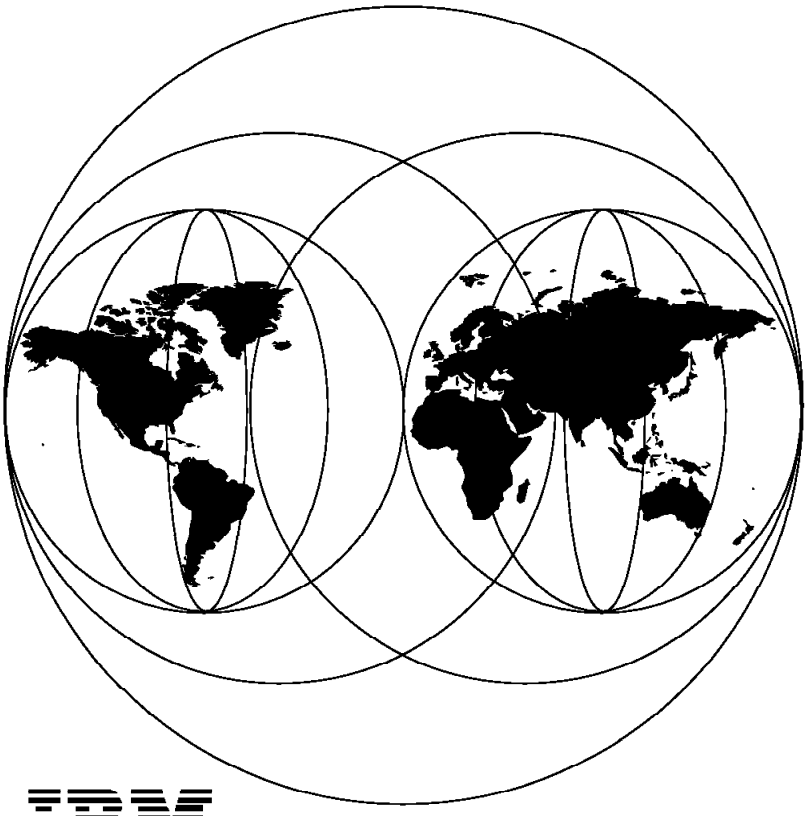
International Technical Support Organization

SG24-4398-01

IBM Systems Monitor
Anatomy of a Smart Agent

February 1996



IBM

International Technical Support Organization
Raleigh Center

International Technical Support Organization

SG24-4398-01

IBM Systems Monitor
Anatomy of a Smart Agent

February 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information under “Special Notices” on page xvii.

Second Edition (February 1996)

This edition applies to: Version 2 Release 2 of the Systems Monitor for AIX feature of SystemView for AIX (program number 5765-527) for use with the AIX for Risc System/6000 operating system.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSO Technical Bulletin Evaluation Form for reader's feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 678
P.O. Box 12195
Research Triangle Park, NC 27709-2195

Abstract

This document describes the use of Systems Monitor for AIX Version 2 Release 2 and related IBM SystemView features. It describes ways in which Systems Monitor can help in providing operational information and control in a distributed systems environment.

The functions provided by Systems Monitor for AIX Version 2 are described in detail, including the ability to distribute network polling away from a central NetView for AIX machine and improved monitoring of distributed AIX systems. This edition adds further details of the enhanced usability of Systems Monitor provided by the features of NetView for AIX Version 4.

This document is intended for use by people who are planning or implementing distributed systems and network management. Some knowledge of the Simple Network Management Protocol (SNMP) is assumed.

(330 pages)

Contents

Figures

Tables

Special Notices

This publication is intended to help systems administrators and programmers to plan, install and use Systems Monitor for AIX and the related IBM Systems Monitor products. The information in this publication is not intended as the specification of any programming interfaces that are provided by Systems Monitor for AIX. See the PUBLICATIONS section of the IBM Programming Announcement for Systems Monitor products for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Ave, Thornwood, NY 10594 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (VENDOR) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX	AS/400
IBM	NetView
Operating System/2	OS/2
RISC System/6000	RMONitor

The following terms are trademarks of other companies:

Cisco	Cisco Systems, Incorporated

DCE	The Open Software Foundation
HP	Hewlett-Packard Company
HP/UX	Hewlett-Packard Company
NCR	NCR Corporation
NFS	Sun Microsystems, Incorporated
Solaris	Sun Microsystems, Incorporated
Informix	Informix Software, Incorporated
Sun	Sun Microsystems, Incorporated
SunOS	Sun Microsystems, Incorporated
UNIX	X/Open Company Limited

Microsoft, Windows and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

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

C-bus is a trademark of Corollary, Inc.

