



Advanced Single-Probe
Advanced Multi-Probe
Expert Probe
User Guide

Trademark Notices

© 1994-2004 by Network Instruments, LLC (Limited Liability Corporation). All rights reserved.

“Observer”, “Network Instruments” and the “N with a dot” logo are registered trademarks of Network Instruments, LLC, Minneapolis, Minnesota, USA.

Limited Warranty—Software

Network Instruments, LLC will replace defective media or documentation for a 60-day period after the shipment of the product from Network Instruments, LLC. Should Network Instruments, LLC release a newer version of the software within 60 days of shipment of the product, Network Instruments, LLC will update the copy of the software upon request, provided request is made by the licensed user within the 60-day period of shipment of the new version. This update may consist of a CD, or a manual, or both at the discretion of Network Instruments, LLC. User may be charged a shipping fee for updates.

Network Instruments, LLC shall not be liable for material, equipment, data, or time loss caused directly or indirectly by proper or improper use of the software. In cases of loss, destruction, or corruption of data, Network Instruments, LLC shall not be liable. Network Instruments, LLC does not take any other responsibility. Network Instruments, LLC does not warrant that the product will meet your requirements or that the operation of the product will be uninterrupted or that the product will be error-free.

NETWORK INSTRUMENTS, LLC SPECIFICALLY DISCLAIMS ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL NETWORK INSTRUMENTS, LLC BE LIABLE FOR ANY LOSS OF PROFIT OR ANY OTHER COMMERCIAL DAMAGE, INCLUDING BUT NOT LIMITED TO SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGES.

Network Instruments, LLC makes no other warranty, expressed or implied.

End User License Agreement

Network Instruments' Observer products are neither shareware nor freeware. Network Instruments' Observer products are commercial software and/or hardware products that are subject to international copyright laws.

Upon purchase and registration of the specific Network Instruments' product, you have a non-transferable right to use the specific product at **one** site on **one** LAN on **one** personal computer (PC). Additional networks can be monitored by purchasing additional Probes or Observer licenses which will grant you the right to use additional Probes, Probe instances, or consoles for each license purchased. The purchase of a Probe does not include a license for Observer. Should you need additional Observer consoles, you will need to purchase additional licenses separately.

To install Network Instruments' Observer on additional PCs or laptops, you will need to purchase an additional Observer license for each system. If you are installing Probes on PCs or laptops, you will need to purchase a Probe for each system.

Network Instruments' Observer software and license numbers are the property of Network Instruments, LLC and may not be copied by any means for purposes other than backup.

After you purchase a Network Instruments software license, you will receive license and activation numbers. These license and activation numbers are your proof of purchase. You will need to produce this information for upgrades. You may need to provide the activation numbers to receive technical support.

This software is licensed as stated above. The license does not constitute ownership of the software, only the right to use the software.

Technical Support

Network Instruments provides technical support:

By phone (depending on where you are located):

US & Countries outside Europe at (952) 932-9899

UK and Europe at +44 (0) 1959 569880

By fax (depending on where you are located):

US & Countries outside of Europe at (952) 932-9545

UK and Europe at +44 (0) 1959 569881

Or by email at:

support@networkinstruments.com

Network Instruments provides technical support for a period of 90 days after the purchase of the product at no charge. After the 90-day initial support period, support will only be provided to those customers who have purchased a maintenance agreement.

Telephone technical support hours are between 9:00am and 5:00pm (local time for each office).

Suggestions are welcomed. Many of the improvements made to our products have originated as end user suggestions. Please submit detailed suggestions in writing to: support@networkinstruments.com or by fax at: (952) 932-9545. Please submit any corrections to or criticism of Network Instruments' publications to: pubs@networkinstruments.com or by fax at (952) 932-9545.

To subscribe to the Network Instruments email newsletter (delivered in HTML format), send an email to:

listserv@networkinstruments.com

with the word "subscribe" in the subject line.

Contents

Introduction	1
Overview	1
What Probes Do	1
How Probes Operate in a Switched Environment	2
Types of Probes	2
Hardware Probes	6
Hardware/Software Requirements and Installation	6
Operating Systems Supported	6
Installation	6
Ethernet Errors By Station & NIC Driver Installation	8
Licensing the Probe	9
Dongles	10
Using the Advanced Single-Probe	11
Advanced Single-Probe Window	11
Using the Advanced Multi-Probe	14
Advanced Multi-Probe Window	15
Configuring Multi-Probe Connections	17
Configuring User Accounts for Secure Access	20
Customizing Statistics and Capture Buffers For Probe Instances	23
Setting the Total System Memory reserved for Probes	25
Using the Expert Probe	27
Advanced Probe Port Usage With Observer	28
Using the RMON Probe	29
RMON Probe Configuration	29
RMON2 Probe Window	29
Appendix	42
RMON1 Tree	42
RMON2 Tree	51
RMON1 Extension Tables	56
HCRMON Table (1.3.6.1.2.1.16 RMON2 Tree)	57
HCRMON Extensions to RMON1/2 Tables	58
FrameRelayDTE Table	63
Index	65

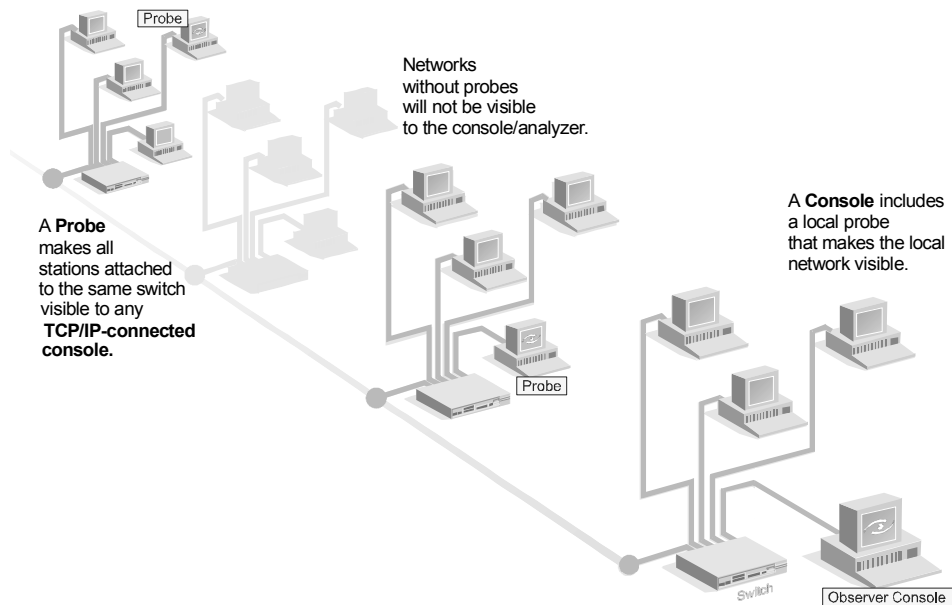


Introduction

Overview

When something goes wrong on your network, seeing what's happening on the wire can quickly lead you to a solution. Network Instruments Observer lets you see all traffic on the network to which it is connected. To monitor multiple networks from a single console, simply install a Network Instruments Probes on every network where a "point of visibility" is required.

Probes collect and report network traffic and statistics about traffic at the request of the Observer console. This enables the network administrator to detect and anticipate problems on both local and remote portions of the network.



What Probes Do

Probes collect data from the network to which they are attached (usually through a switch) and send it to an analyzer/console for display. The Probe can be located on the same system as the Console (every Observer-family analyzer includes a "local Probe"), or it can communicate with remote consoles via TCP/IP.

How Probes Operate in a Switched Environment

The purpose of a switch is to isolate traffic to the local network, thereby reducing the amount of traffic each device on that network can see and process. Although a protocol analyzer puts a network interface card in “promiscuous” mode, it still only sees packets addressed to or transmitted from the port that it is connected to on the switch.

To operate a Probe (or any type of analyzer product) in a switched environment, you must choose a method that provides network visibility to the port where the Probe is connected. Most switches provide a function that “mirrors” all packets received or transmitted from either a single port of interest (i.e. a server or router), or multiple ports of interest. The mirrored traffic can then be captured or analyzed by connecting your analyzer (or in this case, the Probe) to the “mirror port” (which is sometimes called a span port).

Switches typically provide two options for configuring the mirror or span settings. You can either use a web based interface included with your switch to set this, or use SNMP commands to set the port (or ports) to be mirrored. Network Instruments’ Probes are capable of sending the SNMP commands directly to your switch, or you may use the switch-provided method to set the span port.

In addition to port spanning, Network Instruments’ Observer products offer options to use SNMP to directly query your switch and report port-based statistics (see information on the Observer Suite), use RMON to report on any internal RMON statistics the switch may have (also included in the Observer Suite), or create a statistical picture of your switch with port looping (included in all Observer products). Selecting which method is right for you will depend on your switch, and the level of detail you need to troubleshoot the problem at hand. Remember, for capture, decode and Expert event identification, only static port mirroring will provide all the information required for a complete picture of what is happening on your network.

Types of Probes

So far this has been a generic discussion of analyzer/management consoles and probes. As with most computer software components, there are a number of different flavors of probes, both generic and proprietary. Network Instruments Observer analyzers can communicate with 5 types of Probes.



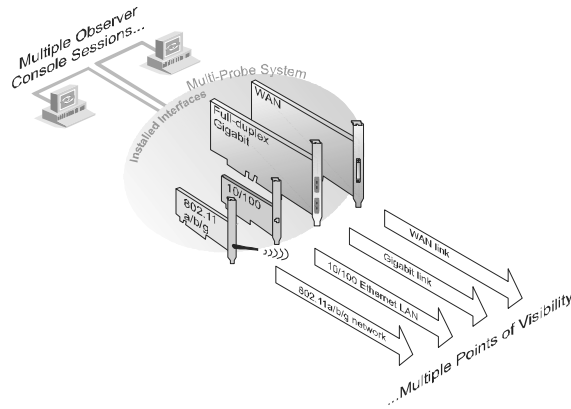
Advanced Probes—Observer’s proprietary probe. Depending on the license you purchased, the Advanced Probe will be a Single Probe, Multi-Probe, or Expert Probe. Advanced Single Probes are appropriate for sites with smaller administrative staffs, as this type of Advanced Probe communicates with only one Observer console at a time.

If you have a need for multiple administrators to view packet decodes and analysis from the same Probe simultaneously, choose an Advanced Multi-Probe. The Multi-Probe feature allows you to run multiple instances of the Probe on a single PC, which means that:

- You can install multiple network interface cards in a single Multi-Probe system, allowing multiple “points of visibility” from that Probe.

- You can connect multiple sessions of Observer to the same PC Multi-Probe system. Any Observer console session can see all of the networks available on the Multi-Probe. The sessions (called Probe instances) are securely encrypted and password-protected.

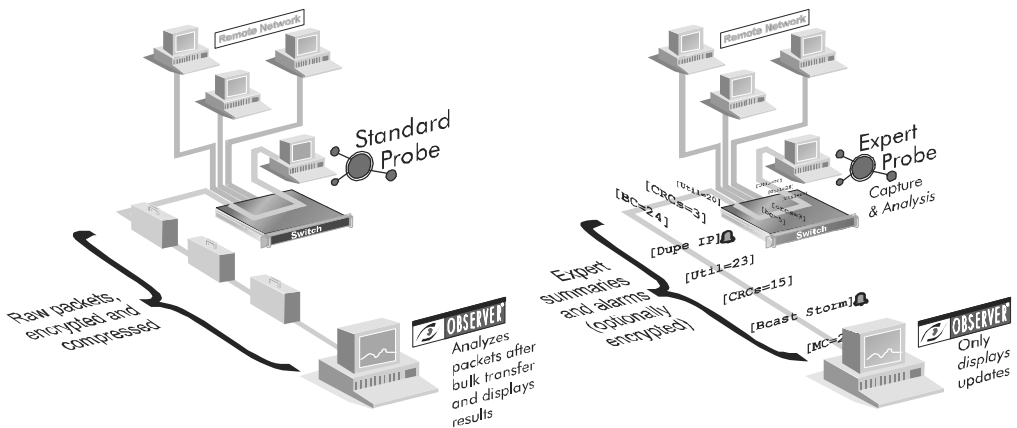
Advanced MultiProbe Features



Although Multi-Probes behave similarly to Single-Probes, the Multi-Probe capability means there is a different interface for configuring a Multi-Probe. See “Configuring Multi-Probe Connections” on page 17.

For the highest level of distributed performance, choose an Expert Probe. An Expert Probe is a specially-licensed Advanced Multi-Probe with local Expert Analysis and capture/decode capabilities built in. This saves bandwidth when performing remote expert analysis, and also allows remote decode views and expert analysis in real time.


Standard Advanced Probe vs. Expert Probe



The Expert Probe can be configured to run as an Observer console when you need to perform troubleshooting from where the Probe is located. See “Using the Expert Probe” on page 27.

All Advanced Probes support a superset of RMON2 functionality. Advanced Probes offer the greatest flexibility, performance and the most information for troubleshooting your LAN.

While Network Instruments’ Advanced Probes are more powerful and more efficient than any RMON2 Probes can be, Advanced Probes will work *only* with the Network Instruments’ Observer console. Although they transmit any and all information that an RMON2 Probe can (as well as kinds of information that the RMON1 and RMON2 specifications do not include) the Advanced Probe sends the information in a slightly different format.


 **RMON2 Probe**—Network Instruments’ RMON2 Probe responds to RMON1 or RMON2 requests with fully-compliant protocol structures. To use Network Instruments’ RMON2 Probe, you will need to either have a third-party RMON1 or RMON2 display console that can query the probe, or purchase the Network Instruments’ RMON2 Management Console for Observer.


In most cases, if you’re using Observer, you won’t need the RMON2 Management Console or an RMON2 probe, unless you’re using the RMON capabilities of the Observer Suite to monitor an embedded RMON Probe. If you are using the Observer Suite, we recommend using the Advanced Probe, as opposed to the RMON2 Probe—unless you have a specific reason to use RMON. When it’s working with Observer or the Observer Suite, the Advanced Probe can do everything an RMON2 Probe can do, and more.

The usual reasons to use RMON are either that your company has standardized on RMON, or that you need to monitor the probe with a program other than Observer. For most users of Observer, the Advanced Probe is a better choice.

During installation, you are prompted for the type of Probe you wish to install. Different executable programs are installed depending on your choice.

If you install one type of Probe and later on find you need the other, just run the installation program again and install the type of probe you need. You may, but *don’t* have to, uninstall the first kind of probe: an Advanced Probe will run side-by-side with an RMON2 Probe with no conflict. On the other hand, both probes will use system resources, which may over-tax the probe system. If you don’t need to run both kinds, it’s best to only run the one you need.

 **HCRMON Probe**—HCRMON is a proposed extension to the RMON2 standard that adds support for capturing frames and statistical data from High Capacity networks such as gigabit Ethernet. When you purchase one of Network Instruments’ hardware probes or systems, the RMON Probe included is HCRMON-compliant, meaning you have a built-in open standards alternative to our Gigabit Ethernet-optimized Advanced Probe that is also included. In addition, you can use the HCRMON Probe Software by connecting the system to the SPAN (or mirror) port of any gigabit switch that includes this functionality.

