A device name defined in the [device] section of NETMGR.INI is duplicated or invalid.
Action: Check the [device] section of NETMGR.INI.

Invalid password.
Cause: Incorrect password.
Action: Type correct password.

Invalid polling interval.
Cause: The specified value is out of range.
Action: The value for the polling interval must be greater than 1 and less than 86400.

Invalid retries value.
Cause: The specified value is out of range.
Action: The value for retries must be greater than 1 and less than 1000.

Invalid target address.
Cause: The address format is incorrect for the current protocol.
Action: Change to the correct address format for the current protocol.

Invalid timeout value.
Cause: The specified value is out of range.
Action: The value for timeout must be greater than 1 and less than 86400.

Link file is corrupt.
Cause: The connection information between objects is corrupt.
Action: Recreate the map file.
Map file and link file mismatch.
Cause: File may be corrupt.
Action: Recreate the map file.

Map file is corrupt.
Cause: File is corrupt.
Action: Recreate the map file.

No protocol available, program terminates ....
Cause: None of the protocol drivers can be loaded.
Action: Check the system environment.

The toolbar bitmap size of xxx is not the same as others.
Cause: The size of the bitmap file xxx is not the same as other toolbar bitmap files.
Action: Redraw the bitmap.

The (total) for [tools] or [util] in NETMGR.INI cannot be 0.
Cause: The total value in the [tools] or [util] section is 0
Action: Include at least one item in the [tools] or [util] section.

Write file error (xxx)
Cause: Write file xxx failed.
Action: Check disk space.

BOOTP Server

Create Dialog windows not successful.
Cause: May be out of memory.
Action: Close some applications and try again.

database NULL!!
Cause: The record or variable cannot be found in the database.
Action: Check the database files BOOTP.DBF and BOOTP.DBT; and delete them if necessary.

database MEMO field error!!
Cause: database error.
Action: Check database files.

database Tagname error!!
Cause: database error.
Action: Check database files.

Input parameter invalid!!
Cause: Input parameters are not correct.
Action: Retype the parameters.

Invoke toolbar not successful.
Cause: May be out of memory.
Action: Close some applications and try again.

Open database error!!
Cause: Open database file failed.
Action: See if the database file exists; or check that hard drive is OK.

Out of memory!!
Cause: Out of memory.
Action: Close some applications and try again.

Out of Memory for allocation.
Cause: Out of memory.
Action: Close some applications and try again.

Record not Found!!
Cause: The input parameter cannot be found in the database.
Action: Check database files.

Write file error!!
Cause: Write to hard drive error.
Action: Delete some files for more space.

BOOTP.DLL

Bind to UDP socket failed with error code = %d!
Cause: Bind Socket fail. Error code comes from function bind( ) of WinSocket specification.
Action: Check the network subsystem by error code.

Call WSAStartUp( ) function failure!
Cause: Call WsaStartup fail, with unknown error code.
Action: Check the version of winsock.dll.

Cannot allocate callback routine.
Cause: Out of Memory.
Action: Close some applications and retry, or restart windows.

Cannot run BTPIFM.EXE. or Cannot run BTPIFM.EXE with error code %d.

Cause: Initial BOOTP.DLL error; may be out of memory.

Action: Close some applications and retry, or restart windows. The error code comes from the SDK WinExec( ) API function.

Cleanup WinSocket failed with error code %d when exiting BTPIFM.EXE!

Cause: The Unregister operation from winsock.dll failed. The error code should refer to the WSACleanup( ) function in the WinSocket specification.

Action: Check the network subsystem by error code.

Close socket failed with error code %d when exiting BTPIFM.EXE!

Cause: Attempt to close the socket BOOTP failed. This error code comes from the function closesocket( ) in the WinSocket specification.

Action: Check the network subsystem by error code.

Enable receive broadcast frame error with error code %d.

Cause: ENABLE receive broadcast option error. Assigned error code comes from the function setsockopt( ) in the WinSocket API.

Action: Check the network subsystem by error code.

Get bootp service info error.

Cause: The error code comes from the function getservbyname( ) of the WinSocket specification.

Action: Check the network subsystem by error code.

Got FD_READ message, but got length error with error code %d.

Cause: Received a packet, but the network subsystem cannot report the length. The error code comes from the function ioctlsocket( ) of WinSocket API.

Action: Check the network subsystem by error code.

Memory allocation is not successful.

Cause: Out of memory.

Action: Close some applications and retry, or restart windows.

No Usable WinSock.dll found.

Cause: The Call WsaStartup failed with an unknown error code.

Action: Check the version of winsock.dll.

Open UDP Socket failed with error code = %d!

Cause: The Open Socket opeation fail. The error code comes from the function socket( ) of the WinSocket API.

Action: Check the network subsystem by error code.

Send packet failed with error code %d.

Cause: The network subsystem is unable to send a packet. The error code comes from the function sendto( ) in the WinSocket specification.

Action: Check the network subsystem by error code.

Setup receive message error with error code = %d.

Cause: Cannot setup a Windows message for received frames. The error code comes from the WSASelect( ) function in the WinSocket specification.

Action: Check the network subsystem by error code.

The BOOTP application is installed.

Cause: BOOTP.DLL only supports the BOOTP Server application.

Action: Never use two BOOTP Server applications.

The network subsystem is not ready!