Other trademarks are trademarks of their respective companies.

Preface

This document is intended to explain the capabilities of Systems Monitor for AIX and the family of associated IBM Systems Monitor products. The capabilities are illustrated by examples and scenarios, and options for extending the function of the product are examined. The majority of the document relates to Systems Monitor for AIX (that is, Systems Monitor Version 2). However, references to and examples of using Systems Monitor Version 1 are made where appropriate.

This document is intended for use by persons who are planning or implementing a distributed systems and network management environment.

How This Document Is Organized

The document is organized as follows:

- Chapter 1, “Introduction to Systems Monitor”

  This chapter describes the variety of functions performed by Systems Monitor, and shows how they may be used to build a comprehensive network and systems management structure.

- Chapter 2, “Installation Notes”

  This chapter describes some of the installation procedures, together with issues and gotchas that users need to be aware of.

- Chapter 3, “The System Information Agent”

  This chapter looks in detail at the data provided by the Systems Information Agent component of Systems Monitor for AIX in different system environments, and compares it with other SNMP-based system information.

- Chapter 4, “Using the Mid-Level Manager for Status Polling,” Chapter 5, “Breaking the Limits: Collections” and Chapter 6, “Other Functions of the Mid-Level Manager and System-Level Manager”

  These chapters look in detail at the mid-level manager capabilities of Systems Monitor for AIX and provides guidance on how best to exploit the capabilities it gives.

- Chapter 7, “Systems Monitor Examples”

  This chapter contains examples of how to set up and use Systems Monitor for AIX in typical systems management scenarios.

- Chapter 8, “Introduction to Agent Policy Manager” and Chapter 9, “Examples Using APM for Threshold and File Monitoring Tasks.”

  These chapters describe and show examples of using the Agent Policy Manager function of NetView for AIX Version 4.

Related Publications

The manuals listed below should be referred to for a more detailed discussion of the topics covered in this document.

- *Systems Monitor for AIX User's Guide,* SC31-7150

International Technical Support Organization Publications

These redbooks deal with the first version of Systems Monitor:

- *Examples of Using NetView for AIX V3,* GG24-4327
- *Examples of Using NetView for AIX V4,* SG24-4515

A complete list of International Technical Support Organization publications, known as redbooks, with a brief description of each, may be found in:

International Technical Support Organization Bibliography of Redbooks, GG24-3070.

To get a catalog of ITSO redbooks, VNET users may type:

```
TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
```

A listing of all redbooks, sorted by category, may also be found on MKTTOOLS as ITSOCAT TXT. This package is updated monthly.

How to Order ITSO Technical Publications

IBM employees in the USA may order ITSO books and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-445-9269. Most major credit cards are accepted. Outside the USA, customers should contact their local IBM office. For guidance on ordering, send a PROFS note to BOOKSHOP at DKIBMVM1 or E-mail to bookshop@dk.ibm.com.

Customers may order hardcopy ITSO books individually or in customized sets, called BOFs, which relate to specific functions of interest. IBM employees and customers may also order ITSO books in online format on CD-ROM collections, which contain books on a variety of products.

ITSO Redbooks on the World Wide Web (WWW)

Internet users may find information about redbooks on the ITSO World Wide Web home page. To access the ITSO Web pages, point your Web browser to the following URL:

```
http://www.redbooks.ibm.com/redbooks
```

IBM employees may access LIST3820s of redbooks as well. The internal Redbooks home page may be found at the following URL:

```
http://w3.itso.ibm.com/redbooks/redbooks.html
```

Subscribing to Internet Listserver

IBM redbook titles/abstracts are now available through Internet E-mail via the IBM Announcement Listserver. With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. All it takes is a few minutes to set up a profile, and you can get news (in ASCII format) from selected categories.