 **WAN RMON Probe**—WAN RMON is an enterprise extension to the RMON2 standard that adds support for capturing frames and statistical data from WAN links. When you purchase one of Network

Probe Capability Matrix

Observer Family Product	Probe Capability	Advanced Single-Probe	RMON1/2 Probe	Advanced MultiProbe	Advanced Expert Probe	HCRMON	WAN RMON
Software-only Products							
Advanced Single Probe		X	X				
Advanced Multi-Probe			X	X		X ¹	
Expert Probe				X	X	X ¹	
Observer Internal Probe		X					
Expert Observer Internal Probe				X	X		
Observer Suite Internal Probe				X	X		
Hardware/Software Bundled Systems							
Gigabit Probe Kit				X	X	X	
Gigabit Rack Mount Probe				X	X	X	
Gigabit Observer Suite System (Internal Probe)							
WAN Probe Kit				X	X		X
WAN Rack Mount Probe				X	X		X
WAN Observer Suite System (Internal Probe)							
GigaTrunk Probe GigaTrunk Edition Observer Probe			X			X ²	
GigaStor Probe GigaStor Edition Observer Probe			X			X	

1. 1000Mb maximum capture rate.

2. Does not support trunk-aware operation. See the *GigaTrunk Probe Installation and Quick Start Guide* for details.

Instruments' WAN Hardware Probes or systems, the RMON Probe included is WAN RMON-compliant, meaning you have a built-in open standards alternative to our WAN-optimized Advanced Probe that is also included.

Hardware Probes

For high volume gigabit and WAN analysis, Network Instruments offers a number of hardware solutions:

- Gigabit Rack Mount Probes and Probe Kits
- WAN Rack Mount Probes and Probe Kits
- 1U 10/100/1000 Probe Appliance
- 100MB Full-Duplex Probe Appliance

All hardware Probe configurations include at least the Advanced Probe as well as RMON1 and RMON2 compliant probes. In addition, Gigabit Probes and Probe Kits also include an HCRMON-compliant probe for open-standards High Capacity applications. Gigabit Observer GOSS and WOSS systems include Observer Suite, the do-everything capture, analysis, and monitoring solution for enterprise network professionals. See the "Probe Capability Matrix" on page 5 for a table that shows what probe capabilities are included with various Network Instruments Observer-family products.

Hardware/Software Requirements and Installation

Operating Systems Supported

Windows 2000 and Windows XP/2003.

For the latest information on Probe system hardware and software requirements, see:

www.networkinstruments.com/products/system_req_10.html

The Network Instruments Probes *may* run on systems with less processing power and RAM than those specified, but Network Instruments does not support configurations with less than the minimum stated requirements.

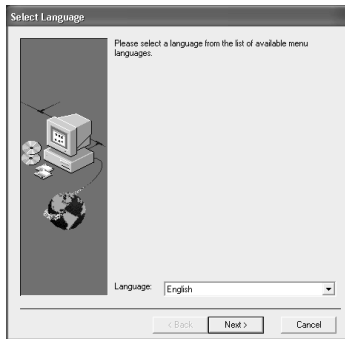
In the short term the system may seem to operate with less than the minimum specified hardware (e.g., a slower processor). Be aware that even if a subminimum installation works momentarily, a later, heavier load on the system can cause it to fail. Network Instruments sells hardware Probes that are guaranteed to keep up with heavy loads. See www.networkinstruments.com for details.

Installation

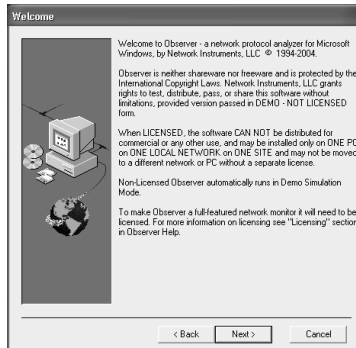
To install the Observer Probe, follow these steps:

1. Insert the Probe CD in your CD Drive (if you copied the installation files from our web site, simply double click the downloaded icon to start the installation).

- When the setup program runs, follow the instructions on your screen.
- Select your installation language from the dropdown and click on the NEXT button.



- The Welcome dialog will be displayed. License terms will be displayed; click on the NEXT button to agree to the terms.



- Setup will ask if you want to install Observer, Advanced Probe, Expert Probe, or RMON Probe. Select the type of Probe you wish to install and click on the NEXT button.



6. Setup will ask where to copy the Probe files.

Unless you have a specific reason to install the probe elsewhere, we suggest that you accept the default destination.



7. Setup will copy the Probe files to your PC.
8. Once the files have been installed on your PC and the network configuration has been modified, Windows 2000/XP/2003 will ask permission to restart. Once you reboot or restart Windows, you will be able to run the Probe by double-clicking on its icon in the “Network Instruments’ Advanced Probe” group.

You won’t be able to run the probe until you restart your computer.

Ethernet Errors By Station & NIC Driver Installation

To view and process Ethernet station errors, it is necessary that the network adapter card has been modified to pass error packets to the probe software (either the built-in probe that is part of Observer or a separate Advanced Probe), and then to the Observer console.

Normally NDIS drivers only keep track of the number of error packets seen on a LAN. The NDIS driver does not process or pass the error packet in any way. Without some way of passing error packets up to the operating system or application, there is no way for the operating system or application to obtain information about the source and nature of the errors.

Network Instruments has worked with a number of card manufacturers to modify the standard network card NDIS driver so that it will both maintain error counts and pass error packets up to Observer for processing. Observer ships with a number of these ErrorTrak™ drivers. They are located in the Drivers directory on the distribution media and are installed to the [usually C:] \Observer Files\Drivers directory during the installation process.

The Network Instruments’ ErrorTrak™ drivers are modified standard drivers and work just as the standard drivers do, with the one addition that error packets are passed to Observer.

Installing ErrorTrak™ Drivers under Windows 2000/XP/2003

1. Select Start > Settings > Control Panel > System > Hardware > Device Manager.
2. From the Device Manager tree, open Network adapters and double-click on the entry for your adapter card.
3. Choose the Driver Tab and click the UPDATE DRIVER... button.
4. This will start the **Update Device Driver** wizard. Select the SEARCH FOR A SUITABLE DRIVER FOR MY DEVICE radio button and click NEXT.
5. From the next dialog, check the SPECIFY A LOCATION button. Click NEXT.
6. From the next dialog, browse to the C:\Observer Files \Drivers\CARD_TYPE\Win2000 directory (where “CARD_TYPE” is the chipset that you are using—e.g., Intel1000Pro for NIC cards using the Intel 1000 Pro chipset).
7. Select the NET2000.INF file and click NEXT.
8. Windows 2000/XP/2003 will update the driver.

Please check the Network Instruments' Web site listed below for more information on supported network adapter cards:

For ISA and PCI Adapters

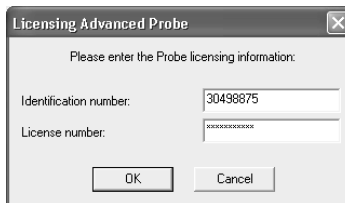
<http://www.networkinstruments.com/html/osup1001.html>

For PCMCIA Adapters

<http://www.networkinstruments.com/html/osup1002.html>

Licensing the Probe

Probes (whether RMON or Advanced) must be licensed before use. If your probe is not licensed, the licensing dialog will be displayed the first time the program is run.



Probes require two unique numbers for licensing: an identification number and a license number. Enter both numbers in this dialog to license the probe (you can find these numbers on the licensing card that

came in your Probe Pack).

The image shows a registration card for Network Instruments' software. The top section is titled "PROBE REGISTRATION" and includes instructions for customers to complete and return the top portion. It contains a form with fields for Name, Company, Address, City, State, Zip/Postal Code, Country, Telephone, Email, Place of Purchase, and Date of Purchase. Below this is a section titled "DO NOT DISCARD!" which contains the "OBSERVER® Probe License" information. This section includes the company name "Network Instruments, LLC", address "8800 West Highway Seven, Fourth Floor, Minneapolis, MN 55426 USA", phone "952.932.9899", fax "952.932.9845", and email "info@networkinstruments.com". It also features logos for "OBSERVER PROBE", "SECURITY PROBE", "SOFTWARE PROBE", and "NOISE PROBE". A large, faint watermark "DO NOT DISCARD" is visible across the bottom half of the card. At the bottom, the website "www.networkinstruments.com" is listed.

Identification and license numbers will be shown in this box.

If you downloaded the software from Network Instruments' Internet site and then purchased a license by phone, Network Instruments will provide with the license codes via fax.

Be sure to write both of these numbers down and keep them in a safe place; if you have to reinstall the software, you may need them again.

Dongles

Your Probe Pack may have included a small plug device (either parallel or USB), called a dongle.



Parallel dongle



USB dongle

If your Probe Pack doesn't contain a dongle, it may not be needed in your environment. Install the probe; if you do need a dongle, the probe program will tell you the first time you try to run it.

If so, the dongle must be plugged into the parallel (or USB) port of your computer for the probe to function. Plug your parallel printer cable, if any, into the other end of the dongle.

If you have Network Instruments' Observer or Observer Suite, it may have come with a blue-colored dongle. You can plug one dongle into the other—in either order—and your parallel printer cable into the end of the second dongle.

If you run a probe without the dongle installed and if the dongle is required in your environment, the program will inform you that it's needed. If you do not have but need a dongle, please contact your Network Instruments' sales representative.

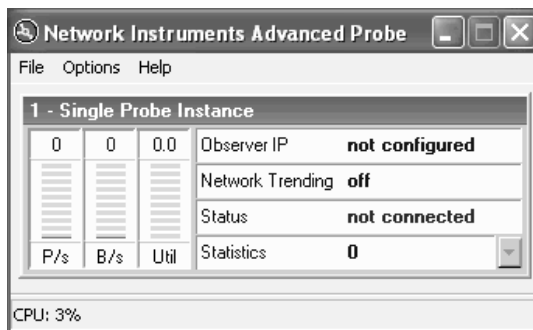
Using the Advanced Single-Probe

The default options for the Advanced Probe work in most cases, and once configured, are “set it and forget it.”

Once you have verified that the probe's PC is communicating with the Observer console (or another management program, if you're not using Observer or Observer Suite), you can start a probe by double-clicking on the Probe icon from the Probe or Observer group.

For information on installing the probe software, see the Installation instructions at the beginning of this section.

Advanced Single-Probe Window



The **Probe Activity** window displays summary statistics of data taken from the network (or networks, in the case of Multi-Probe deployments).

The main probe window has the following menu choices:

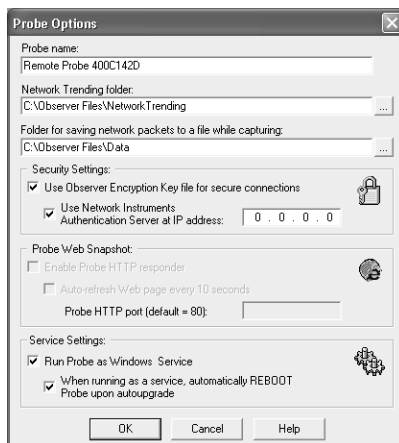
File

- License Probe—displays the dialog where you would enter or re-enter your probe activation numbers: the identification number and the license number.
- Exit—exits the probe program.

Options

Probe Options

Displays the Probe Options dialog:



- “Probe name” textbox—allows you to specify a name for the Probe which will appear in the Observer console Probe list.

Network Trending Folder:

- “Network trending folder” textbox—allows you to specify where the Network Trending data will be saved. Defaults to C:\Observer Files\NetworkTrending.

Unless you have a specific reason to use another directory, we suggest using the default.

Security Settings:

- “Use Observer Encryption Key file for secure Connections” checkbox—If checked, the Probe will not connect with a console unless a matching Observer Encryption Key (.oek) file is present in the in both the console and probe installation directories. Refer to the **EncryptUtil.chm** help file for details on running the utility to generate the file.
- “Use Network Instruments Authentication Server at IP Address”—Choose this option if you have licensed and installed Network Instruments Authentication Server. The Authentication Server centralizes user/password administration, superceding the Probe Security tab settings. Matching Observer Encryption Key files are required in the installation directory of both console and probe for the connection to complete.

Probe Web Snapshot:

- “Enable Probe HTTP responder” checkbox—allows you to enable the Web probe snapshot.

The probe offers a “snapshot” of general data by connecting to the probe using a standard Web browser. Information includes the state of the probe, the number of modes running, and the general network statistics that the probe displays within its own interface.

The probe will send more extensive and specific statistics to the console management program when asked to.

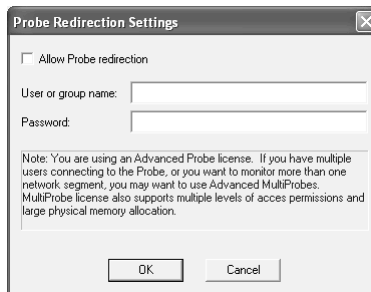
- “Auto-refresh Web page every 10 seconds” checkbox—allows you to configure the probe to refresh the browser display every 10 seconds.
- Probe HTTP port (default = 80) textbox—allows you to specify the port to use for reporting of Web information (by default, the probe uses port 80). This port can be changed in the Probe’s Options > Probe Settings dialog. To use the specified port, point your browser to the system running the probe, with the alternate port number specified at the end of the address. For example, if your probe is configured to report on port 79, and the address is 200.100.1.1, the http address would be: `http://200.100.1.1:79`

Miscellaneous:

- “Run Probe as Windows Service” checkbox— This ensures that the Probe is started every time the system reboots, which can be especially important for a remote Probe. The change will take effect on the next system restart.
- “Allow to automatically REBOOT system to complete the probe autoupgrade (will reboot only if probe runs as a service)” checkbox—allows you to elect to have the system reboot automatically after the probe is installed.

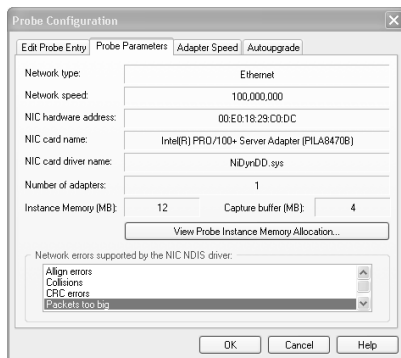
Probe Redirection Settings (Single-Probe Licenses only)

Displays the Probe Redirection Settings dialog, which lets you enable Probe redirection for a user or group of users who have the ID and password:



Probe Capabilities (Single-Probe Licenses only)

Displays the local probe NIC card and probe capture buffer information. This is the same dialog that is available from Observer's **Options -> Probe Options -> Probe Parameters** tab.



Redirecting a Single-Probe from the Observer console

Advanced Single-Probes can be redirected from one Observer console to another. This is done from the Observer console by selecting **Actions -> Redirect Probe**. Note that probe redirection can be password protected or disabled. Please see the previous section entitled “Probe Settings” for redirection options.