Cause: The TCP/IP stacks for the platform may not be ready.

Action: Check your WinSocket environment setting.

The version of winsock.dll should at least support ver. 1.1

Cause: The version of winsock.dll is too old.

Action: Upgrade the TCP/IP stacks platform.

The windows socket's version specified by the application is not supported by this winsock.dll!

Cause: The version of winsock.dll did not match the requirement of BOOTP.DLL.

Action: Change TCP/IP stacks and winsock.dll.

Windows socket version %d.%d required, but not supported by winsock.dll.

Cause: The version of winsock.dll did not match the requirement of BOOTP.DLL.

Action: Change the TCP/IP stacks and winsock.dll.

Winsock.dll was not able to support the minimum number of sockets BOOTP module required.

Cause: Too many WinSocket applications are running.

Action: Close some WinSocket applications and retry.

Discovery

Cannot find %%%%%%% or this bitmap file has problem!

Cause: The bitmap file %%%%%%% is missing or the format is not correct.

Action: Get the correct file of %%%%%%%.

Cannot find IP general node in [device] section of NETMGR.INI.

Cause: No IP general node information in NETMGR.INI.

Action: Use a text editor to add it.

Cannot find IPX general node in [device] section of NETMGR.INI.

Cause: No IPX general node information in NETMGR.INI.

Action: Use a text editor to add it.

Cannot find IPX Server device in [device] section of NETMGR.INI.
Cause: No IPX Server general node information in NETMGR.INI.
Action: Use a text editor to add it.

Cannot find SNMP general node in [device] section of NETMGR.INI.
Cause: No SNMP general node information in NETMGR.INI.
Action: Use a text editor to add it.

Create Dialog windows not successful.
Cause: The main windows of discover.exe cannot be successfully created.
Action: Close other applications and try again.

Create status window not successful.
Cause: May be out of memory.
Action: Close some applications and try again.

Dump all ETHERNET objects to namebase failed.
Cause: Write to hard drive failed.
Action: The hard drive space may not be enough.

Dump all IPX objects to namebase failed.
Cause: Write to hard drive failed.
Action: The hard drive space may not be enough.

Dump all UDP objects to namebase failed.
Cause: Write to hard drive fail.
Action: The hard drive space may not be enough.

Dump function cannot work when searching for objects.
Cause: Dump function can only work when Discover is busy.
Action: Wait until the search job is complete.

Initial namebase failure.
Cause: Name database files have a problem.
Action: Use name database manager to verify.

Invoke alive test program failed with error reason code = %d.
Cause: The error code comes from the return code of SDK's function WinExec.
Action: Refer to the Microsoft SDK function reference.

Invoke MESSAGE.DLL failed!
Cause: This program may not have quit successfully last time.
Action: Quit Windows and test again.

Invoke toolbar not successful.
Cause: Create toolbar window failure.
Action: Close other applications and try again.

IPX Diagnostic Service Socket open failure.
Cause: The network sublayer has a problem.
Action: Use the alive test program to verify.

No bitmap file information in [bitmaps] of NETMGR.INI.
Cause: Cannot find any bitmap file information in NETMGR.INI.
Action: Use a text editor to edit NETMGR.INI.

No device information in [device] of NETMGR.INI.
Cause: Cannot find any device information in NETMGR.INI.
Action: Use a text editor to edit NETMGR.INI.

None of the protocols are enabled, please use network setup to configure again.
Cause: The record in NETMGR.INI is not correct.
Action: You should enable at least one protocol; use the network setup program to configure again.

None of the SNMP/UDP or SNMP/IPX protocols work!
Cause: All protocols supported by discover.exe are not working.
Action: Check to see if your network subsystem is OK.

Not enough memory to allocate channel parameter structure.
Cause: Out of memory.
Action: Close some applications and try again.

Not enough memory to allocate for IPX packet data.
Cause: The memory is not enough.
Action: Close some applications and retry.

Open ICMP channel failure.
Cause: The ICMP network sublayer has failed.
Action: Memory may not be enough. Otherwise, use the alive test program to verify network status.

Open IPX channel failure. IPX protocol will be disabled!
Cause: The IPX network sublayer has a problem or has returned a NULL IPX address.
Action: Restart Windows or start another ELITEVIEW SNMP or IPX application first.

Open SNMP/IPX channel failure.
Cause: The SNMP or IPX network sublayer has failed.
Action: Retstart Windows and retry.

Open SNMP/UDP channel failure.
Cause: UDP/IP network sublayer has failed.
Action: Use the alive test program to verify the ELITEVIEW platform. If based on a third-party vendor's platform, make sure it is OK.

Search for MIB object not successful.
Cause: Cannot find the sysObjectID variable in the MIB database.
Action: Make sure the MIB database contains MIB II information.

The autosave= entity of [discover] in NETMGR.INI is not correct!
Cause: The content of autosave= entity is not correct; it should be ON or OFF.
Action: Use a text editor to edit NETMGR.INI.

The ipnode= entity of [discover] in NETMGR.INI is not correct!
Cause: The content of ipnode= entity is not correct; it should be ON or OFF.
Action: Use a text editor to edit NETMGR.INI.

The ipxnode= entity of [discover] in NETMGR.INI is not correct!
Cause: The content of ipxnode= entity is not correct; it should be ON or OFF.
Action: Use a text editor to edit NETMGR.INI.