To initiate the service, send an E-mail note to:

announce@webster.ibmlink.ibm.com

with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

To obtain more details about this service, employees may type the following:

TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE

Note: INEWS users can select RelInfo from the action bar to execute this command automatically.

Acknowledgments

This document is the result of two residencies conducted at the International Technical Support Organization, Raleigh Center.

The advisors for the projects were:

Rob Macgregor	ITSO-Raleigh
Dave Shogren	ITSO-Raleigh

The author of this version of this document was:

Peter Glasmacher	IBM Germany

The authors of the previous version were:

Emma Locke	IBM UK
Yohichiroh Ishii	IBM Japan

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Jonathan Beakley
Richard Buckman
Jim Chou
Martha Crisson
Judith Dietz
Bob Gabor
Tom Hemp
Heather Kreger
Fred Niemi
of IBM Raleigh Networking Development staffs

Request for Feedback

Readers of this document are encouraged to feed back any information or comments regarding *any* of the material in this document. Please send your comments to:

Dave Shogren
ITSO-Raleigh
VNET: SHOGREN at WTSCPOK

or: IBM Corporation HZ8D/B678/D100
Attn: Dave Shogren or Rob Macgregor
Building 678 Rm D100
1001 Winstead Drive
Cary NC 27513

INTERNET: shogren@vnet.ibm.com
robmacg@vnet.ibm.com

Chapter 1. Introduction to Systems Monitor

This chapter provides an introduction to network management and the use of the Simple Network Management Protocol (SNMP), and then goes on to give an overview of Systems Monitor for AIX. You may want to go directly to the description of the different Systems Monitor for AIX versions. If so go to 1.2, “History of AIX Systems Monitor/6000” on page 11.

1.1 Network Management and Simple Network Management Protocol

Standards for Network Management in an IP network are defined in a series of Request for Comments (RFC) documents. These standards are commonly referred to as SNMP, but in fact Simple Network Management Protocol describes only the means by which management data is transported through the network. See Appendix G, “SNMP-Related Requests for Comment” on page 323 for more information about SNMP RFCs and how to get copies of them. Equally important are the standards that define the structure of the data being transported, the Management Information Base (MIB).

We now look at some of the elements of IP network management.

1.1.1 Managers and Agents

SNMP uses a client/server approach to management, defining two roles, manager and agent:

- The Manager (Client)

  This is where the network operators manage the overall network activity, using an application such as NetView for AIX to monitor and control the network. Critical traps are sent to this manager to alert network operators of problems in the network, and this manager also acts as the central point for storing and displaying statistics, (for example, performance data).

- The Agent (Server)

  The agent is responsible for reporting on and maintaining the data pertaining to a device, when the manager requests it to. Agents can run on several different types of managed nodes (for example, routers, hubs, servers, and workstations).

SNMP only defines a relationship between manager and agent; there is no concept of a manager-to-manager connection. In Systems Monitor, Version 2 a limited manager-to-manager relationship was introduced, between the main (central) manager and a Mid-Level Manager.

Mid-Level Managers are in fact hybrid entities. To the manager they appear as an agent, but to the agents under their control, they appear as a manager. They may be distributed throughout the network, and are responsible for performing systems management on specific nodes in their own area, thereby off loading some of the work and responsibility from the top-level manager. Thresholding and data collection can be performed, and the collected data is then sent to the manager, when requested, for storage and analysis. Filtering can also be performed to filter the data that is sent to the top-level manager.

Figure 1 on page 2 shows a hypothetical network, where the company has four main locations, in Seattle, New York, and London, with its headquarters in Raleigh, North Carolina.