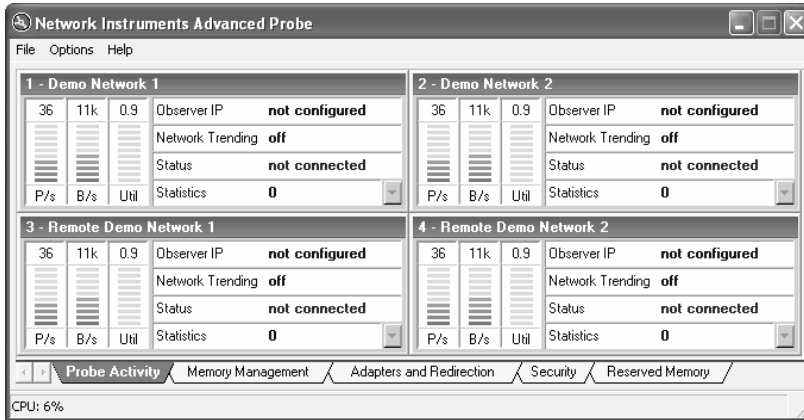
Using the Advanced Multi-Probe

Once you have verified that the probe's PC is communicating with the Observer console (or another management program, if you're not using Observer or Observer Suite), you can start a probe by double-clicking on the Probe icon from the Probe or Observer group.

For information on installing the probe software, see the Installation instructions at the beginning of this section.

The Advanced Multi-Probe main window has a number of tabs that allow you to control Probe network connections and memory usage, administer Probe security, and monitor Probe activity.

Advanced Multi-Probe Window



The Advanced Multi-Probe has five tabs:

- The **Probe Activity** tab, which shows summary statistics of data taken from the network (or networks, in the case of Multi-Probe deployments).
- The **Adapters and Redirection** tab, which allows you to create Multi-Probe instances and associate each with an adapter installed on the local PC. If you do not have a Multi-Probe license, this tab will be disabled.
- The **Security** tab, which allows you to create user accounts and set passwords for access to the Probe. This will prevent unauthorized Observer consoles from having access to corporate data if such security is a concern in your organization.
- The **Memory Management** tab, which lets you configure how much memory each probe or probe instance should use for various functions.
- The **Reserved Memory** tab, which lets you set how much memory to set aside for exclusive use by the Probe.

The main probe window has the following menu choices:

File

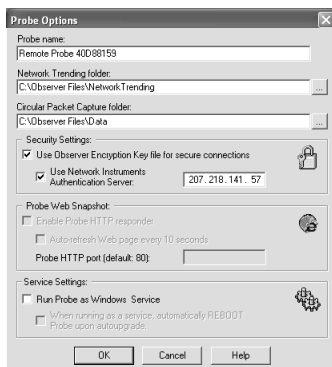
- License Probe—displays the dialog where you would enter or re-enter your probe activation numbers: the identification number and the license number.
- Exit—exits the probe program.

Options

There is one option on the menu:

- Probe Options (which is described below).

The Probe Options dialog lets you name the Probe and set various run-time, storage and security options:



- “Probe name” textbox—allows you to specify a name for the Probe which will appear in the Observer console Probe list.
- “Network trending folder” textbox—allows you to specify where the Network Trending data will be saved. Defaults to C:\Observer Files\NetworkTrending.

Unless you have a specific reason to use another directory, we suggest using the default.

- “Circular Packet Capture Folder” textbox—allows you to specify where automatically saved packet capture data will be stored. Automatic saving of data is configured in Observer’s Packet Capture Settings dialog. Defaults to C:\Observer Files\Data. When data is over

Unless you have a specific reason to use another directory, we suggest using the default.

- “Use Observer encryption key file for secure connections” checkbox—Negotiates triple-DES encryption with Consoles that have the same encryption key file installed locally. Refer the Observer Reference Guide and the readme file for ObsEncKey.Exe include with the Probe for details. Triple-DES is an extension of the original 56-bit key Data Encryption Standard approved by the National Security Agency. By making 3 DES encryption passes, it increases the effective key length to 168 bits. Only use this option if you need strong encryption, because it imposes a significant performance cost. Even with this option turned off, the Probe will not send raw, easily-readable data; it will be concealed by the proprietary compression algorithm.
- “Use Network Instruments Authentication Server” The Authentication server is a separately-licensed Network Instruments product that allows centralized security management of Observer Console/Probe. Unless you have purchased this product, leave this option unchecked. If you have purchased the product, you must supply the IP address of the system running the Authentication Server software. Refer to the documentation and help file distributed with the NI Authentication Server for details on its configuration and operation.
- “Enable Probe HTTP responder” checkbox—allows you to enable the Web probe snapshot.

The probe offers a “snapshot” of general data by connecting to the probe using a standard Web browser. Information includes the state of the probe, the number of modes running, and the general network statistics that the probe displays within its own interface.

The Probe will send more extensive and specific statistics to the console management program when asked to.

- “Auto-refresh Web page every 10 seconds” checkbox—allows you to configure the probe to refresh the browser display every 10 seconds.
- Probe HTTP port (default = 80) textbox—allows you to specify the port to use for reporting of Web information (by default, the probe uses port 80). To use the specified port, point your browser to the system running the probe, with the alternate port number specified at the end of the address. For example, if your probe is configured to report on port 79, and the address is 200.100.1.1, the http address would be: `http://200.100.1.1:79`.
- “Run Probe as Windows Service” checkbox— This ensures that the Probe is started every time the system reboots, which can be especially important for a remote Probe. The change will take effect on the next system restart.

Configuring Multi-Probe Connections

If you have a Multi-Probe license, you can:

- configure the Probe to view multiple networks if multiple NICs are installed on the local PC
- configure the Probe to provide multiple Observer consoles with views of the local network interfaces

About Probe Instances

The Probe accomplishes these capabilities by allowing multiple *instances* of itself. A Probe instance is a “virtual” Probe with attributes that define:

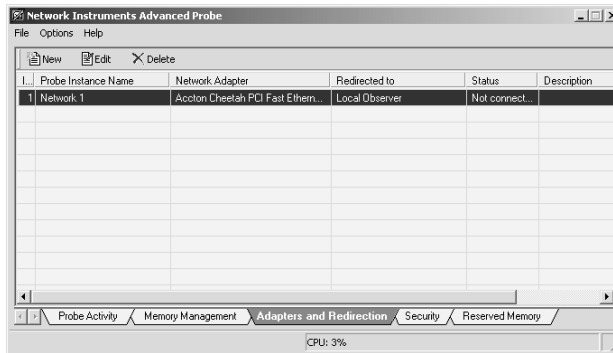
- which network interface on the local PC to capture data from
- which Observer console (local or remote) to direct the data to

You must have at least 12MB Reserved Memory *available* to add a Probe instance. See “Setting the Total System Memory reserved for Probes” on page 25 for details on allocating memory for Observer Probes.

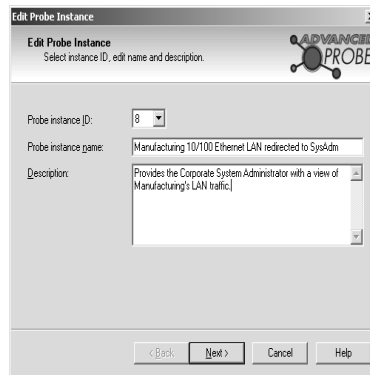
Creating a Probe Instance

To set up a Probe Instance, follow these steps:

1. Click the **Adapters and Redirection** tab to display the current list of instances:



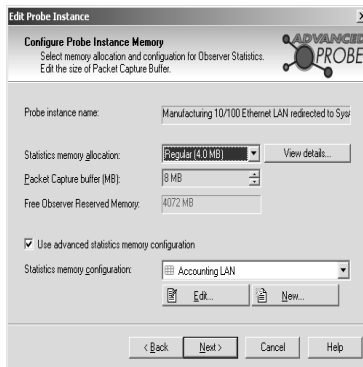
2. Click **New Instance...** to begin the Instance wizard, which steps you through naming and setup of the new instance:



If the **New Instance...** button is grayed out, it probably means you don't have enough Observer-allocated memory to add another instance. You must have a minimum of 256MB RAM to run the Probe with a single instance, plus 12MB for each additional Probe instance. See "Setting the Total System Memory reserved for Probes" on page 25 for details on allocating memory for Observer Probes.

3. Select an instance ID, then name and describe the instance you are creating. Click **Next...** when you are finished.

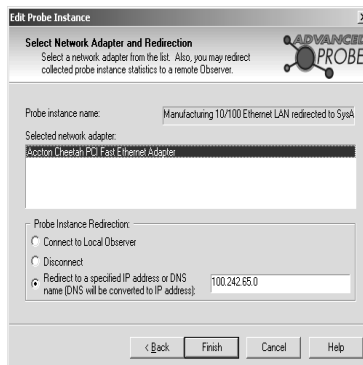
The Memory Configuration dialog is displayed:



4. Select an appropriate Capture Buffer size given the local system's available memory and how much traffic you plan on capturing from the given network.

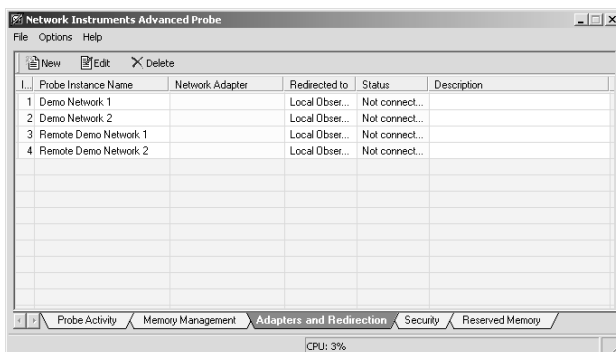
Statistical reporting uses different memory and much less of it. Although it is possible to customize the amounts of memory used by Observer's various statistical displays (by checking the **Used Advanced Statistics Memory Configuration** option), for most situations the defaults will work perfectly well.

Click Next to continue, and the adapter/redirection configuration dialog is displayed



5. Choose an adapter to associate with this instance, and a destination for the Probe to direct its analysis data. "Local Observer" means the Observer console through which the Probe is being configured; when configuring a stand-alone Probe this option will be grayed out. Click **Finish** when you are done.

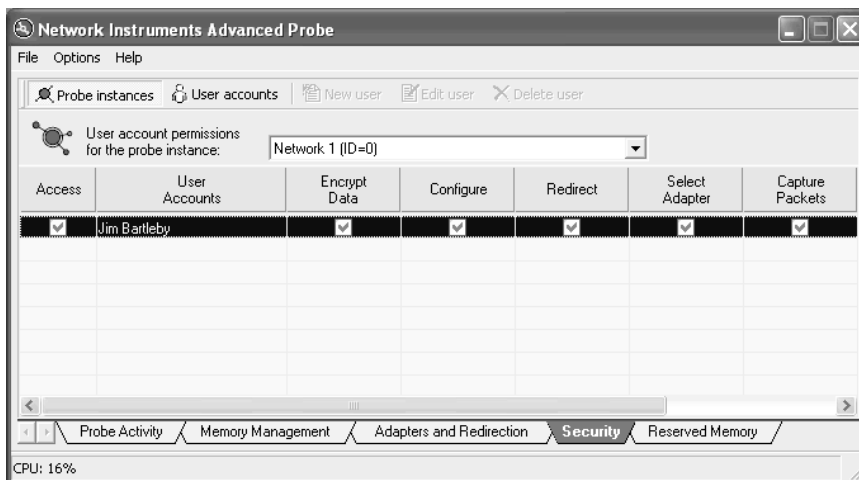
The Probe Adapters and Redirection tab will now list the new Probe instance:



Configuring User Accounts for Secure Access

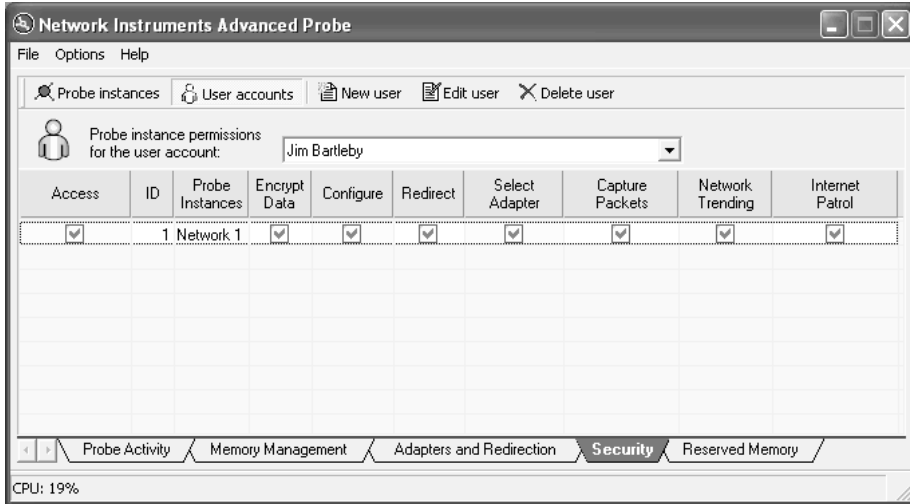
Note that if you have purchased and configured the Network Instruments Authentication Server option, user authentication is configured via the Authentication Server rather than the Probe Security tab, which will be grayed out if the “Use Network Instruments Authentication Server” option is checked in the **Options->Probe Options** dialog.

If you wish to restrict access to packet captures and reporting provided by a Probe instance, you can define security attributes of the Probe by clicking the **Security** tab:



The example above shows the Security tab as it appears when the Probe Instances button in the upper left corner of the display is selected. This view lets you select a Probe instance from the dropdown list box and display users that have access to that instance and their permissions.

To display security information by user account, press the User Account button to the left of the Probe Instances button. This lets you see what permissions the currently selected user has access to on each instance of the Probe:



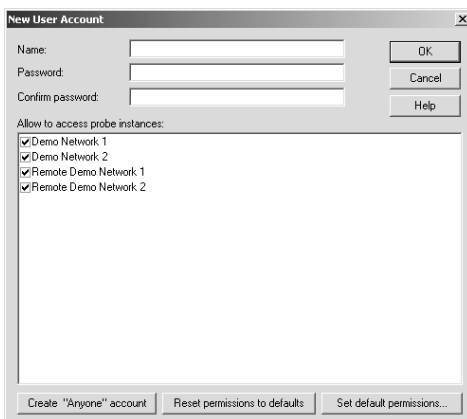
When displaying a user account’s permissions as above, you can use the checkboxes to fine-tune the permissions that user has on each account by clicking on the Permissions checkboxes to select or deselect the particular option. The different types of permission are described below:

Permission	Explanation
Encrypt data	Data sent to the console will be triple-DES encrypted during transmission. Triple-DES is an extension of the original 56-bit key Data Encryption Standard approved by the National Security Agency. By making 3 DES encryption passes, it increases the effective key length to 168 bits. Only use this option if you need strong encryption, because it imposes a significant performance cost. Even with this option turned off, the Probe will not send raw, easily-readable data; it will be concealed by the proprietary compression algorithm.
Configure	User is allowed to change the Probe’s configuration options (such as memory usage, etc.).
Redirect	User is allowed to change the destination console for Probe analysis data.
Select Adapter	User is allowed to change the adapter setting for the Probe.
Capture Packets	User is allowed to view captured packets from the Probe’s network.
Network Trending	User is allowed to view Network Trending data from the Probe’s network.
Internet Patrol	User is allowed to run Internet Patrol on the Probe’s network.

Creating or Editing a User Account

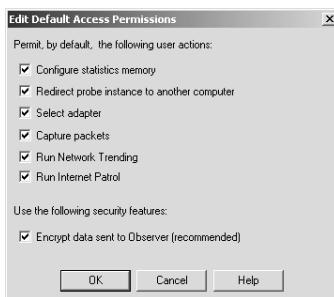
To create a new account click New User Account; to edit an existing account, select the account and click Edit User Account. These options are also available on the right-click menu.