The nwserver= entity of [discover] in NETMGR.INI is not correct!
Cause: The content of nwserver= entity is not correct; it should be ON or OFF.
Action: Use a text editor to edit NETMGR.INI.

The path= entity of [bitmaps] in NETMGR.INI is not specified.
Cause: Cannot find the path information for bitmap files.
Action: Use a text editor to edit NETMGR.INI.

The protocol= entity of [discover] in NETMGR.INI is not supported.
Cause: The content of protocol= entity is not correct; should be UDP, IPX or ETHERNET.
Action: Use a text editor to edit NETMGR.INI.

The retry value should be in the range of 1 to 10!
Cause: The retry value is out of range.
Action: Use a text editor to edit NETMGR.INI.

The scan value should be in the range of 1 to 10!
Cause: The scan value is out of range.
Action: Use a text editor to edit NETMGR.INI.

The total= entity of [device] in NETMGR.INI is not specified.
Cause: The total= entity must be specified.
Action: Use a text editor to edit NETMGR.INI.

The value of repoll in NETMGR.INI is not in the range of 1 to 10!
Cause: The repoll valude is out of range.
Action: Use text editor to edit NETMGR.INI

There are two nodes with the same IP address - %%!.
Cause: An IP address has been duplicated.
Action: Write down the MAC addresses for troubleshooting.

Event Manager

Out of memory.
Cause: Not enough memory.
Action: Close some applications and retry.

Write Error! Event disabled
Cause: Disk full.
Action: Free up some disk space and try again.

ICMP.DLL

The following error messages may occur when running EliteView under a WinSocket platform.

Bind ICMP.DLL to ICMP socket failed with error code = %d!
Cause: The Bind Socket operation failed. Error code comes from the function bind( ) of the WinSocket specification.
Action: Check the network subsystem by error code.

Call WSAStartUp of Winsock.dll failed.
Cause: The WSAStartUp operation failed, but with no error code.
Action: Check the network subsystem.

Cannot allocate callback routine.
Cause: Out of memory.
Action: Close some applications and retry, or restart Windows.

Cannot run ICMPIFM.EXE - error code %d.
Cause: An error occurred initializing ICMP.DLL; you may be out of memory. The error code comes from the WinExec( ) function in the Windows SDK.
Action: If memory is not enough, close some applications and retry, or restart Windows. Otherwise take the action indicated by the error code.

Cleanup WinSocket failed with error code %d!
Cause: the Unregister operation from winsock.dll failed. The error code refers to the WSACleanup( ) function in the WinSocket specification.
Action: Check the network subsystem by error code.

Cleanup WinSocket failed with error code %d when exiting ICMPIFM.EXE!
Cause: The Unregister operation from winsock.dll failed. The error code refers to the WSACleanup( ) function in the WinSocket specification.
Action: Check the network subsystem by error code.

Close icmp socket failed with error code = %d when exiting ICMPIFM.EXE!

Cause: The Close socket operation failed. This error code comes from function closesocket( ) in the WinSocket specification.

Action: Check the network subsystem by error code.

Enable receive broadcast frame error - error code %d.

Cause: Enable receive broadcast option error. The assigned error code comes from the function setsockopt( ) in the WinSocket API.

Action: Check the network subsystem by error code.

Get protocol information failed for icmp.

Cause: The error code comes from the function getservbyname( ) of the WinSocket specification.

Action: Check the network subsystem by error code.

Got FD_READ message, but got length error - error code %d.

Cause: Received a packet, but the network subsystem cannot report the length. The error code comes from the function ioctlsocket( ) of the WinSocket API.

Action: Check the network subsystem by error code.

ICMP communication channel has terminated.

Cause: This is an internal error of the application program.

Action: Please contact SMC Technical Support for help.

Memory allocation is not successful.

Cause: Out of memory.

Action: Close some applications and retry, or restart Windows.

No Usable WinSock.dll found.

Cause: The WinSocket network subsystem has failed.

Action: Check the network subsystem.

Open RAW Socket failed with error code = %d!;

Cause: The Open Socket operation failed. The error code comes from the function socket( ) of the WinSocket API.

Action: Check the network subsystem by error code. Make sure your platform supports the optional RAW Socket.

Receive frame for socket %d. Not for ICMP socket %d.

Cause: A Receive process error in winsock.dll.

Action: Check the network subsystem.

Send request failed with error code = %d.

Cause: The network subsystem has failed. The error code comes from the function sendto( ) in the WinSocket specification.

Action: Check the network subsystem by error code.

Setup receive message error - error code = %d.

Cause: Cannot setup a Windows message for receive frames. The error code comes from the WSASelect( ) function in the WinSocket specification.

Action: Check the network subsystem by error code.

The application of session number does not exist.
So, the program cannot close communication channel.

Cause: This is an internal error of the application program.

Action: Please contact SMC Technical Support for help.

The network subsystem is not ready!

Cause: The network subsystem may not be the correct platform.

Action: Check for errors in platform.

The protocol stack is saturated.

Cause: The ICMP module cannot process applications at a high layer than your setting.

Action: Modify ProtocolStack in the [icmp] section of the network configuration file network.ini.

The session number of application is duplicate.

Cause: Another client application has the same Windows handle registered to ICMP.DLL.