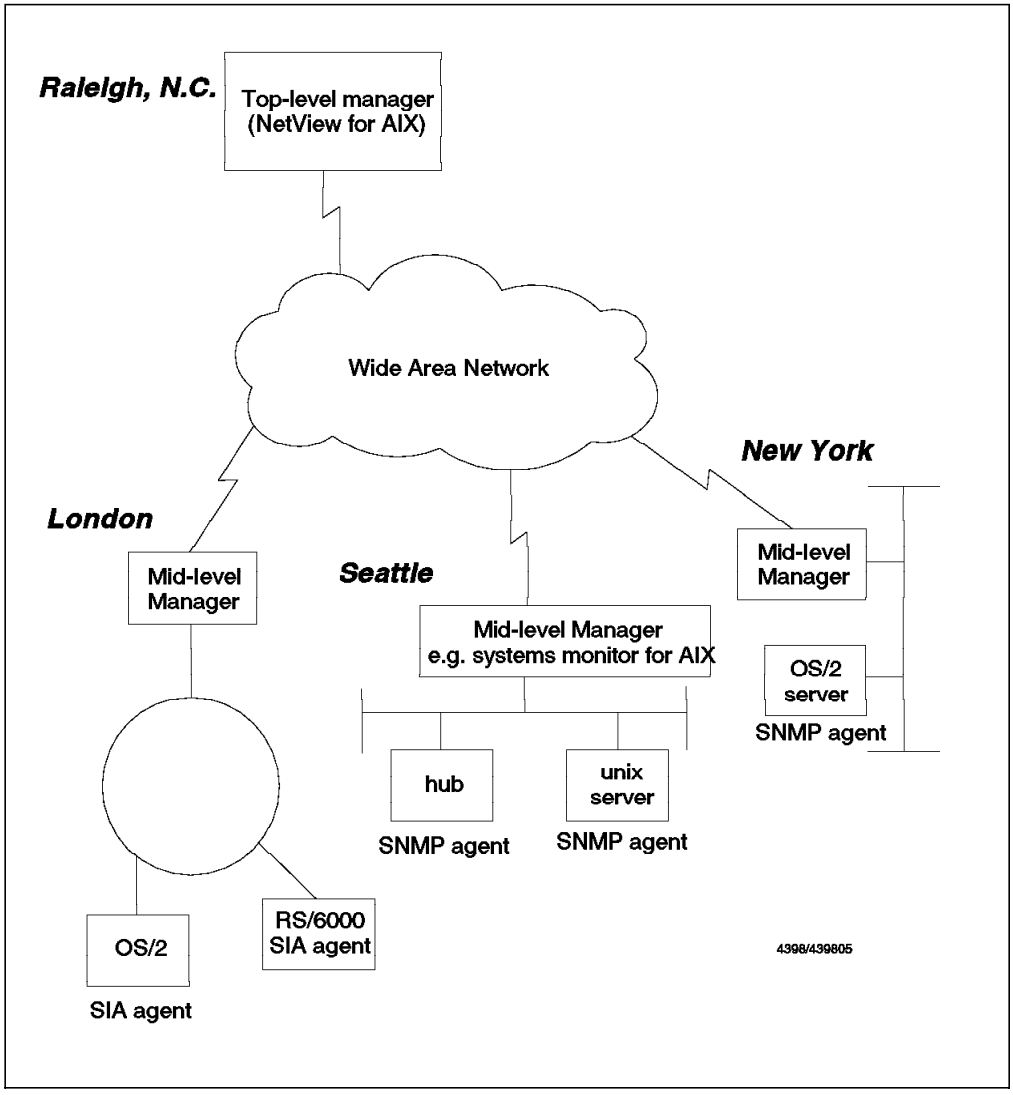


Figure 1. Distributing Managers and Agents

In this network, it can be seen that the management has been distributed with the top-level manager located in Raleigh, and Mid-Level Managers located in each of the other locations. This has the effect of reducing the amount of traffic that would otherwise flow across the WAN, since the Mid-Level Managers can perform local status monitoring, receive traps from all the nodes in their network, and only send critical information to the NetView for AIX node located in Raleigh.

1.1.2 Simple Network Management Protocol

Agents and managers that are running the TCP/IP suite of protocols use the Simple Network Management Protocol (SNMP) to communicate with each other, where SNMP is one of the application protocols that forms part of the TCP/IP suite of protocols. (Telnet for terminal emulation, and FTP for file transfer, are other examples.) Every agent supports a Management Information Base (MIB),

which is best described as a set of variables that represent the physical and logical resources of the managed systems or agent. The MIB is not a database, in the sense of a monolithic collection of data, but rather it represents *live* information. The values of the variables are maintained by different system functions such as the kernel, device drivers, and subsystems.