The setup options are the same whether you are creating a new account or editing an existing account:



Fill out the name and password fields and select the instances you want this account to have access to.

By default, when you give an account access to an instance, that account will have permission to do everything it is possible to do with a Probe instance: receive all statistics and capture packets, redirect it, configure its memory, etc. If you want to change the default permissions for the user you are creating or editing, click **Change Default Permissions...**, which displays the **Set Default Permissions** dialog:



Check the desired options and click **OK**. When you grant this account access to another Probe instance, the permissions will be automatically set to match what you have selected here. You also will be able to reset this user's permission to these values on any Probe instance by right-clicking the account or instance and choosing the **Reset User Account Permissions** option from the popup menu.

Customizing Statistics and Capture Buffers For Probe Instances

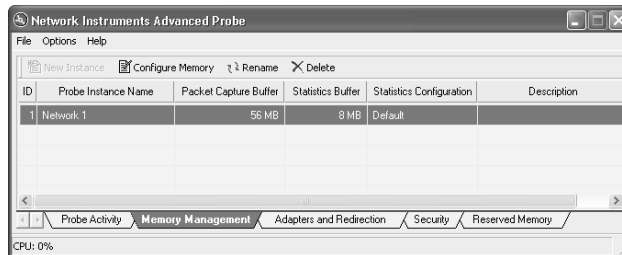
There are two kinds of buffers that a Probe uses to store data in real-time: Capture buffers and statistical buffers. The capture buffer is used to store the raw data captured from the network; the statistical buffers store data entries which are series of snapshots of a given statistical datapoint.

Selecting an appropriate capture buffer size given system resources is all most users need to worry about; the default settings for the statistical buffers work perfectly fine in the vast majority of circumstances.

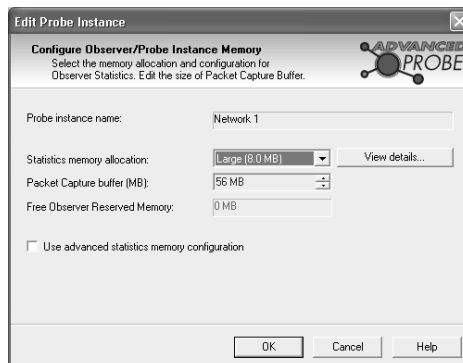
However, if you are pushing the limits of the PC system on which the Probe is installed by creating many instances, you may be able to avoid some performance problems by fine-tuning the memory allocation for each instance.

For example, suppose you want to give a number of remote administrators access to Top Talkers data from a given Probe. You will be able to add more instances within a given system's memory constraints if you set up the statistics buffers to only allocate memory for tracking Top Talkers and to not allocate memory for statistics that no one will be looking at.

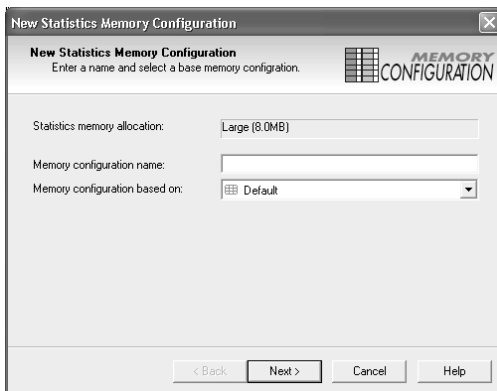
To view and manage memory allocation for Probe instances, click the Memory Management tab to display the list of instances and their buffer sizes:



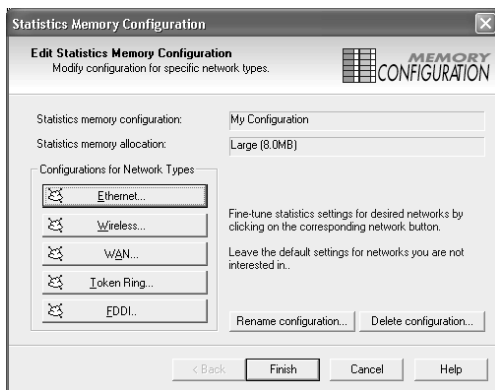
Right click any instance and select **Edit Probe Instance...** to access the memory allocation dialog:



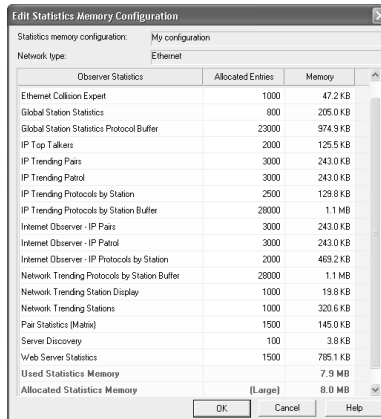
This dialog lets you select the Capture buffer size, as well as letting you pick from a number of statistics memory “presets” (Regular, Large, and Extra Large). If you want finer control over the statistics memory allocation, check the **Use Advance Statistics Memory Configuration** option, which lets you select from a number of statistics memory presets that you can define and edit yourself. Clicking **New...** or **Edit...** displays the setup dialog:



Enter a descriptive name for the custom memory configuration and select a previous configuration as a model for the new configuration if desired. Click **Next>** to display the second setup dialog:



By clicking on one of the Network Types buttons, you can view and change the number of entries allocated for each statistical type:

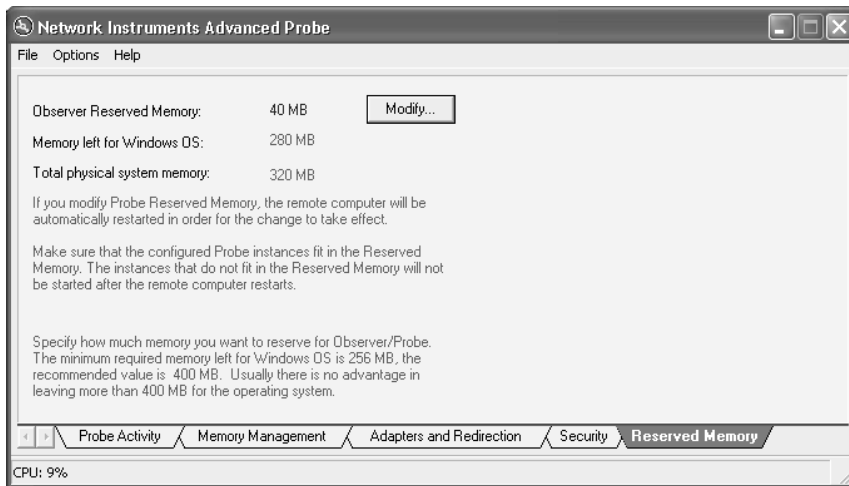


An *entry* is a record of the given statistic; for example, a Top Talker entry consists of a station, for errors, an entry would consist of error listing. When you constrain a report to n number of entries, the Probe will only report the last n entries to the Observer console; entries after the n th entry are never reported or displayed on the Observer console. Observer informs you when the Probe is exceeding its memory buffer for a particular statistic by displaying a message.

Setting the Total System Memory reserved for Probes

Because Observer operates in real-time, its buffers must always remain in RAM; if the buffers resided in standard Windows user memory, nothing would prevent the buffer file from being swapped out to disk and subsequent packet loss. For this reason, the Probe reserves its memory from Windows upon startup so that no other applications can use it and cause the buffer to be swapped out to disk.

Although the default amount of total reserved memory should work perfectly in most situations, you can change it. Click the Observer Reserved Memory tab to display how much memory is reserved for Probe operation and how much memory is left for Windows:



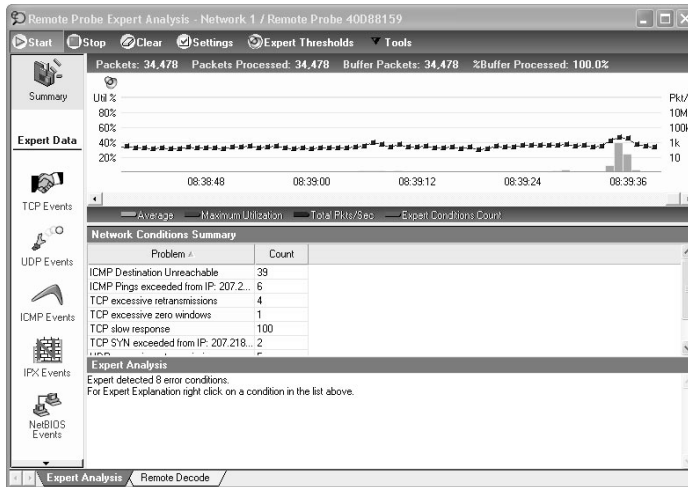
The setup screen will not allow you to reserve memory in excess of what Windows needs to run, but it will allow you to leave less than the optimum amount necessary for Windows to perform at its best. Proceed with caution; any performance benefits you might gain by increasing Observer's allotment can be lost if you do not leave enough memory for Windows to perform well.

Using the Expert Probe

If you have purchased an Expert Probe, it has all the capabilities of the Advanced Multi-Probe, plus a distributed Expert that gives you unparalleled power and flexibility in using the Probe both remotely and on site.

Connecting to an Expert Probe from a Remote Observer console

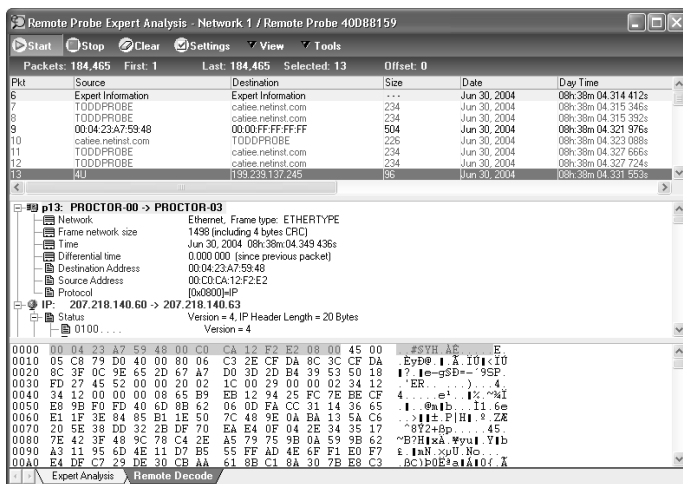
Once the Expert Probe is running, you can connect to it (or rather, the instances that have been defined for it) from any Expert Observer or Observer Suite system. You will now be able to run all statistical displays just as with other probes. You will also be able to capture packets remotely and perform analysis locally using the same mechanisms as with standard Probes. The only difference is an additional menu option, **Remote Probe Expert Analysis**, available on the **Trending/Analysis** menu. The resulting window includes a tab for viewing remote analysis, and another for real-time viewing of decoded packets as they are being captured remotely:



The Remote Expert has exactly the same look, feel, and functionality of a local Expert Observer. The advantage is Expert Probes perform analysis locally, allowing smart updates to remote Expert Observer and Observer Suite consoles in real-time while minimizing network load.

The Remote Decode provides an efficient mechanism for viewing decode buffers remotely. Again, the look and feel are identical to that of a local decode display, but an Expert Probe transfers decode data

only when you select the packet from the one-line summary pane, which is updated with packet header information in real time.



Packet header display is updated in real time...

...but packet data is only transferred when you highlight the packet's header.

Switching Between Probe and Console Interfaces

To temporarily change the Expert Probe interface to load as a fully-featured Observer console, choose **Options->Switch Between Observer and Expert Probe Interface**. The change will take effect after you restart the program. For more detailed information about using the Observer console, refer to the *Observer Reference Manual* or online help.

Advanced Probe Port Usage With Observer

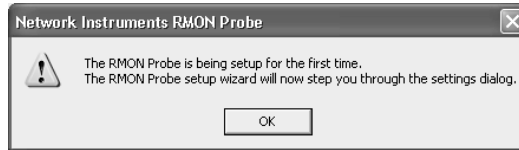
Advanced Probes and Observer use TCP port 25901 and 25903 to transfer data and commands between the probe and the Observer console. Knowledge of these port numbers may be required for users with a firewall between the probe and the Observer console.

The optional Network Instruments Authentication Server (which authenticates Probe/console connections via a central server) uses UDP port 25901.

Using the RMON Probe

RMON Probe Configuration

Immediately after an RMON Probe is licensed, the following screen is displayed:



Click **OK** to start the new RMON Probe wizard, which will walk you through eleven screens of configuring information for Network Instruments' RMON Probe.

Each of these screens can be accessed at any time the RMON Probe is running from Options > Probe Options.

The Network Instruments' Probe Pack can install either an Advanced Probe or an RMON2 Probe. When an RMON2 Probe is installed, a number of configuration parameters need to be set initially to begin using the probe.

The Network Instruments' RMON2 Probe supports all RMON2 groups (RFC2021 and RFC2074) and all RMON1 groups (RFC1757 and RFC1513). Complete listings of the group objects supported are listed at the end of this section.

As noted earlier in this section, if you are using Observer as your management console it is recommended that you use an Advanced Probe (the Advanced Probe offers a superset of the RMON2 functionality for troubleshooting and management). But if your company has standardized on RMON or you are using a third-party management console, Network Instruments' RMON2 Probe offers a complete implementation of RMON1 and RMON2.

If you're planning on using the Observer console as your management station and want to use an RMON2 Probe, you must upgrade the Observer console with the RMON2 Management Console to view RMON2 Probes (Network Instruments' or anyone else's) within Observer. Without the RMON2 Management Consoles, Observer doesn't speak RMON.

Generally speaking, however, if you're using the Observer console, you'll be better off using the Advanced Probe, not the RMON2 Probe. On the other hand, for example, if another system administrator on a UNIX workstation is using an RMON-compliant monitoring program, the Network Instruments' RMON2 Probe and Network Instruments' RMON2 Management Console to Observer would permit both of you to use the same probe.

RMON2 Probe Window

The RMON2 Probe interface offers the RMON Probe PC user a number of current statistics and other connection information as well as a list of available interfaces that can be managed.

The RMON statistics displayed on the RMON Probe PC are a subset of those reported to the RMON management console; the Network Instruments' RMON Probe reports *all* RMON1 and RMON2 statistics to the management console.

Statistics include:

- Last Manager IP—displays the IP address of the last management station to control the probe.
- Interfaces—displays the number of interfaces being monitored.
- P/s—displays the current packets per second.
- B/s—displays the current bits per second.
- Util—displays the current bandwidth utilization (as a percentage).
- CPU—displays the current local probe CPU PC utilization (as a percentage). This is the percentage of the CPU's capacity that is being utilized for all purposes, not just the RMON2 Probe. Under most circumstances, a Network Instruments' Advanced Probe or RMON2 Probe will use only a small percentage of the CPU's capacity.

Note: the above listed statistics are for the interface listed on the right side of the display. You may change the current interface by using the dropdown arrow on the right of the adapter listed.

The main RMON2 Probe window has the following menu choices:

File

- License Probe—displays the License dialog where you would enter or re-enter your probe license and identification numbers in order to activate the probe.
- Exit—exits the probe program.

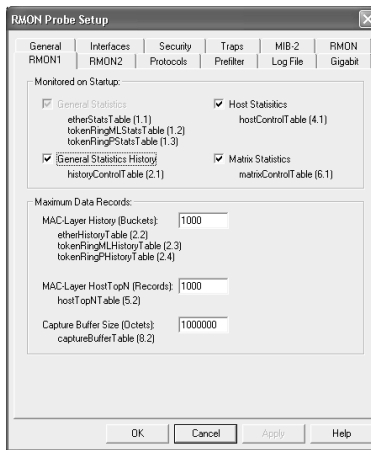
View

- View log—loads the RMON2 Probe log into Microsoft Notepad for viewing.