Action: May have a program error; restart Windows and retry.

The slot is saturated.

Cause: Too many client applications of ICMP.DLL are running.

Action: Close some client applications of ICMP.DLL and retry.

The windows socket's version specified by application is not supported by this winsock.dll!

Cause: The version of winsock.dll did not match the requirements of ICMP.DLL.

Action: Change the TCP/IP stacks and winsock.dll.

The winsock.dll should at lease support version 1.1

Cause: The version of winsock.dll is too old.

Action: Upgrade the TCP/IP stacks platform.

Windows socket version %d.%d required, but not supported by winsock.dll.

Cause: The version of winsock.dll did not match the requirements of ICMP.DLL.

Action: Change the TCP/IP stacks and winsock.dll.

Winsock.dll not able to support the minimum number of sockets required by ICMP module.

Cause: Too many WinSocket applications in use.

Action: Close some WinSocket applications and retry.

IPX.DLL

See ICMP.DLL

Log Manager

Duplicate LOG event.
Cause: Requested LOG event is a duplicate.
Action: Provide a unique key including a MIB variable label, target node address, and index.

Invalid filter.
Cause: Incorrect filter format.
Action: Refer to the filter formula section in Chapter 7 of this User's Guide.

Invalid threshold.
Cause: Incorrect threshold format.
Action: Refer to threshold formula section in Chapter 7 of this User's Guide.

Invalid start time.
Cause: The specified time is incorrect.
Action: The format for time is (MMDDYYHHMM) where MM is month, DD is day, YY is year, HH is hour, and MM is minute.

Invalid stop time.
Cause: The specified time is incorrect.
Action: The format for time is (MMDDYYHHMM) where MM is month, DD is day, YY is year, HH is hour, and MM is minute.

LOG database is corrupt.
Cause: The database file is corrupt.
Action: Recreate the database file.

MESSAGE.DLL

The following are error messages that may be generated by any module.

Invalid number.
Cause: Invalid number for polling interval, timeout or retry.
Action: Enter a valid number.

Invalid target address.
Cause: You input an invalid target address.
Action: Enter a correct target address.

Number out of range. (1 <= N <= 1000)
Cause: The number is out of range for “retry.”
Action: Enter a valid number.

Number out of range. (1 <= N <= 86400)
Cause: The number out of range for “polling interval” or “timeout.”
Action: Enter a valid number.

MIB Browser

Cannot create MIB tree window.
Cause: May be out of memory.
Action: Close other windows and create the MIB tree window again

Cannot create new output file.
Cause: The file already exists or the disk is full.
Action: Specify another filename or skip this action.

Cannot create profile: filename
Cause: The file already exists or the disk full.
Action: Verify the filename or free up disk space.

Cannot create statistics window.
Cause: Cannot create a statistics window.
Action: Close other statistics windows

Cannot create Toolbar.
Cause: Out of memory.
Action: Close other programs and restart the MIB Browser, or continue without using the toolbar.

Cannot find Profile Window to add object.
Cause: No Profile Window created.
Action: Create a profile window from the “Window/New Profile” menu.

Cannot load accelerators.
Cause: Out of memory.
Action: Close other programs and restart the MIB Browser, or continue without accelerators.

Cannot open file.
Cause: The file doesn’t exist.
Action: Verify that you have specified the correct path and filename.

Cannot open profile: filename
Cause: Profile not found.
Action: Verify the path and filename.

Cannot open SNMP session.
Cause: There are too many sessions or a lower layer (SNMPAPI.DLL) crashed.
Action: Close other windows based on SNMPAPI.DLL, or reset the computer and restart Windows.

Cannot read file.
Cause: Cannot read a text file from disk.
Action: Verify that the correct file was specified, or recreate the file.

Cannot read the profile.
Cause: Cannot read a profile from disk.
Action: Verify the filename or recreate the profile.

Cannot write data to output file.
Cause: Disk full.
Action: Free up disk space and try again.

Index variable(%s) listed in the INDEX clause of %s not found in MIB database.
Cause: The index variable(s) listed in the entry variable cannot be found in the MIB database.
Action: Load MIB file which contains the index variable(s) into the MIB database

Invalid object type. It must be COUNTER, INTEGER, GAUGE or TIMETICKS.
Cause: Invalid object type.
Action: Object type must be COUNTER, INTEGER, GAUGE or TIMETICKS.

Line xxx: Bad Record.
Cause: A bad record found in line xxx.
Action: None. (The record will not display in the list box.)

Line xxx: Invalid value will be discarded.
Cause: An object in line xxx has an invalid value.
Action: None. (The invalid value will be discarded and given a default value.)

Line xxx: Object not found.
Cause: The object listed in line xxx of the profile not found in the MIB database.
Action: None. (The record will not be displayed in the list box.)

No SNMP Request.
Cause: Improper command sequence.
Action: Perform SNMP Get, Set or GetNext before action (pause, resume or delete).

Next object not under current subtree.
Cause: The SNMP GetNextRequest cannot display the next subtree in the current window.
Action: Create another subtree (“Tree/Subtree” menu) which contains the object and perform the SNMP GetNextRequest again.

Object not found.
Cause: Specified object not found in MIB database.
Action: Specify exact object label or object ID again, or load the MIBs file into the MIB database.