The manager can read and update these values via a limited number of requests which are listed below:

GET Requests the SNMP agent to retrieve the value of the specified variable and return it to the requester (or manager).

GET NEXT Requests the SNMP agent to retrieve the value of the next variable, after the one specified in the request, and return it to the manager. This is especially useful for retrieving tabular information, and multiple variables can be requested in it.

SET Requests the SNMP agent to update the value of a variable. This is not supported for all variables. For example, the MIB variable sysUpTime, which measures the time since the network management portion of the system was last reinitialized, is a read-only variable. This variable should never be changed by a systems administrator, as it is an internal measurement and will be updated automatically by system processes.

However the MIB variables SysContact and SysLocation are read/write MIB variables and can be changed via a SET command. These variables are likely to change if ownership of the machine changes or it is moved to an alternative location.

Notes

It is important to realize that the SNMP manager can only set one of the agents variables if it has read/write access to the MIB and is configured to use the correct community name. The community name is effectively a password and both the manager and the agent must be configured to use the same name. Otherwise the agent will refuse to act upon any of the manager's requests.

Instructions on how to configure community names will be described in section 2.2, "Configuring SNMP" on page 25.

The agent performs the following two functions:

- Responding to requests sent by the manager
- Notifying the manager when it experiences a problem

The SNMP manager will keep itself up-to-date on the status of the agents in the network by polling (that is, performing an SNMP get request on all the agents in the network, on a regular basis). However, if an agent experiences a problem, it will alert the SNMP manager to this effect by sending a trap. The trap does not necessarily describe the problem in any detail; it simply informs the manager that there is something amiss with the agent. The manager may then poll the agent via an SNMP GET or GET NEXT, to determine more details about the problem.

Simple Network Management Protocol defines six generic types of traps which include:

Cold Start	Generic trap 0
Warm Start	Generic trap 1
Link Down	Generic trap 2
Link Up	Generic trap 3
Authentication Failure	Generic trap 4
Egp Neighbor Loss	Generic trap 5
Enterprise Specific	Generic trap 6

Enterprise-specific traps will always have a generic trap number of 6, and a specific trap number that is specified by the originator. They also include an *enterprise ID* within the trap. This is a MIB variable that uniquely defines the agent that generated the trap. In this way different kinds of agents (for example, routers from two different manufacturers) may generate traps with the same specific ID. As long as the enterprise ID/specific ID combinations have been defined to the manager, it can distinguish between them.

Examples of creating enterprise-specific traps can be seen in Chapter 7, “Systems Monitor Examples” on page 161.

1.1.3 MIB Variables

The MIB variables used to represent resources on the managed systems are encoded as *managed objects* in a platform-independent way. This is achieved using an OSI data-description language called Abstract Syntax Notation 1 (ASN.1). The managed objects are organized hierarchically, where as you move downwards in the hierarchy the data represented by the object is more fully defined. Each *branch* in the hierarchy tree has a unique name and number, as shown in Figure 2 on page 5.