Options

- The Options menu contains two items: Probe Settings and Probe Statistics.
- The Probe Settings item brings up a dialog containing the following tabs:

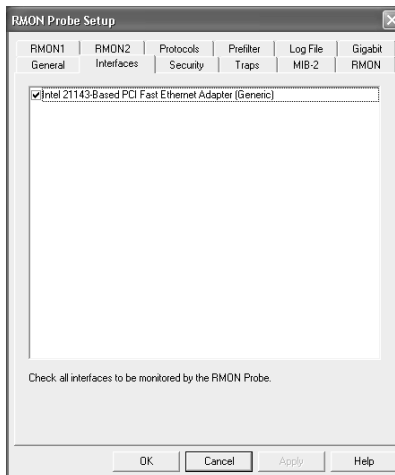
General Tab



General SNMP Settings:

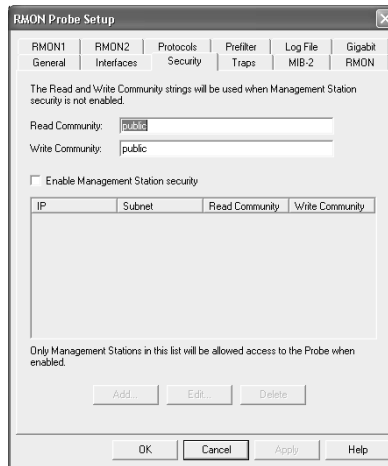
- “Trap IP Address” dropdown—enter the trap IP address to be used in the SNMP header. This is the address that identifies the system running the RMON Probe and is, by default, the IP address of the RMON Probe PC.
- “SNMP Port” textbox—enter the port to be used for SNMP conversations.

Interfaces Tab



- The Interfaces tab contains the available NICs which may be monitored by the RMON Probe and allows you to select the interface (or interfaces) to be monitored by the RMON Probe.

Security Tab



The Security tab contains the following items:

- “Read Community” textbox—allows you to enter the string for the “Read Community” name.

This is a read-only password for the RMON Probe. Use of this password will enable an RMON-compliant program to query the RMON Probe, but not make any changes in its configuration.
- “Write Community” textbox—allows you to enter the string for the “Write Community” name.

This is a read-write password for the RMON Probe. Use of this password will enable an RMON-compliant program to query the RMON Probe, *and* to make changes in its configuration through standard RMON commands.
- “Enable Management Station Security” checkbox—allows you to enable management station security, permitting you to choose which IP addresses and/or ranges of IP addresses will have access to the RMON Probe, as well as optionally setting individual read and write community names.

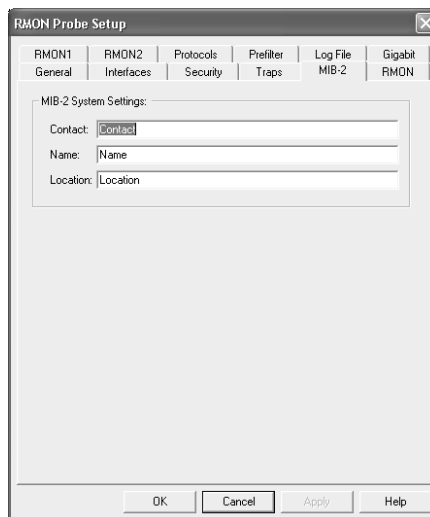
Traps Tab



- Click the ADD button to add the IP address of those stations to which the RMON Probe will send SNMP trap messages. If any stations are listed in the dialog, they can be modified by using the EDIT button or deleted with the DELETE button.

The SNMP Probe can send trap messages to 10 IP addresses.

MIB-2 Tab



The MIB-2 tab contains the following items:

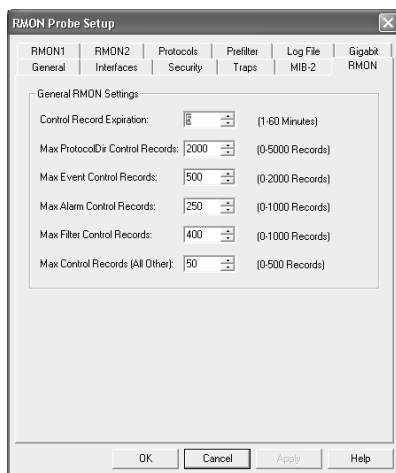
- “Contact” textbox—allows you to enter the designated contact for the RMON Probe.

This is usually the name of the network administrator who uses the RMON Probe, but it can contain any alphanumeric string, such as the phone or pager number of the designated contact.
- Location textbox—allows you to enter the location of the RMON Probe PC.

This is usually the name of the physical location of the RMON Probe PC, (e.g., “J. Phillips’ Main Computer, Room 221”), but it can contain any alphanumeric string.

This information will be displayed for the MIB-2 OIDs for contact and location.

RMON Tab

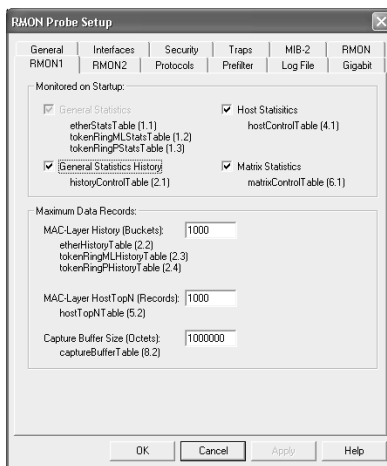


The RMON tab contains the following items:

- “Control Record Expiration” spinbox—allows you to set the number of minutes, from 1-60, before a control record expires and is removed.
- “Max ProtocolDir Control Records” spinbox—allows you to set the maximum number of Protocol Dir Control records, from 0-5000, that are to be maintained at a given time. When the number of records exceeds this number, the oldest records are purged in order to make room for newer information. If this number is set at 0, no records will be saved by the RMON Probe.
- “Max Event Control Records” spinbox—allows you to set the maximum number of Event Control records, from 0-2000, that are to be maintained at a given time. When the number of records exceeds this number, the oldest records are purged in order to make room for newer information. If this number is set at 0, no records will be saved by the RMON Probe.

- “Max Alarm Control Records” spinbox—allows you to set the maximum number of Alarm Control records, from 0-1000, that are to be maintained at a given time. When the number of records exceeds this number, the oldest records are purged in order to make room for newer information. If this number is set at 0, no records will be saved by the RMON Probe.
- “Max Filter Control Records” spinbox—allows you to set the maximum number of Filter Control records, from 0-2000, that are to be maintained at a given time. When the number of records exceeds this number, the oldest records are purged in order to make room for newer information. If this number is set at 0, no records will be saved by the RMON Probe.
- “Max Control Records (All Other)” spinbox—allows you to set the maximum number of Control records that are not Protocol Dir, Event, Alarm, or Filter Control records, from 0-500, that are to be maintained at a given time. When the number of records exceeds this number, the oldest records are purged in order to make room for newer information. If this number is set at 0, no records will be saved by the RMON Probe.

RMON1 Tab



Monitored on Startup:

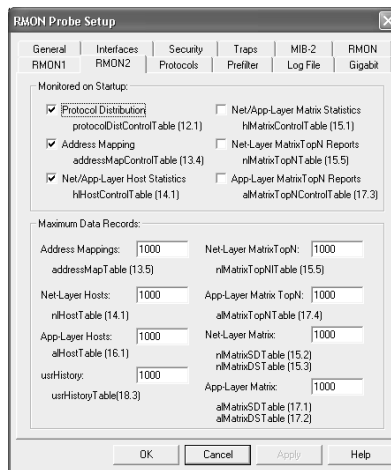
- “General Statistics” checkbox—etherStatsTable, tokenRingMLStatsTable, and tokenRingPStatsTable.
This box is checked and grayed out, as under the RMON1 specification, such groups are always monitored.
- “General Statistics History” checkbox—historyControlTable, a list of history control entries, group 2.1.
- “Host Statistics” checkbox—hostControlTable, a list of host control table entries, group 4.1.

- “Matrix Statistics” checkbox—matrixControlTable, a list of information entries for the traffic matrix on each interface, group 6.1.

The following items allow you to configure Maximum Data Records maintained for RMON1 groups:

- “MAC-Layer History (Buckets)” textbox—the number of buckets, or data collections, to be kept for: etherHistoryTable, a list of ethernet history entries (group 2.2); tokenRingMLHistoryTable, a list of Mac-layer token ring statistic entries, (group 2.3); and tokenRingPHistoryTable, a list of promiscuous Token Ring statistics entries, (group 2.4).
- “MAC-Layer HostTopN (Records)” textbox—the number of records to be kept in hostTopNTable, a list of top-N host entries, group 5.2.
- “Capture Buffer Size (Octets)” textbox—the size of the capture buffer, in octets, group 8.2.

RMON2 Tab



Monitored on Startup:

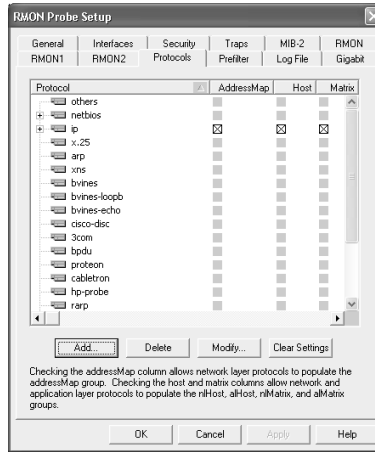
- “Protocol Distribution” checkbox—protocolDistControlTable, a table which controls the setup of protocol type distribution statistics, group 12.1.
- “Address Mapping” checkbox—addressMapControlTable, a table which controls the collection of network layer address to physical address to interface mappings, group 13.4.
- “Net/App-Layer Host Statistics” checkbox—hlHostControlTable, a list of higher layer post table control entries which enable the collection of the network in applicable level host tables index by network addresses, group 14.1.
- “Net/App-Layer Matrix Statistics” checkbox—hlMatrixControlTable, a list of higher layer matrix control entries, which enable the collection of the network and application level matrix tables containing conversation statistics indexed by pairs of network addresses, group 15.1.

- “Net-Layer MatrixTopN Reports” checkbox—nlMatrixTopNTable, a set of statistics for those network layer matrix entries that account for the highest number of octets or packets, group 15.5.
- “App-Layer MatrixTopN Reports” checkbox—alMatrixTopNControlTable, a set of parameters that control the creation of a report of the top N matrix entries according to a selected metric, group 17.3.

The RMON2 Tab’s Maximum Data Records box contains the following textboxes, from which you can set limits for the following data records:

- “Address Mappings” textbox—a table of network layer addressed to physical address to interface mappings, group 13.5.
- “Net-Layer Hosts” textbox—a list of higher layer host table control entries, which enabled the collection of the network and application level host tables indexed by network addresses, group 14.1
- “App-Layer Hosts” textbox—a collection of statistics for a particular protocol from a particular network address that has been discovered on an interface of a specific device, group 16.1.
- “usrHistory” textbox—usrHistoryTable, a list of user-defined history entries, group 18.3.
- “Net-Layer MatrixTopN” textbox—nlMatrixTopNITable, a set of statistics for those network layer matrix entries that account for the highest number of octets or packets, group 15.5.
- “App-Layer MatrixTopN” textbox—alMatrixTopNITable, a set of statistics for those application layer matrix entries that have accounted for the highest number of octets or packets, group 17.4.
- “Net-Layer Matrix” textbox—nlMatrixSDTable, group 15.2; and nlMatrixDSTable, group 15.3: two indexed lists of traffic matrix entries which collect statistics for conversation between two network-level addresses.
- “App-Layer Matrix” textbox—alMatrixSDTable, group 17.1; alMatrixDSTable, group 17.2; two indexed lists of traffic matrix entries which collect statistics for conversation between two network-level addresses.

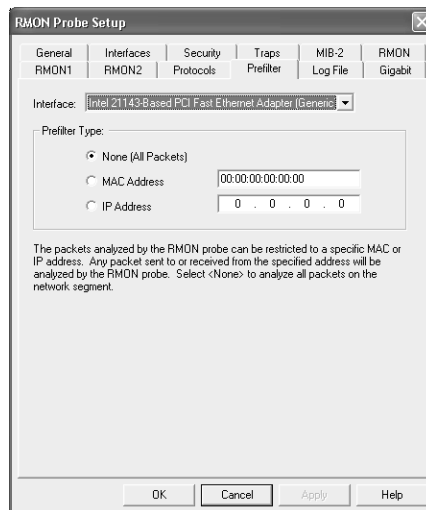
Protocols Tab



The Protocols tab contains a listing of the protocols that the RMON Probe will recognize. Checking the AddressMap column allows the check network layer protocols to populate the AddressMap group. Checking the Host and Matrix columns allow network and application layer protocols to populate the nlHost, alHost, nlMatrix, and alMatrix groups.

These are typically set by the RMON management console and only need to be set here if no RMON management console will be used.

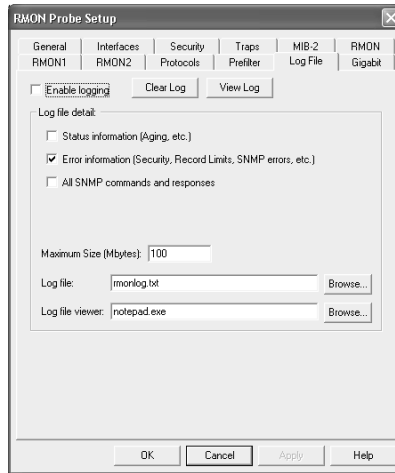
Prefilter Tab



The Prefilter dialog permits the user to restrict packet capture and analysis of the RMON Probe to a specific address. Click the MAC Address or IP Address radio button and enter the desired MAC Address or IP Address in the related edit box.

The RMON Probe is, by default, configured to “see” and analyze all packets on its local network. By selecting either the MAC address or IP address radio buttons and entering the desired address in the corresponding edit box, the RMON Probe can be configured to analyze only packets originating from and those explicitly addressed to (i.e., not including broadcast and multicast packets) the specified device.

Log File Tab



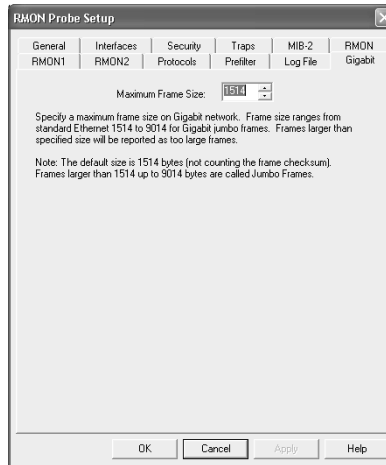
- “Enable logging” checkbox—allows you to enable logging of RMON Probe data.
- CLEAR LOG button—allows you to clear the log.
- VIEW LOG button—allows you to view the log.

Log file detail:

- “Status information (Aging, etc.)” checkbox—allows you to enable the logging of status information, such as aging.
- “Error information (Security, Record Limits, SNMP errors, etc.)” checkbox—allows you to enable the logging of error information, such as security errors, exceeding of record limits, or SNMP errors.
- “All SNMP commands and responses” checkbox—allows you to enable the logging of all SNMP commands and responses sent or received by the RMON Probe.
- “Maximum size (Mbytes)” textbox—allows you to set the maximum file size, in megabytes.
- “Log file” textbox—allows you to set the name of the log file.