Out of memory (UpTree).
Cause: There is not enough memory to expand the MIB tree.
Action: Close other MIB tree windows or collapse some subtrees.

Polling interval must not be less than timeout.
Cause: The polling interval must not be less than the timeout.
Action: Modify the value for the polling interval or timeout.

Root node not found.
Cause: (In MIB tree window) Specfied root object not found in the MIB database.
Action: Add a new MIB that contains the required object into the MIB database.

SNMP GetRequest Error.
Cause: Lower layer (e.g., SNMPAPI.DLL) cannot build the SNMP GetRequest.
Action: Please contact SMC Technical Support for help.

SNMP GetNextRequest Error.
Cause: Lower layer (SNMPAPI.DLL) cannot build the SNMP GetNextRequest.
Action: Please contact SMC Technical Support for help.

SNMP SetRequest Error.
Cause: Lower layer (SNMPAPI.DLL) cannot build the SNMP SetRequest.
Action: Please contact SMC Technical Support for help.

The object has no index.
Cause: Object appended with an invalid index. (i.e., index = ".0")
Action: None.

MIB Compiler

Cannot open response file: "filename"
Cause: The file cannot be found.
Action: Verifiy the filename.

Could not initialize virtual memory manager.
Cause: Out of memory.
Action: Close other application(s); or reconfigure system files (e.g., autoexec.bat and config.sys), reboot the system and try again.

DATABASE corrupted.
Cause: MIB database error.
Action: Rebuild the MIB database.

%s has already been imported.
Cause: A duplicate MIB Name has been found in line xxx of the “mib_file.”
Action: If they are different MIB modules, rename one of the modules, and then compile it into the database. If they are the same, reload the latest module, and compile the new MIB.

Trap error.
Cause: Trap defined in MIB file is invalid.
Action: Fix trap defiinition and recompile.

Object list is not a tree.
Cause: You did not define or import some needed nodes.
Action: Check all undefined node or syntax listed in the import list, fix any errors, and recompile.

The parser had an error.
Line: %d
Message: %s
Cause: MIB macro clause has some errors. You may have forgotten to define some mandatory groups.
Action: Check the MIB definition, fix any errors, and recompile.

Node “%s” syntax is undefined in the file and syntax database.
Cause: You defined a node with invalid syntax.
Action: Check the node’s syntax clause, fix it, and then recompile.

%s, %s ... defined in module %s could not be found. Please import this module first.
Cause: The MIB you are compiling is trying to import a node or syntax from another MIB.
Action: Compile the required MIB first.

Error happened in merge tree node.
Cause: MIB Compiler cannot merge the current MIB file into the database.
Action: Check the MIB file defiinition.

Syntax node “%s” is undefined both in the MIB file and the syntax database.
Cause: You used undefined syntax in the node definition.
Action: Check the syntax in the MIB file, fix any errors, and import it in the import list.

%s near line %d has an invalid subidentifier
Cause: Variable has a bad node subidentifier in line xxx.
Action: Correct the node subidentifier and compile again.

%s near line %d has a duplicated node in the MIB file.
Cause: The variable is duplicated in line xxx.
Action: Rename the variable and compile again.

ACCESS %s in %s near line %d is unknown.
Cause: The ACCESS clause of a variable has an invalid value in line xxx.
Action: Correct it and compile the MIB file again.

Status "%s" in %s near line %d is unknown
Cause: The STATUS clause of a variable has an invalid value in line xxx.
Action: Correct it and compile the MIB file again.

MIB Compiler cannot decide object identifier of "%s" near line %d.
Cause: Cannot find the parent of a node (variable).
Action: Correct the node definition and compile again.

Merge Tree Error.
Cause: Some ancestor node(s) cannot be found for new_variable in line xxx.
Action: Correct the node definition and compile again.

%s near line %d is a self-defined identifier.
Cause: An improper node definition.
Action: Correct it and compile again.

Error: Peers with the same ID (node1 = node2 = subidentifier)
Cause: The subidentifiers of node1 and node2 are the same.
Action: Correct them and compile again.

%s in MIB database and %s in MIB file have the same object identifier but different name.
Cause: The object identifiers of new_node and old_node are the same.
Action: Correct them and compile again.

The lexical analyzer could not recognize the token.
Line: %d
Message: %s
Cause: Unacceptable character in MIB field.
Action: Check MIB file and fix it.

Warning: **MIB Compiler cannot find the import node "%s" in the database. But it found a node with the same name in the module "%s". Do you want to continue.**
Cause: In an MIB file, an import node is composed by name-module pair. If this warning has happened, it means the exact (name, import module name) can not be found in the MIB database, but another node with the same name but a different module name (name, different module name) exists in database.
Action: If you continue to process this MIB file without importing the correct one, errors may occur. You should import the correct MIB list in the import list first.

Index "%s" is not defined in the database or MIB file.
Cause: The MIB Compiler cannot find the referenced index node in the database.
Action: You should check if the index is defined in other MIBs. If yes, add the MIB in the import list. Otherwise, you should define the index node first.

MIB.DLL

Cannot find any traps for the MIB module.
Cause: No traps in the MIB database.
Action: None.

Cannot find next node.
Cause: The next MIB variable cannot be found.
Action: None.

Cannot find the node's parent.
Cause: The MIB variable has no parent.
Action: None.