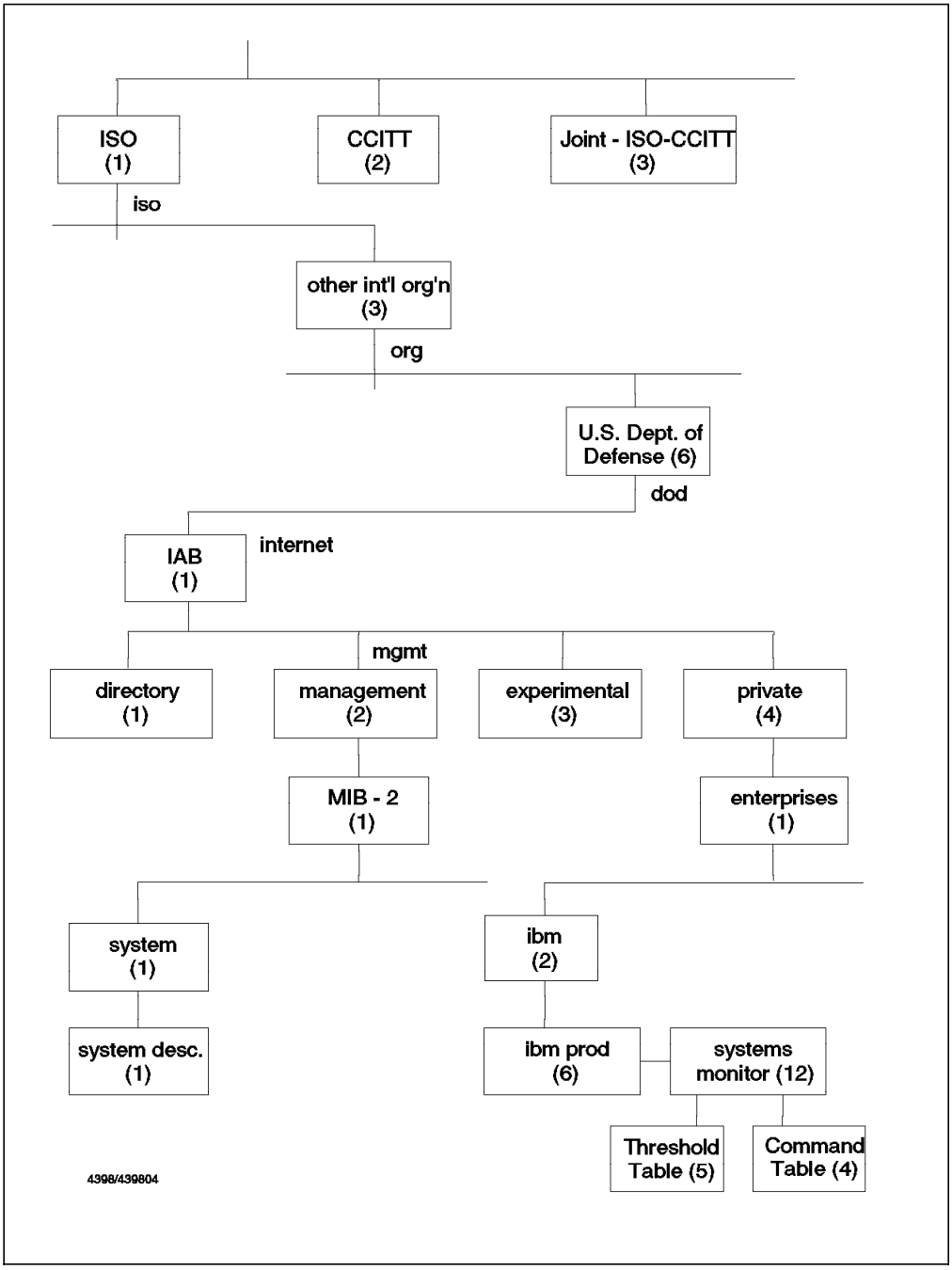


Figure 2. Object Identifier Tree

The Internet Engineering Task Force (IETF) has defined a standard set of MIB variables that all SNMP agents are required to support, which is referred to as MIB-2. MIB-2 was developed from MIB-1 (the original set of MIB variables) and is a superset of that earlier standard. Most SNMP agents available in the marketplace today support MIB-2, and any agent that claims to be MIB-2 compliant, *must* support all of the variables defined in the MIB-2 set.

From Figure 2 on page 5 it can be seen that the object ID of the top of the MIB-2 subtree may be written in dotted-decimal form as *1.3.6.1.2.1* and its name is *iso.org.dod.internet.mgmt.mib-2*.

The standard MIB-2 contains 171 objects that relate to aspects of IP network connectivity, and basic system information. These are system-independent variables, such as system contact name, system location, interface description and speed.

In most cases, however, we want to extend beyond this basic data structure. The following are the two ways in which such extensions are defined:

1. IETF-sponsored extensions

   These are MIB extensions that are defined in RFCs, and they usually describe some other network standard. For example, for LAN bridging there is a MIB extension described in RFC1493. Typically such extensions are inserted into the object tree just below the MIB-2 branch.