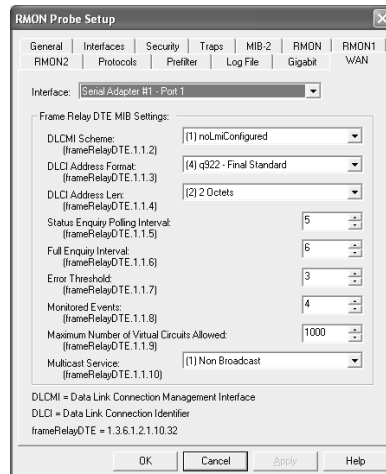
- “Log file viewer” textbox—allows you to set the name of the program, usually an editor, to be used to view the log file when the VIEW LOG button is clicked.

Gigabit Tab



Lets you set the maximum frame size for HCRMON (gigabit) analysis. Note that if you specify a frame size smaller than the actual frame size allowed on your network, decoding and analysis will not work correctly.

WAN Tab



Lets you set MIB definitions for WAN analysis. First, select the adapter number and port from the dropdown menu at the top of the dialog; MIB definitions are applied by adapter # and port. The source

files Network Instruments WAN MIB are installed in the directory where you installed the Probe, in the SNMP subdirectory, and are named as follows:

NETINST-PRODUCTS-MIB.MIB
NETINST-SMI.MIB
NETINST-TC.mib
NETINST-WAN-RMON-MIB.MIB

DLCMI Scheme

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

MIB Object	Description
DLCMI Scheme	Defines which Data Link Connection Management scheme is active (and by implication, which DLCI it uses) on the Frame Relay interface.
DLCI Address Format	Defines which address format is in use on the Frame Relay interface: (1) q921 13 bit DLCI (2) q922March90 11 bit DLCI (3) q922November90 10 bit DLCI (4) q922 Final Standard
DLCI Address Len	Defines the address length in octets. In the case of Q922 format, the length indicates the entire length of the address including the control portion.
Status Enquiry Polling Interval	Defines the number of seconds between successive status enquiry messages.
Full Enquiry Interval	Defines the number of status enquiry intervals that pass before issuance of a full status enquiry message.
Error Threshold	Defines the maximum number of unanswered Status Enquiries the equipment shall accept before declaring the interface down.
Monitored Events	Defines the number of Status Polling Intervals over which the Error Threshold is calculated. In other words, if the number of errors exceeds "Error Threshold" within "Monitored Events" number of Status Polling Intervals, the interface is flagged as down.
Maximum Number of Virtual Circuits Allowed	Defines the maximum number of virtual circuits allowed for this interface, usually defined by the Frame Relay network.
Multicast Service	Defines whether the Frame Relay interface is using a multicast service. Choices are "Broadcast" and "Non-Broadcast."

Appendix

This is a complete listing of all the RMON1 and RMON2 objects and structures, as well as the HCRMON and Frame-Relay-DTE objects and structures used by Network Instruments HCRMON and WAN RMON probes. All of the RMON1/2 objects and structures are supported by the Network Instruments' RMON2 Probe.

RMON1 Tree

1 statistics

- 1.1 etherStatsTable
 - etherStats Entry
 - 1 etherStatsIndex
 - 2 etherStatsDataSource
 - 3 etherStatsDropEvents
 - 4 etherStatsOctets
 - 5 etherStatsPkts
 - 6 etherStatsBroadcastPkts
 - 7 etherStatsMulticastPkts
 - 8 etherStatsCRCAlignErrors
 - 9 etherStatsUndersizePkts
 - 10 etherStatsOversizePkts
 - 11 etherStatsFragments
 - 12 etherStatsJabbers
 - 13 etherStatsCollisions
 - 14 etherStatsPkts64Octets
 - 15 etherStatsPkts65to127Octets
 - 16 etherStatsPkts128to255Octets
 - 17 etherStatsPkts256to511Octets
 - 18 etherStatsPkts512to1023Octets
 - 19 etherStatsPkts1024to1518Octets
 - 20 etherStatsOwner
 - 21 etherStatsStatus
- 2.1 tokenRingMLStatsTable
 - tokenRingMLStatsEntry
 - 1 tokenRingMLStatsIndex
 - 2 tokenRingMLStatsDataSource
 - 3 tokenRingMLStatsDropEvents
 - 4 tokenRingMLStatsMacOctets
 - 5 tokenRingMLStatsMacPkts
 - 6 tokenRingMLStatsRingPurgeEvents
 - 7 tokenRingMLStatsRingPurgePkts
 - 8 tokenRingMLStatsBeaconEvents

9	tokenRingMLStatsBeaconTime
10	tokenRingMLStatsBeaconPkts
11	tokenRingMLStatsClaimTokenEvents
12	tokenRingMLStatsClaimTokenPkts
13	tokenRingMLStatsNAUNChanges
14	tokenRingMLStatsLineErrors
15	tokenRingMLStatsInternalErrors
16	tokenRingMLStatsBurstErrors
17	tokenRingMLStatsACErrors
18	tokenRingMLStatsAbortErrors
19	tokenRingMLStatsLostFrameErrors
20	tokenRingMLStatsCongestionErrors
21	tokenRingMLStatsFrameCopiedErrors
22	tokenRingMLStatsFrequencyErrors
23	tokenRingMLStatsTokenErrors
24	tokenRingMLStatsSoftErrorReports
25	tokenRingMLStatsRingPollEvents
26	tokenRingMLStatsOwner
27	tokenRingMLStatsStatus
3.1	tokenRingPStatsTable
	tokenRingPStatsEntry
1	tokenRingPStatsIndex
2	tokenRingPStatsDataSource
3	tokenRingPStatsDropEvents
4	tokenRingPStatsDataOctets
5	tokenRingPStatsDataPkts
6	tokenRingPStatsDataBroadcastPkts
7	tokenRingPStatsDataMulticastPkts
8	tokenRingPStatsDataPkts18to63Octets
9	tokenRingPStatsDataPkts64to127Octets
10	tokenRingPStatsDataPkts128to255Octets
11	tokenRingPStatsDataPkts256to511Octets
12	tokenRingPStatsDataPkts512to1023Octets
13	tokenRingPStatsDataPkts1024to2047Octets
14	tokenRingPStatsDataPkts2048to4095Octets
15	tokenRingPStatsDataPkts4096to8191Octets
16	tokenRingPStatsDataPkts8192to18000Octets
17	tokenRingPStatsDataPktsGreaterThan18000Octets
18	tokenRingPStatsOwner
19	tokenRingPStatsStatus

2 history

1.1	historyControlTable
-----	---------------------

	historyControlEntry
	1 historyControlIndex
	2 historyControlDataSource
	3 historyControlBucketsRequested
	4 historyControlBucketsGranted
	5 historyControlInterval
	6 historyControlOwner
	7 historyControlStatus
2.1	etherHistoryTable
	etherHistoryEntry
	1 etherHistoryIndex
	2 etherHistorySampleIndex
	3 etherHistoryIntervalStart
	4 etherHistoryDropEvents
	5 etherHistoryOctets
	6 etherHistoryPkts
	7 etherHistoryBroadcastPkts
	8 etherHistoryMulticastPkts
	9 etherHistoryCRCAlignErrors
	10 etherHistoryUndersizePkts
	11 etherHistoryOversizePkts
	12 etherHistoryFragments
	13 etherHistoryJabbers
	14 etherHistoryCollisions
	15 etherHistoryUtilization
3.1	tokenRingMLHistoryTable
	tokenRingMLHistoryEntry
	1 tokenRingMLHistoryIndex
	2 tokenRingMLHistorySampleIndex
	3 tokenRingMLHistoryIntervalStart
	4 tokenRingMLHistoryDropEvents
	5 tokenRingMLHistoryMacOctets
	6 tokenRingMLHistoryMacPkts
	7 tokenRingMLHistoryRingPurgeEvents
	8 tokenRingMLHistoryRingPurgePkts
	9 tokenRingMLHistoryBeaconEvents
	10 tokenRingMLHistoryBeaconTime
	11 tokenRingMLHistoryBeaconPkts
	12 tokenRingMLHistoryClaimTokenEvents
	13 tokenRingMLHistoryClaimTokenPkts
	14 tokenRingMLHistoryNAUNChanges
	15 tokenRingMLHistoryLineErrors
	16 tokenRingMLHistoryInternalErrors

- 17 tokenRingMLHistoryBurstErrors
- 18 tokenRingMLHistoryACErrors
- 19 tokenRingMLHistoryAbortErrors
- 20 tokenRingMLHistoryLostFrameErrors
- 21 tokenRingMLHistoryCongestionErrors
- 22 tokenRingMLHistoryFrameCopiedErrors
- 23 tokenRingMLHistoryFrequencyErrors
- 24 tokenRingMLHistoryTokenErrors
- 25 tokenRingMLHistorySoftErrorReports
- 26 tokenRingMLHistoryRingPollEvents
- 27 tokenRingMLHistoryActiveStations
- 4.1 tokenRingPHistoryTable
 - tokenRingPHistoryEntry
 - 1 tokenRingPHistoryIndex
 - 2 tokenRingPHistorySampleIndex
 - 3 tokenRingPHistoryIntervalStart
 - 4 tokenRingPHistoryDropEvents
 - 5 tokenRingPHistoryDataOctets
 - 6 tokenRingPHistoryDataPkts
 - 7 tokenRingPHistoryDataBroadcastPkts
 - 8 tokenRingPHistoryDataMulticastPkts
 - 9 tokenRingPHistoryDataPkts18to63Octets
 - 10 tokenRingPHistoryDataPkts64to127Octets
 - 11 tokenRingPHistoryDataPkts128to255Octets
 - 12 tokenRingPHistoryDataPkts256to511Octets
 - 13 tokenRingPHistoryDataPkts512to1023Octets
 - 14 tokenRingPHistoryDataPkts1024to2047Octets
 - 15 tokenRingPHistoryDataPkts2048to4095Octets
 - 16 tokenRingPHistoryDataPkts4096to8191Octets
 - 17 tokenRingPHistoryDataPkts8192to18000Octets
 - 18 tokenRingPHistoryDataPktsGreater Than 18000Octets

3 alarm

- 1.1 alarmTable
 - alarmEntry
 - 1 alarmIndex
 - 2 alarmInterval
 - 3 alarmVariable
 - 4 alarmSampleType
 - 5 alarmValue
 - 6 alarmStartupAlarm
 - 7 alarmRisingThreshold
 - 8 alarmFallingThreshold

- 9 alarmRisingEventIndex
- 10 alarmFallingEventIndex
- 11 alarmOwner
- 12 alarmStatus

4 host

- 1.1 hostControlTable
 - hostControlEntry
 - 1 hostControlIndex
 - 2 hostControlDataSource
 - 3 hostControlTableSize
 - 4 hostControlLastDeleteTime
 - 5 hostControlOwner
 - 6 hostControlStatus
- 2.1 hostTable
 - hostEntry
 - 1 hostAddress
 - 2 hostCreationOrder
 - 3 hostIndex
 - 4 hostInPkts
 - 5 hostOutPkts
 - 6 hostInOctets
 - 7 hostOutOctets
 - 8 hostOutErrors
 - 9 hostOutBroadcastPkts
 - 10 hostOutMulticastPkts
- 3.1 hostTimeTable
 - hostTimeEntry
 - 1 hostTimeAddress
 - 2 hostTimeCreationOrder
 - 3 hostTimeIndex
 - 4 hostTimeInPkts
 - 5 hostTimeOutPkts
 - 6 hostTimeInOctets
 - 7 hostTimeOutOctets
 - 8 hostTimeOutErrors
 - 9 hostTimeOutBroadcastPkts
 - 10 hostTimeOutMulticastPkts

5 hostTopN

- 1.1 hostTopNControlTable
 - hostTopNControlEntry
 - 1 hostTopNControlIndex

- 2 hostTopNHostIndex
- 3 hostTopNRateBase
- 4 hostTopNTimeRemaining
- 5 hostTopNDuration
- 6 hostTopNRequestedSize
- 7 hostTopNGrantedSize
- 8 hostTopNStartTime
- 9 hostTopNOwner
- 10 hostTopNStatus
- 2.1 hostTopNTable
 - hostTopNEntry
 - 1 hostTopNReport
 - 2 hostTopNIndex
 - 3 hostTopNAddress
 - 4 hostTopNRate
- 6 matrix
 - 1.1 matrixControlTable
 - matrixControlEntry
 - 1 matrixControlIndex
 - 2 matrixControlDataSource
 - 3 matrixControlTableSize
 - 4 matrixControlLastDeleteTime
 - 5 matrixControlOwner
 - 6 matrixControlStatus
 - 2.1 matrixSDTable
 - matrixSDEntry
 - 1 matrixSDSourceAddress
 - 2 matrixSDDestAddress
 - 3 matrixSDIndex
 - 4 matrixSDPkts
 - 5 matrixSDOctets
 - 6 matrixSDErrors
 - 3.1 matrixDSTable
 - matrixDSEntry
 - 1 matrixDSSourceAddress
 - 2 matrixDSDestAddress
 - 3 matrixDSIndex
 - 4 matrixDSPkts
 - 5 matrixDSOctets
 - 6 matrixDSErrors
- 7 filter
 - 1.1 filterTable
 - filterEntry

	1	filterIndex
	2	filterChannelIndex
	3	filterPktDataOffset
	4	filterPktData
	5	filterPktDataMask
	6	filterPktDataNotMask
	7	filterPktStatus
	8	filterPktStatusMask
	9	filterPktStatusNotMask
	10	filterOwner
	11	filterStatus
2.1		channelTable
		channelEntry
	1	channelIndex
	2	channelIfIndex
	3	channelAcceptType
	4	channelDataControl
	5	channelTurnOnEventIndex
	6	channelTurnOffEventIndex
	7	channelEventIndex
	8	channelEventStatus
	9	channelMatches
	10	channelDescription
	11	channelOwner
	12	channelStatus
8 capture		
1.1		bufferControlTable
		bufferControlEntry
	1	bufferControlIndex
	2	bufferControlChannelIndex
	3	bufferControlFullStatus
	4	bufferControlFullAction
	5	bufferControlCaptureSliceSize
	6	bufferControlDownloadSliceSize
	7	bufferControlDownloadOffset
	8	bufferControlMaxOctetsRequested
	9	bufferControlMaxOctetsGranted
	10	bufferControlCapturedPackets
	11	bufferControlTurnOnTime
	12	bufferControlOwner
	13	bufferControlStatus
2.1		captureBufferTable
		captureBufferEntry

- 1 captureBufferControlIndex
- 2 captureBufferIndex
- 3 captureBufferPacketID
- 4 captureBufferPacketData
- 5 captureBufferPacketLength
- 6 captureBufferPacketTime
- 7 captureBufferPacketStatus

9 event

- 1.1 eventTable
 - eventEntry
 - 1 eventIndex
 - 2 eventDescription
 - 3 eventType
 - 4 eventCommunity
 - 5 eventLastTimeSent
 - 6 eventOwner
 - 7 eventStatus
- 2.1 logTable
 - logEntry
 - 1 logEventIndex
 - 2 logIndex
 - 3 logTime
 - 4 logDescription