Cannot find the specific node.
Cause: The MIB variable cannot be found in the MIB database.
Action: Load new MIBs into the MIB database.

Cannot find the specific trap.
Cause: Trap not found.
Action: Load new MIBs into the MIB database.

Cannot locate previous position.
Cause: (Internal error.) The MIB database is inconsistent.
Action: Please contact Accton Technical Support for help.

database is corrupt.
Cause: The MIB database is corrupt.
Action: Rebuild the MIB database.

Invalid search mode.
Cause: (Internal error.) An application used the wrong search mode.
Action: Please contact Accton Technical Support for help.

MIB database error.
Cause: (General error.)
Action: Please contact Accton Technical Support for help.

Out of memory.
Cause: Out of memory.
Action: Close other programs and retry.

MIB-2 Viewer

Cannot find the text any more!
Cause: The text you specified cannot be found at the other position.
Action: None.

Cannot find this text.
Cause: The text you specified cannot be found.
Action: Verify text.

Cannot open statistics window.
Cause: System may be out of resources.
Action: Close some applications and retry.

Create root window failed.
Cause: System may be out of memory and resource.
Action: Close some applications and retry.

Initial Tree failure.
Cause: System error.
Action: Reset the system and retry. If the same error occurs, contact Accton Technical Support.

Mib2 viewer cannot get any SNMP response from target.
Cause: Have received no SNMP response frame.
Action: Make sure the target address is correct, the network subsystem OK, and the target node supports this variable group.

Mib2 viewer got no snmp response for variable.
Cause: The objects may be not implemented.
Action: None.

Out of Memory.
Cause: Out of memory.
Action: Close some applications and retry.

Output Window out of space.
Cause: Output window is full.
Action: Save contents, and then clear the window with a New Output command.

SNMPAPI reported allocation memory error.
Cause: The Windows system is out of memory.
Action: Close some applications and invoke again.

Report

Cannot access printer.

Cause: Cannot create device context.

Action: Memory may not be enough.

Cannot find the text specified.

Cause: Report cannot find the specified text.

Action: Check the case, and be sure the text is correct.

File is too large, the content of file will be truncated.

Cause: The file to be opened is too large.

Action: Use another program such as write.exe to open it.

Printer error.

Cause: Possible reasons may be:

a. General error.

b. Not enough disk space available for spooling, and no more space will become available.

c. Not enough memory is available for spooling.

d. User terminated the job through the printer manager.

Action: Check printer or disk space. If memory not enough, print directly, not through the printer manager.

Reports contents have changed. Save it?

Cause: The contents of report the window have been changed.

Action: Save it if important.

TFTP Server

Cannot open local file.

Cause: The TFTP Server cannot locate the specified file.

Action: Make sure the file exists and that the path in the SETUP dialog box is correct.

Cannot read public directory.

Cause: The public directory is not correct or the directory has too many files.

Action: Modify the public directory setting.

Create TFTP Server main window failed!

Cause: The TFTP Server cannot create the main window.

Action: Close some applications and retry.

Network Setup program cannot run with the other modules!
Please close the other modules first.

Cause: Another EliteView module was open when running the network setup program, or the EliteView program closed abnormally and left some module resident in memory.

Action: Close all EliteView programs or restart Windows.

Create TFTP Server status window failed!
Cause: The TFTP Server cannot create the status bar.
Action: Close some applications and retry.

Initialize mib database failed, the download function will be disabled.
Cause: The MIB may not contain the SMC MIB.
Action: Use the MIB compiler to add the SMC MIB.

Initialize namebase failed.
Cause: The name database dynamic library failed to initialize.
Action: The Namebase.dbf file may be corrupted, use the namebase manager to check it.

Invalid Retry value.
Cause: The retry value should be between 0 and 2000.
Action: Change it.

Invalid target address.
Cause: The target IP address or name is not correct.
Action: Use the name in the name database or correct the IP address.

Invalid Timeout value.
Cause: The timeout value should be between 0 and 2000.
Action: Change it.

Open SNMP communication channel failed!
Cause: The network subsystem may have failed.
Action: Use the alive test program to make sure the network subsystem is OK.

Open TFTP communication channel failed!
Cause: The network subsystem may have failed.
Action: Use the alive test program to make sure the network subsystem is OK.

Out of memory.
Cause: Memory is not enough.
Action: Add system memory or close some applications and retry.

Run AccInfo failed!
Cause: tftpsvr.exe may have abnormally terminated the last time it was run.
Action: Restart Windows and retry.

TFTP.DLL

Bind to UDP socket failed with error code = %d.

Cause: The Bind Socket operation failed. The error code comes from the function bind( ) of the WinSocket specification.

Action: Check the network subsystem by error code.

Cannot allocate callback routine.

Cause: Out of memory.

Action: Close some applications and retry, or restart Windows.

Cannot open IFM window.

Cause: An error occurred initializing TFTP.DLL. May be out of memory.

Action: If memory is not enough, close some applications and retry, or restart Windows. Otherwise take the action indicated by the error code.

Cleanup WinSocket failed with error code %d when closing IFM window.

Cause: The Unregister operation from the winsock.dll failed. The error code refers to the WSACleanup( ) function in the WinSocket specification.

Action: Check the network subsystem by error code.

Close socket failed with error code %d when closing IFM window.

Cause: A Close socket operation used by the TFTP.DLL failed. The error code comes from the function closesocket( ) in the WinSocket specification.