10 tokenRing

- 1.1 ringStationControlTable
 - ringStationControlEntry
 - 1 ringStationControlIfIndex
 - 2 ringStationControlTableSize
 - 3 ringStationControlActiveStations
 - 4 ringStationControlRingState
 - 5 ringStationControlBeaconSender
 - 6 ringStationControlBeaconNAUN
 - 7 ringStationControlActiveMonitor
 - 8 ringStationControlOrderChanges
 - 9 ringStationControlOwner
 - 10 ringStationControlStatus
- 2.1 ringStationTable
 - ringStationEntry
 - 1 ringStationIfIndex
 - 2 ringStationMacAddress
 - 3 ringStationLastNAUN

4	ringStationStationStatus
5	ringStationLastEnterTime
6	ringStationLastExitTime
7	ringStationDuplicateAddresses
8	ringStationInLineErrors
9	ringStationOutLineErrors
10	ringStationInternalErrors
11	ringStationInBurstErrors
12	ringStationOutBurstErrors
13	ringStationACErrors
14	ringStationAbortErrors
15	ringStationLostFrameErrors
16	ringStationCongestionErrors
17	ringStationFrameCopiedErrors
18	ringStationFrequencyErrors
19	ringStationTokenErrors
20	ringStationInBeaconErrors
21	ringStationOutBeaconErrors
22	ringStationInsertions
3.1	ringStationOrderTable
	ringStationOrderEntry
1	ringStationOrderIfIndex
2	ringStationOrderOrderIndex
3	ringStationOrderMacAddress
4.1	ringStationConfigControlTable
	ringStationConfigControlEntry
1	ringStationConfigControlIfIndex
2	ringStationConfigControlMacAddress
3	ringStationConfigControlRemove
4	ringStationConfigControlUpdateStats
5.1	ringStationConfigTable
	ringStationConfigEntry
1	ringStationConfigIfIndex
2	ringStationConfigMacAddress
3	ringStationConfigUpdateTime
4	ringStationConfigLocation
5	ringStationConfigMicrocode
6	ringStationConfigGroupAddress
7	ringStationConfigFunctionalAddress
6.1	sourceRoutingStatsTable
	sourceRoutingStatsEntry
1	sourceRoutingStatsIfIndex
2	sourceRoutingStatsRingNumber

- 3 sourceRoutingStatsInFrames
- 4 sourceRoutingStatsOutFrames
- 5 sourceRoutingStatsThroughFrames
- 6 sourceRoutingStatsAllRoutesBroadcastFrames
- 7 sourceRoutingStatsSingleRouteBroadcastFrames
- 8 sourceRoutingStatsInOctets
- 9 sourceRoutingStatsOutOctets
- 10 sourceRoutingStatsThroughOctets
- 11 sourceRoutingStatsAllRoutesBroadcastOctets
- 12 sourceRoutingStatsSingleRoutesBroadcastOctets
- 13 sourceRoutingStatsLocalLLCFrames
- 14 sourceRoutingStats1HopFrames
- 15 sourceRoutingStats2HopsFrames
- 16 sourceRoutingStats3HopsFrames
- 17 sourceRoutingStats4HopsFrames
- 18 sourceRoutingStats5HopsFrames
- 19 sourceRoutingStats6HopsFrames
- 20 sourceRoutingStats7HopsFrames
- 21 sourceRoutingStats8HopsFrames
- 22 sourceRoutingStatsMoreThan8HopsFrames
- 23 sourceRoutingStatsOwner
- 24 sourceRoutingStatsStatus

RMON2 Tree

11 protocolDir

- 1 protocolDirLastChange
- 2.1 protocolDirTable.protocolDirEntry
 - 1 protocolDirID
 - 2 protocolDirParameters
 - 3 protocolDirLocalIndex
 - 4 protocolDirDescr
 - 5 protocolDirType
 - 6 protocolDirAddressMapConfig
 - 7 protocolDirHostConfig
 - 8 protocolDirMatrixConfig
 - 9 protocolDirOwner
 - 10 protocolDirStatus

12 protocolDist

- 1.1 protocolDistControlTable
 - protocolDistControlEntry
 - 1 protocolDistControlIndex

- 2 protocolDistControlDataSource
- 3 protocolDistControlDroppedFrames
- 4 protocolDistControlCreateTime
- 5 protocolDistControlOwner
- 6 protocolDistControlStatus
- 2.1 protocolDistStatsTable.protocolDistStatsEntry
 - 1 protocolDistStatsPkts
 - 2 protocolDistStatsOctets

13 addressMap

- 1 addressMapInserts
- 2 addressMapDeletes
- 3 addressMapMaxDesiredEntries
- 4.1 addressMapControlTable
 - addressMapControlEntry
 - 1 addressMapControlIndex
 - 2 addressMapControlDataSource
 - 3 addressMapControlDroppedFrames
 - 4 addressMapControlOwner
 - 5 addressMapControlStatus
- 5.1 addressMapTable
 - addressMapEntry
 - 1 addressMapTimeMark
 - 2 addressMapNetworkAddress
 - 3 addressMapSource
 - 4 addressMapPhysicalAddress
 - 5 addressMapLastChange

14 nlHost

- 1.1 hlHostControlTable
 - hlHostControlEntry
 - 1 hlHostControlIndex
 - 2 hlHostControlDataSource
 - 3 hlHostControlNIDroppedFrames
 - 4 hlHostControlNIInserts
 - 5 hlHostControlNIDeletes
 - 6 hlHostControlNIMaxDesiredEntries
 - 7 hlHostControlAIDroppedFrames
 - 8 hlHostControlAIInserts
 - 9 hlHostControlAIDeletes
 - 10 hlHostControlAIMaxDesiredEntries
 - 11 hlHostControlOwner
 - 12 hlHostControlStatus

- 2.1 nlHostTable
 - nlHostEntry
 - 1 nlHostTimeMark
 - 2 nlHostAddress
 - 3 nlHostInPkts
 - 4 nlHostOutPkts
 - 5 nlHostInOctets
 - 6 nlHostOutOctets
 - 7 nlHostOutMacNonUnicastPkts
 - 8 nlHostCreateTime

15 nlMatrix

- 1.1 hlMatrixControlTable
 - hlMatrixControlEntry
 - 1 hlMatrixControlIndex
 - 2 hlMatrixControlDataSource
 - 3 hlMatrixControlNIDroppedFrames
 - 4 hlMatrixControlNIInserts
 - 5 hlMatrixControlNIDeletes
 - 6 hlMatrixControlNIMaxDesiredEntries
 - 7 hlMatrixControlAIDroppedFrames
 - 8 hlMatrixControlAIInserts
 - 9 hlMatrixControlAIDeletes
 - 10 hlMatrixControlAIMaxDesiredEntries
 - 11 hlMatrixControlOwner
 - 12 hlMatrixControlStatus
- 2.1 nlMatrixSDTable
 - nlMatrixSDEntry
 - 1 nlMatrixSDTimeMark
 - 2 nlMatrixSDSourceAddress
 - 3 nlMatrixSDDestAddress
 - 4 nlMatrixSDPkts
 - 5 nlMatrixSDOctets
 - 6 nlMatrixSDCreateTime
- 3.1 nlMatrixDSTable
 - nlMatrixDSEntry
 - 1 nlMatrixDSTimeMark
 - 2 nlMatrixDSSourceAddress
 - 3 nlMatrixDSDestAddress
 - 4 nlMatrixDSPkts
 - 5 nlMatrixDSOctets
 - 6 nlMatrixDSCreateTime
- 4.1 nlMatrixTopNControlTable

- nlMatrixTopNControlEntry
 - 1 nlMatrixTopNControlIndex
 - 2 nlMatrixTopNControlMatrixIndex
 - 3 nlMatrixTopNControlRateBase
 - 4 nlMatrixTopNControlTimeRemaining
 - 5 nlMatrixTopNControlGeneratedReports
 - 6 nlMatrixTopNControlDuration
 - 7 nlMatrixTopNControlRequestedSize
 - 8 nlMatrixTopNControlGrantedSize
 - 9 nlMatrixTopNControlStartTime
 - 10 nlMatrixTopNControlOwner
 - 11 nlMatrixTopNControlStatus

5.1 nlMatrixTopNTable

- nlMatrixTopNEntry
 - 1 nlMatrixTopNIndex
 - 2 nlMatrixTopNProtocolDirLocalIndex
 - 3 nlMatrixTopNSourceAddress
 - 4 nlMatrixTopNDestAddress
 - 5 nlMatrixTopNPktRate
 - 6 nlMatrixTopNReversePktRate
 - 7 nlMatrixTopNOctetRate
 - 8 nlMatrixTopNReverseOctetRate

16 alHost

1.1 alHostTable

- alHostEntry
 - 1 alHostTimeMark
 - 2 alHostInPkts
 - 3 alHostOutPkts
 - 4 alHostInOctets
 - 5 alHostOutOctets
 - 6 alHostCreateTime

17 alMatrix

1.1 alMatrixSDTable

- alMatrixSDEntry
 - 1 alMatrixSDTimeMark
 - 2 alMatrixSDPkts
 - 3 alMatrixSDOctets
 - 4 alMatrixSDCreateTime

2.1 alMatrixDSTable

- alMatrixDSEntry
 - 1 alMatrixDSTimeMark

- 2 alMatrixDSPkts
- 3 alMatrixDSOctets
- 4 alMatrixDSCreateTime
- 3.1 alMatrixTopNControlTable
 - alMatrixTopNControlEntry
 - 1 alMatrixTopNControlIndex
 - 2 alMatrixTopNControlMatrixIndex
 - 3 alMatrixTopNControlRateBase
 - 4 alMatrixTopNControlTimeRemaining
 - 5 alMatrixTopNControlGeneratedReports
 - 6 alMatrixTopNControlDuration
 - 7 alMatrixTopNControlRequestedSize
 - 8 alMatrixTopNControlGrantedSize
 - 9 alMatrixTopNControlStartTime
 - 10 alMatrixTopNControlOwner
 - 11 alMatrixTopNControlStatus
- 4.1 alMatrixTopNTable
 - alMatrixTopNEntry
 - 1 alMatrixTopNIndex
 - 2 alMatrixTopNProtocolDirLocalIndex
 - 3 alMatrixTopNSourceAddress
 - 4 alMatrixTopNDestAddress
 - 5 alMatrixTopNAppProtocolDirLocalIndex
 - 6 alMatrixTopNPktRate
 - 7 alMatrixTopNReversePktRate
 - 8 alMatrixTopNOctetRate
 - 9 alMatrixTopNReverseOctetRate

18 usrHistory

- 1.1 usrHistoryControlTable
 - usrHistoryControlEntry
 - 1 usrHistoryControlIndex
 - 2 usrHistoryControlObjects
 - 3 usrHistoryControlBucketsRequested
 - 4 usrHistoryControlBucketsGranted
 - 5 usrHistoryControlInterval
 - 6 usrHistoryControlOwner
 - 7 usrHistoryControlStatus
- 2.1 usrHistoryObjectTable
 - usrHistoryObjectEntry
 - 1 usrHistoryObjectIndex
 - 2 usrHistoryObjectVariable
 - 3 usrHistoryObjectSampleType

- 3.1 usrHistoryTable
 - usrHistoryTableEntry
 - 1 usrHistorySampleIndex
 - 2 usrHistoryIntervalStart
 - 3 usrHistoryIntervalEnd
 - 4 usrHistoryAbsValue
 - 5 usrHistoryValStatus

19 **ProbeConfig**

- 1 ProbeCapabilities
- 2 ProbeSoftwareRev
- 3 ProbeHardwareRev
- 4 ProbeDateTime
- 5 ProbeResetControl
- 6 ProbeDownloadFile
- 7 ProbeDownloadTFTPServer
- 8 ProbeDownloadAction
- 9 ProbeDownloadStatus
- 13.1 trapDestTable
 - trapDestEntry
 - 1 trapDestIndex
 - 2 trapDestCommunity
 - 3 trapDestProtocol
 - 4 trapDestAddress
 - 5 trapDestOwner
 - 6 trapDestStatus

RMON1 Extension Tables

1 Statistics

- 4.1 etherStats2Table
 - etherStats2Entry
 - 1 etherStatsDroppedFrames
 - 2 etherStatsCreateTime
- 5.1 tokenRingMLStats2Table
 - tokenRingMLStats2Entry
 - 1 tokenRingMLStatsDroppedFrames
 - 2 tokenRingMLStatsCreateTime
- 6.1 tokenRingPStats2Table
 - tokenRingPStats2Entry
 - 1 tokenRingPStatsDroppedFrames
 - 2 tokenRingPStatsCreateTime

2 History

- 5.1 historyControl2Table
historyControl2Entry
1 historyControlDroppedFrames

4 Host

- 4.1 hostControl2Table
hostControl2Entry
1 hostControlDroppedFrames
2 hostControlCreateTime

6 Matrix

- 4.1 matrixControl2Table
matrixControl2Entry
1 matrixControlDroppedFrames
2 matrixControlCreateTime

7 Filter

- 3.1 channel2Table
channel2Entry
1 channelDroppedFrames
2 channelCreateTime
- 4.1 filter2Table
filter2Entry
1 filterProtocolDirDataLocalIndex
2 filterProtocolDirLocalIndex

10 TokenRing

- 7.1 ringStationControl2Table
ringStationControl2Entry
1 ringStationControlDroppedFrames
2 ringStationControlCreateTime
- 8.1 sourceRoutingStats2Table
sourceRoutingStats2Entry
1 sourceRoutingStatsDroppedFrames
2 sourceRoutingStatsCreateTime

HCRMON Table (1.3.6.1.2.1.16 RMON2 Tree)

21 mediaIndependentStats

- 1.1 mediaIndependentTable.mediaIndependentEntry
1 mediaIndependentIndex (Index-1)

2	mediaIndependentDataSource
3	mediaIndependentDropEvents
4	mediaIndependentDroppedFrames
5	mediaIndependentInPkts
6	mediaIndependentInOverflowPkts
7	mediaIndependentInHighCapacityPkts
8	mediaIndependentOutPkts
9	mediaIndependentOutOverflowPkts
10	mediaIndependentOutHighCapacityPkts
11	mediaIndependentInOctets
12	mediaIndependentInOverflowOctets
13	mediaIndependentInHighCapacityOctets
14	mediaIndependentOutOctets
15	mediaIndependentOutOverflowOctets
16	mediaIndependentOutHighCapacityOctets
17	mediaIndependentInNUCastPkts
18	mediaIndependentInNUCastOverflowPkts
19	mediaIndependentInNUCastHighCapacityPkts
20	mediaIndependentOutNUCastPkts
21	mediaIndependentOutNUCastOverflowPkts
22	mediaIndependentOutNUCastHighCapacityPkts
23	mediaIndependentInErrors
24	mediaIndependentOutErrors
25	mediaIndependentInputSpeed
26	mediaIndependentOutputSpeed
27	mediaIndependentDuplexMode
28	mediaIndependentDuplexChanges
29	mediaIndependentDuplexLastChange
30	mediaIndependentOwner
31	mediaIndependentStatus

HCRMON Extensions to RMON1/2 Tables

1 statistics

7.1	etherStatsHighCapacityTable.etherStatsHighCapacityEntry (<i>Index = etherStatsIndex</i>)
1	etherStatsHighCapacityOverflowPkts
2	etherStatsHighCapacityPkts
3	etherStatsHighCapacityOverflowOctets
4	etherStatsHighCapacityOctets
5	etherStatsHighCapacityOverflowPkts64Octets
6	etherStatsHighCapacityPkts64Octets
7	etherStatsHighCapacityOverflowPkts65to127Octets

- 8 etherStatsHighCapacityPkts65to127Octets
- 9 etherStatsHighCapacityOverflowPkts128to255Octets
- 10 etherStatsHighCapacityPkts128to255Octets
- 11 etherStatsHighCapacityOverflowPkts256to511Octets
- 12 etherStatsHighCapacityPkts256to511Octets
- 13 etherStatsHighCapacityOverflowPkts512to1023Octets
- 14 etherStatsHighCapacityPkts512to1023Octets
- 15 etherStatsHighCapacityOverflowPkts1024to1518Octets
- 16 etherStatsHighCapacityPkts1024to1518Octets

2. history

- 6.1 etherHistoryHighCapacityTable.EtherHistoryHighCapacityEntry
(*Index = etherHistoryIndex, etherHistorySampleIndex*)
 - 1 etherHistoryHighCapacityOverflowPkts
 - 2 etherHistoryHighCapacityPkts
 - 3 etherHistoryHighCapacityOverflowOctets
 - 4 etherHistoryHighCapacityOctets

4. host

- 5.1 hostHighCapacityTable. hostHighCapacityEntry
(*Index = hostIndex, hostAddress*)
 - 1 hostHighCapacityInOverflowPkts
 - 2 hostHighCapacityInPkts
 - 3 hostHighCapacityOutOverflowPkts
 - 4 hostHighCapacityOutPkts
 - 5 hostHighCapacityInOverflowOctets
 - 6 hostHighCapacityInOctets
 - 7 hostHighCapacityOutOverflowOctets
 - 8 hostHighCapacityOutOctets

4. host

- 6.1 hostTimeHighCapacityTable.hostTimeHighCapacityEntry
(*Index = hostTimeIndex, hostTimeCreationOrder*)
 - 1 hostTimeHighCapacityInOverflowPkts
 - 2 hostTimeHighCapacityInPkts
 - 3 hostTimeHighCapacityOutOverflowPkts
 - 4 hostTimeHighCapacityOutPkts
 - 5 hostTimeHighCapacityInOverflowOctets
 - 6 hostTimeHighCapacityInOctets
 - 7 hostTimeHighCapacityOutOverflowOctets
 - 8 hostTimeHighCapacityOutOctets

5. hostTopN

- 3.1 hostTopNHighCapacityTable.hostTopNHighCapacityEntry
(*Index = hostTopNReport, hostTopNIndex*)
- 1 hostTopNHighCapacityAddress
 - 2 hostTopNHighCapacityBaseRate
 - 3 hostTopNHighCapacityOverflowRate
 - 4 hostTopNHighCapacityRate

6. Matrix

- 5.1 matrixSDHighCapacityTable.matrixSDHighCapacityEntry
(*Index = matrixSDIndex, matrixSDSourceAddress, matrixSDDestAddress*)
- 1 matrixSDHighCapacityOverflowPkts
 - 2 matrixSDHighCapacityPkts
 - 3 matrixSDHighCapacityOverflowOctets
 - 4 matrixSDHighCapacityOctets

8. capture

- 3.1 captureBufferHighCapacityTable.captureBufferHighCapacityEntry
INDEX { captureBufferControlIndex, captureBufferIndex }
- 1 captureBufferPacketHighCapacityTime Integer32

12. protDist

- 3.1 protocolDistStatsHighCapacityTable.protocolDistStatsHighCapacityEntry
INDEX { protocolDistControlIndex, protocolDirLocalIndex }
- 1 protocolDistStatsHighCapacityOverflowPkts
 - 2 protocolDistStatsHighCapacityPkts
 - 3 protocolDistStatsHighCapacityOverflowOctets
 - 4 protocolDistStatsHighCapacityOctets

14. nlHost

- 3.1 nlHostHighCapacityTable.nlHostHighCapacityEntry
INDEX { nlHostControlIndex, nlHostTimeMark, protocolDirLocalIndex, nlHostAddress }
- 1 nlHostHighCapacityInOverflowPkts
 - 2 nlHostHighCapacityInPkt
 - 3 nlHostHighCapacityOutOverflowPkts
 - 4 nlHostHighCapacityOutPkts
 - 5 nlHostHighCapacityInOverflowOctets
 - 6 nlHostHighCapacityInOctets
 - 7 nlHostHighCapacityOutOverflowOctets
 - 8 nlHostHighCapacityOutOctets

15. nlMatrix

6.1 nlMatrixSDHighCapacityTable.nlMatrixSDHighCapacityEntry

*INDEX { hlMatrixControlIndex, nlMatrixSDTimeMark,
protocolDirLocalIndex, nlMatrixSDSourceAddress,
nlMatrixSDDestAddress }*

- 1 nlMatrixSDHighCapacityOverflowPkts
- 2 nlMatrixSDHighCapacityPkts
- 3 nlMatrixSDHighCapacityOverflowOctets
- 4 nlMatrixSDHighCapacityOctets

7.1 nlMatrixDSHighCapacityTable.nlMatrixDSHighCapacityEntry

*INDEX { hlMatrixControlIndex, nlMatrixDSTimeMark,
protocolDirLocalIndex, nlMatrixDSDestAddress,
nlMatrixDSSourceAddress }*

- 1 nlMatrixDSHighCapacityOverflowPkts
- 2 nlMatrixDSHighCapacityPkts
- 3 nlMatrixDSHighCapacityOverflowOctets
- 4 nlMatrixDSHighCapacityOctets

8.1 nlMatrixTopNHighCapacityTable.nlMatrixTopNHighCapacityEntry

INDEX { nlMatrixTopNControlIndex, nlMatrixTopNIndex }

- 1 nlMatrixTopNHighCapacityProtocolDirLocalIndex
- 2 nlMatrixTopNHighCapacitySourceAddress
- 3 nlMatrixTopNHighCapacityDestAddress
- 4 nlMatrixTopNHighCapacityBasePktRate
- 5 nlMatrixTopNHighCapacityOverflowPktRate
- 6 nlMatrixTopNHighCapacityPktRate
- 7 nlMatrixTopNHighCapacityReverseBasePktRate
- 8 nlMatrixTopNHighCapacityReverseOverflowPktRate
- 9 nlMatrixTopNHighCapacityReversePktRate
- 10 nlMatrixTopNHighCapacityBaseOctetRate
- 11 nlMatrixTopNHighCapacityOverflowOctetRate
- 12 nlMatrixTopNHighCapacityOctetRate
- 13 nlMatrixTopNHighCapacityReverseBaseOctetRate
- 14 nlMatrixTopNHighCapacityReverseOverflowOctetRate
- 15 nlMatrixTopNHighCapacityReverseOctetRate

16. alHost

2.1 alHostHighCapacityTable.alHostHighCapacityEntry

*INDEX { hlHostControlIndex, alHostTimeMark,
protocolDirLocalIndex, nlHostAddress,
protocolDirLocalIndex }*

- 1 alHostHighCapacityInOverflowPkts
- 2 alHostHighCapacityInPkts
- 3 alHostHighCapacityOutOverflowPkts
- 4 alHostHighCapacityOutPkts
- 5 alHostHighCapacityInOverflowOctets
- 6 alHostHighCapacityInOctets
- 7 alHostHighCapacityOutOverflowOctets
- 8 alHostHighCapacityOutOctets

17. alMatrix

- 5.1 alMatrixSDHighCapacityTable.alMatrixSDHighCapacityEntry
INDEX { hlMatrixControlIndex, alMatrixSDTimeMark, protocolDirLocalIndex, nlMatrixSDSourceAddress, nlMatrixSDDestAddress, protocolDirLocalIndex }
 - 1 alMatrixSDHighCapacityOverflowPkts
 - 2 alMatrixSDHighCapacityPkts
 - 3 alMatrixSDHighCapacityOverflowOctets
 - 4 alMatrixSDHighCapacityOctets

- 6.1 alMatrixDSHighCapacityTable.alMatrixDSHighCapacityEntry
INDEX { hlMatrixControlIndex, alMatrixDSTimeMark, protocolDirLocalIndex, nlMatrixDSDestAddress, nlMatrixDSSourceAddress, protocolDirLocalIndex }
 - 1 alMatrixDSHighCapacityOverflowPkts
 - 2 alMatrixDSHighCapacityPkts
 - 3 alMatrixDSHighCapacityOverflowOctets
 - 4 alMatrixDSHighCapacityOctetsZeroBasedCounter64

- 7.1 alMatrixTopNHighCapacityTable.alMatrixTopNHighCapacityEntry
INDEX { alMatrixTopNControlIndex, alMatrixTopNIndex }
 - 1 alMatrixTopNHighCapacityProtocolDirLocalIndex
 - 2 alMatrixTopNHighCapacitySourceAddress
 - 3 alMatrixTopNHighCapacityDestAddress
 - 4 alMatrixTopNHighCapacityAppProtocolDirLocalIndex
 - 5 alMatrixTopNHighCapacityBasePktRate
 - 6 alMatrixTopNHighCapacityOverflowPktRate
 - 7 alMatrixTopNHighCapacityPktRate
 - 8 alMatrixTopNHighCapacityReverseBasePktRate
 - 9 alMatrixTopNHighCapacityReverseOverflowPktRate
 - 10 alMatrixTopNHighCapacityReversePktRate
 - 11 alMatrixTopNHighCapacityBaseOctetRate
 - 12 alMatrixTopNHighCapacityOverflowOctetRate
 - 13 alMatrixTopNHighCapacityOctetRate

- 14 alMatrixTopNHighCapacityReverseBaseOctetRate
- 15 alMatrixTopNHighCapacityReverseOverflowOctetRate
- 16 alMatrixTopNHighCapacityReverseOctetRate

18. usrHistory

- 4.1 usrHistoryHighCapacityTable.usrHistoryHighCapacityEntry
INDEX { usrHistoryControlIndex, usrHistorySampleIndex, usrHistoryObjectIndex }
 - 1 usrHistoryHighCapacityOverflowAbsValue
 - 2 usrHistoryHighCapacityAbsValue

19. probeConfig

- 16 hcRMONCapabilitiesSYNTAX BITS
 - mediaIndependentGroup(0)
 - etherStatsHighCapacityGroup(1)
 - etherHistoryHighCapacityGroup(2)
 - hostHighCapacityGroup(3)
 - hostTopNHighCapacityGroup(4)
 - matrixHighCapacityGroup(5)
 - captureBufferHighCapacityGroup(6)
 - protocolDistributionHighCapacityGroup(7)
 - nlHostHighCapacityGroup(8)
 - nlMatrixHighCapacityGroup(9)
 - nlMatrixTopNHighCapacityGroup(10)
 - lHostHighCapacityGroup(11)
 - lMatrixHighCapacityGroup(12)
 - lMatrixTopNHighCapacityGroup(13)
 - usrHistoryHighCapacityGroup(14)

FrameRelayDTE Table

1.1 frDlcmiTable. frDlcmiEntry *(Index = frDlcmiIfIndex)*

1	frDlcmiIfIndex	InterfaceIndex
2	frDlcmiState	INTEGER
3	frDlcmiAddress	INTEGER
4	frDlcmiAddressLen	INTEGER
5	frDlcmiPollingInterval	Integer32
6	frDlcmiFullEnquiryInterval	Integer32
7	frDlcmiErrorThreshold	Integer32
8	frDlcmiMonitoredEvents	Integer32
9	frDlcmiMaxSupportedVCs	DLCI
10	frDlcmiMulticast	INTEGER

11	frDlcmiStatus	INTEGER
12	frDlcmiRowStatus	RowStatus

2.1 frCircuitTable.FrCircuitEntry (INDEX = frCircuitIfIndex, frCircuitDlci)

1	frCircuitIfIndex	InterfaceIndex
2	frCircuitDlci	DLCI
3	frCircuitState	INTEGER
4	frCircuitReceivedFECNs	Counter32
5	frCircuitReceivedBECNs	Counter32
6	frCircuitSentFrames	Counter32
7	frCircuitSentOctets	Counter32
8	frCircuitReceivedFrames	Counter32
9	frCircuitReceivedOctets	Counter32
10	frCircuitCreationTime	TimeStamp
11	frCircuitLastTimeChange	TimeStamp
12	frCircuitCommittedBurst	Integer32
13	frCircuitExcessBurst	Integer32
14	frCircuitThroughput	Integer32
15	frCircuitMulticast	INTEGER
16	frCircuitType	INTEGER
17	frCircuitDiscards	Counter32
18	frCircuitReceivedDEs	Counter32
19	frCircuitSentDEs	Counter32
20	frCircuitLogicalIfIndex	InterfaceIndex
21	frCircuitRowStatus	RowStatus

3.1 frErrTable.FrErrEntry (INDEX = frErrIfIndex)

1	frErrIfIndex	InterfaceIndex
2	frErrType	INTEGER
3	frErrData	OCTET STRING
4	frErrTime	TimeStamp
5	frErrFaults	Counter32
6	frErrFaultTime	TimeStamp
0	frameRelayTraps	OBJECT IDENTIFIER

4 frameRelayTrapControl

1	frTrapState	INTEGER
2	frTrapMaxRate	Integer32

Index

A

- Advanced Probe 2
 - port usage 27
- Advanced Probe Window 14
- Advanced Single-Probe 2
- Appendix 42
- Authentication Server 12

D

- Dongles 10

E

- End User License Agreement i
- ErrorTrak
 - drivers 9
- Ethernet Errors By Station 8
- EULA i
- Expert Probe 27

G

- General Tab 31
- Gigabit Tab 40

H

- HCRMON Probe 4
- HCRMON Probe Configuration Parameters 40
- HCRMON Table 57

I

- Installation 6
- Interfaces Tab 31
- Introduction 1

L

- License Agreement i
- licensing 9, 11, 15
- Limited Warranty i
- Log File Tab 39
- logging 39

M

- maximum frame size for HCRMON 40
- memory management 23
- MIB-2 Tab 33
- Minimum Hardware 6
- multiple sessions, configuring 17
- Multi-Probes 17

N

- NIC driver installation 8

O

- Observer Encryption Key file 12
- Operating Systems Supported 6
- Overview 1

P

- port usage 27
- Prefilter Tab 38
- Probe
 - licensing 9, 11, 15
- Probe Instances 17
- Probe Settings 15
- Probe Web snapshot 12, 16
- Probe, RMON2 4
- Protocols Tab 38

R

- Read Community name 32
- Recommended Further Reading 28
- RMON Probe Configuration 29
- RMON Tab 34
- RMON1 Extension Tables 56
- RMON1 Tab 35
- RMON1 Tree 42
- RMON2 Probe 4
- RMON2 Probe Window 29
- RMON2 Tab 36
- RMON2 Tree 51

S

- Security Tab 32
- sessions 3
- Single-Probe, Advanced 2
- SNMP Tab 40

T

- Technical Support i–ii
- Traps Tab 33

U

- User Accounts, configuring 20
- Using the Advanced Multi-Probe 14
- Using the Advanced Probe 11
- Using the RMON Probe 27

W

- WAN RMON Probe 4
- WAN Tab 40
- What Probes Do 1
- Write Community name 32