Action: Check the network subsystem by error code.

Enable receive broadcast frame error - error code %d.

Cause: Enable receive broadcast option error. The error code comes from the function setsockopt( ) in the WinSocket API.

Action: Check the network subsystem by error code.

Get tftp service information error.

Cause: The error code comes from the function getservbyname( ) of the WinSocket specification.

Action: Check the network subsystem by error code.

Got FD_READ message, but got length error - error code %d.

Cause: Have received a packet, but the network subsystem cannot report its length. The error code comes from the function ioctlsocket( ) of the WinSocket API.

Action: Check the network subsystem by error code.

Memory allocation is not successful.

Cause: Out of memory.

Action: Close some applications and retry, or restart Windows.

No usable WinSock.dll found.

Cause: The WinSocket network subsystem failed. Verify the platform.

Action: Check the network subsystem.

Open UDP socket failed with error code = %d.

Cause: An Open Socket operation failed. The error code comes from the function socket( ) of the WinSocket API.

Action: Check the network subsystem by error code.

Receive frame from wrong socket %d.

Cause: The Receive process in winsock.dll had an error.

Action: Check the network subsystem.

Send packet failed with error code %d.

Cause: The network subsystem has a problem. The error code comes from function sendto( ) in the WinSocket specification.

Action: Check the network subsystem by error code.

Setup received message error - error code = %d.

Cause: Cannot setup a Windows message for received frames. The error code comes from the WSASelect( ) function in the WinSocket specification.

Action: Check the network subsystem by error code.

The network subsystem is not ready.

Cause: The network subsystem may not be the correct platform.

Action: Verify the platform.

The port number of application is duplicate.

Cause: Two client programs are using the same port number for the IP protocol. Some TFTP application may not have quit normally.

Action: Check your applications.

The slot is saturated.

Cause: TFTP.DLL can only support 10 sessions.

Action: Never open more than 10 sessions.

The windows socket's version specified by application is not supported by this winsock.dll.

Cause: The version of winsock.dll did not match the requirement for TFTP.DLL.

Action: Change the TCP/IP stacks and winsock.dll.

The winsock.dll should at least support version 1.1

Cause: The version of winsock.dll is too old.

Action: Upgrade the TCP/IP stacks platform.

Action: Check the network subsystem by error code.

Windows socket version %d.%d required, but not supported by winsock.dll.
Cause: The version of winsock.dll did not match the requirement for TFTP.DLL.
Action: Change the TCP/IP stacks and winsock.dll.

Winsock.dll not able to support minimum number of sockets required by TFTP module.
Cause: Too many WinSocket applications are in use.
Action: Close some WinSocket applications and retry.

Trap Manager

Invalid format in section [enterprise] entry xxx.
Cause: An invalid format in TRAP.INI entry xxx.
Action: Correct it and restart Trap Manager. (Refer to Appendix B in this User's Guide for information on writing TRAP.INI)

MIB variable (var_label) not found.
Cause: MIB variable(s) for specific Trap(s) not found in the MIB Database.
Action: Load the MIB(s) which contains these variable(s) into the MIB Database, and restart the Trap Manager.

Out of memory.
Cause: Out of memory.
Action: Close other programs and retry.

Trap Manager cannot register trap channel.
Cause: (Internal error.) Cannot register a trap channel.
Action: Contact SMC Technical Support.

APPENDIX J

PRODUCT SUPPORT SERVICES

SMC is an international corporation that offers worldwide technical support. Many of these services are available 24 hours a day.

To help us serve you more effectively, please have the following information available before calling SMC:

- AUTOEXEC.BAT
- CONFIG.SYS
- SYSTEM.INI
- A list of files in your EliteView directory (C:\EV60)
- Detailed information about your network adapter

Technical Support Questions

Be prepared to answer the following questions.

Questions	Example
Version of EliteView your are using?	6.0
Network adapter: manufacturer and model?	SMC EZ Card 10/100 (SMC1211TX/WL)
Exact wording of any error messages you may have seen?	Packet Driver not loaded
Step-by-step description of what happened and what you were doing when the problem occurred?	1. Double-clicked on EliteView icon 2. Saw error message
Description of the actions you have taken to correct the problem?	Re-booted computer
Anything else we should know about your computer or installation?	Computer soaked by Great Flood of '93

For Technical Support, Call:

From U.S.A. and Canada (8:30 AM - 8:00 PM Pacific Time)
(800) SMC-4-YOU; (949) 707-2400; (949) 707-2460 (Fax)
From Europe (8:00 AM - 5:30 PM UK Greenwich Mean Time)
44 (0) 1189 748740; 44 (0) 1189 748741 (Fax)

Internet

E-mail addresses:	techsupport@smc.com european.techsupport@smc-europe.com
Driver updates:	http://www.smc.com/support.html
World Wide Web:	http://www.smc.com/
FTP Site:	ftp.smc.com

GLOSSARY

EliteView
EliteView is a complete network management platform. EliteView is composed of a core program and group of related modules. EliteView is a complete network management product with modules for managing SMC and third-party SNMP devices.

Address
Identification of entities in a communication protocol.

BOOTP (Boot Protocol)
BOOTP is a popular protocol that runs on top of the UDP/IP stack. BOOTP is used by devices to discover their own IP address. In EliteView, the BOOTP server provides the services of IP addresses and filenames.

Broadcast Packet
A packet transmitted to all nodes attached to the network.

Community
A character string embeded in SNMP messages that’s used to authenticate the access rights of the service requester.

Connection
A logical binding between two or more users of a service.

Ethernet
A 10 Mbps baseband LAN that uses a bus configuration and CSMA/CD.

Ethernet Frame
A packaging structure for Ethernet data and control information. It consists of the destination address, source address, field length, data, pad, and a frame check sequence.

Gateway
A synonym for router in internet protocol (IP). A gateway connects several IP networks. It builds a routing database by exchanging information with other gateways or with information input by the network administrator. It relays IP data packets between connected networks.

IP Address
A 32-bit quantity representing a point of attachment to the Internet. It is usually represented by four 8-bit integers separated by dots. Each decimal integer represents a byte in an IP address. The IP address is divided into a network part and a host part. For example, 192.9.211.151. See Appendix E for more information on Internet and IP addresses.

IPX
Internetwork Packet Exchange is a NetWare protocol providing datagram message delivery.

Local Area Network (LAN)
A group of interconnected computers and other devices.

MAC Address
Media Access Control address that represents a unique physical address for each port in a local area network.

Map
A network diagram showing devices managed by EliteView.

MIB
An acronym for Management Information Base. It is a set of objects that contain information about the device. Note that MIB-2 is simply a subordinate component of the overall MIB.

Multicast Packet
A packet transmitted to a specified set of nodes on the network.

Netmask
The netmask divides a network into logical subnets. This number uses a binary representation to include or exclude address. For example, these netmasks correspond to various internet classes:

Class A: 255.0.0.0
Class B: 255.255.0.0
Class C: 255.255.255.0

Network Management Station (NMS)
The computer used to run network management software, e.g., EliteView. Messages from EliteView are displayed on the NMS.

Out-of-Band
A way of communicating with a network device from outside the standard network channels.

Packet
The unit of data transfer over a local area network. For Ethernet, it includes the number of preamble bits, the start of frame delimeter, the destination and source addresses, the data to be transferred, and the frame check sequence (CRC) bytes.

Protocol
A set of rules that allows computers to communicate with one another, specifying the format, timing, sequencing, and error-checking for data transmission.

Retries

The number of times EliteView attempts to communicate with a device before quitting. Also see timeout.

SNMP
Simple Network Management Protocol (SNMP). The application protocol offering network management services in the Internet suit of protocols.

Software Interrupt
The software interrupt is the communication channel between the device driver (e.g., ODIPD or NWPD) and other applications (e.g., WINPD or ELITEVIEW). The recommended software interrupt is 0x60 (60 hex).

Subnet
A LAN segment which may be reached through a gateway.

Subnet Mask
A decimal number (four integers) which specifies the logical network subnet of the terminal. A typical subnet mask is 255.255.255.0. Also see Netmask.

Unicast Packet
A packet transmitted to a specific node on the network.

Timeout
The elapsed time (in seconds) that EliteView waits for a response from a device.

TFTP (Trivial File Transfer Protocol)
A file-transfer protocol for downloading files.

WINSOCKET
Provides a common network programming interface for Microsoft Windows that allows applications using TCP/IP software from different vendors to be easily implemented.

INDEX

FOR TECHNICAL SUPPORT, CALL:

From U.S.A. and Canada (8:30 AM - 8:00 PM Pacific Time)
(800) SMC-4-YOU; (949) 707-2400; (949) 707-2460 (Fax)
From Europe (8:00 AM - 5:30 PM UK Greenwich Mean Time)
44 (0) 1189 748740; 44 (0) 1189 748741 (Fax)

INTERNET

E-mail addresses: techsupport@smc.com
european.techsupport@smc-europe.com
Driver updates:
http://www.smc.com/support.html
World Wide Web:
http://www.smc.com/
FTP Site:
ftp.smc.com

FOR LITERATURE OR ADVERTISING RESPONSE, CALL:

U.S.A. and Canada:	(800) SMC-4-YOU;	Fax (949) 707-2460
Spain:	34-93-477-4920;	Fax 34-93-477-3774
UK:	44 (0) 1189 748700;	Fax 44 (0) 1189 748701
Southern Europe:	33 (1) 41.18.68.68;	Fax 33 (1) 41.18.68.69
Central/Eastern Europe:	49 (0) 89 92861-200;	Fax 49 (0) 89 92861-230
Nordic:	46 (8) 564 33145;	Fax 46 (8) 87 62 62
Middle East:	971-48818410;	Fax 971-48817993
South Africa:	27 (0) 11-3936491;	Fax 27 (0) 11-3936491
PRC:	86-10-6235-4958;	Fax 86-10-6235-4962
Taiwan:	886-2-2747-4780;	Fax 886-2-2747-9220
Asia Pacific:	(65) 238 6556;	Fax (65) 238 6466
Korea:	82-2-553-0860;	Fax 82-2-553-7202
Japan:	81-45-224-2332;	Fax 81-45-224-2331
Australia:	61-2-9416-0437;	Fax 61-2-9416-0474
India:	91-22-8204437;	Fax 91-22-8204443

SMC®
6 Hughes
Irvine, CA 92618
Phone: (949) 707-2400

January 2001 R01