2. Private extensions

   In order to monitor machine-specific variables (such as paging space for a computer system, or slot assignments for a LAN hub) it is necessary to further extend the MIB. Manufacturers of different products have developed MIB extensions that are specific to their product. These are located off the *private* branch of the object tree (see Figure 2 on page 5).

Systems Monitor for AIX is an example of a private MIB extension written by IBM for the systems management of AIX and other machines. As for all objects found under the ibmProd branch of the tree, the Systems Monitor for AIX subtree starts with an object ID of *1.3.6.1.4.1.2.6.* and the name *iso.org.dod.internet.private.enterprises.ibm.ibmProd*.

The Systems Monitor for AIX MIB has an object ID of *1.3.6.1.4.1.2.6.12* and the name *iso.org.dod.internet.private.enterprises.ibm.ibmProd.systemsMonitor6000*

The best method of determining the object ID of a particular variable is to use the MIB Browser function of NetView for AIX. It is possible to move up and down the tree using the Up Tree and Down Tree buttons, and the dotted-decimal notation can be displayed by selecting a variable and then using the Describe button. The output of using Describe for the MIB-2 variable ifDescr is shown in Figure 3 on page 7.

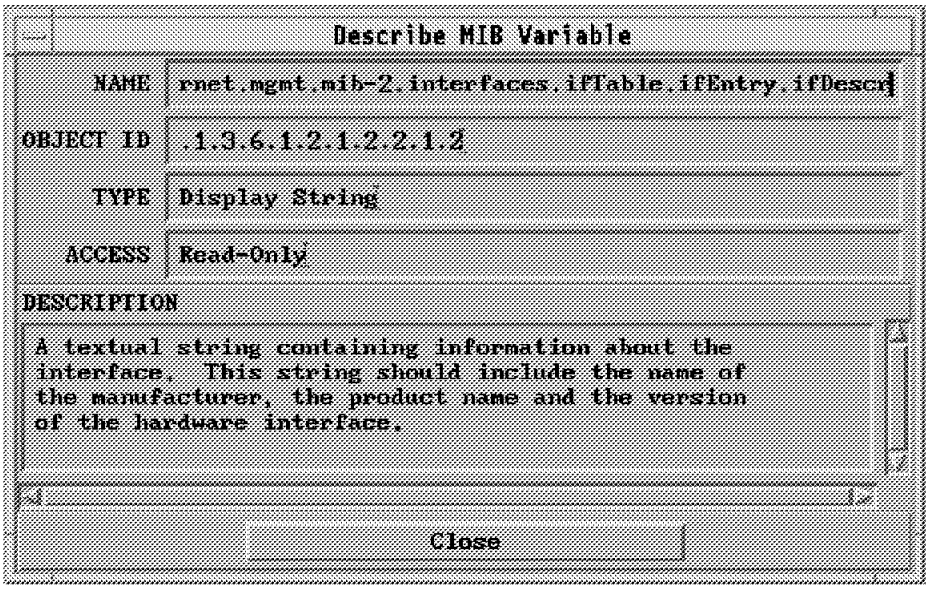


Figure 3. Using the MIB Browser to Determine the Object ID of a Variable. The object ID of the variable ifDescr is displayed here. This output also identifies whether the variable is read-only or read/write.

1.1.4 MIB Instances

Up to this point we have been discussing MIB *objects*; these may be thought of as generic information. For example, the object ifDescr that we viewed in the MIB Browser (Figure 3) represents the description of an interface. However, when we do an SNMP GET request to an agent, we want to retrieve the description for a specific interface, that is, an *instance* of the object type.

Every MIB variable we access through SNMP has an instance ID, which is appended to the object ID. For example, the MIB variable ifDescr (interface description), which has an ASN.1 number of 1.3.6.1.2.1.2.2.1.2, will return as many values as there are interfaces installed on that system. The example shown in Figure 4 on page 8 displays the results of querying the ifDescr variable using the MIB Browser function of NetView for AIX.

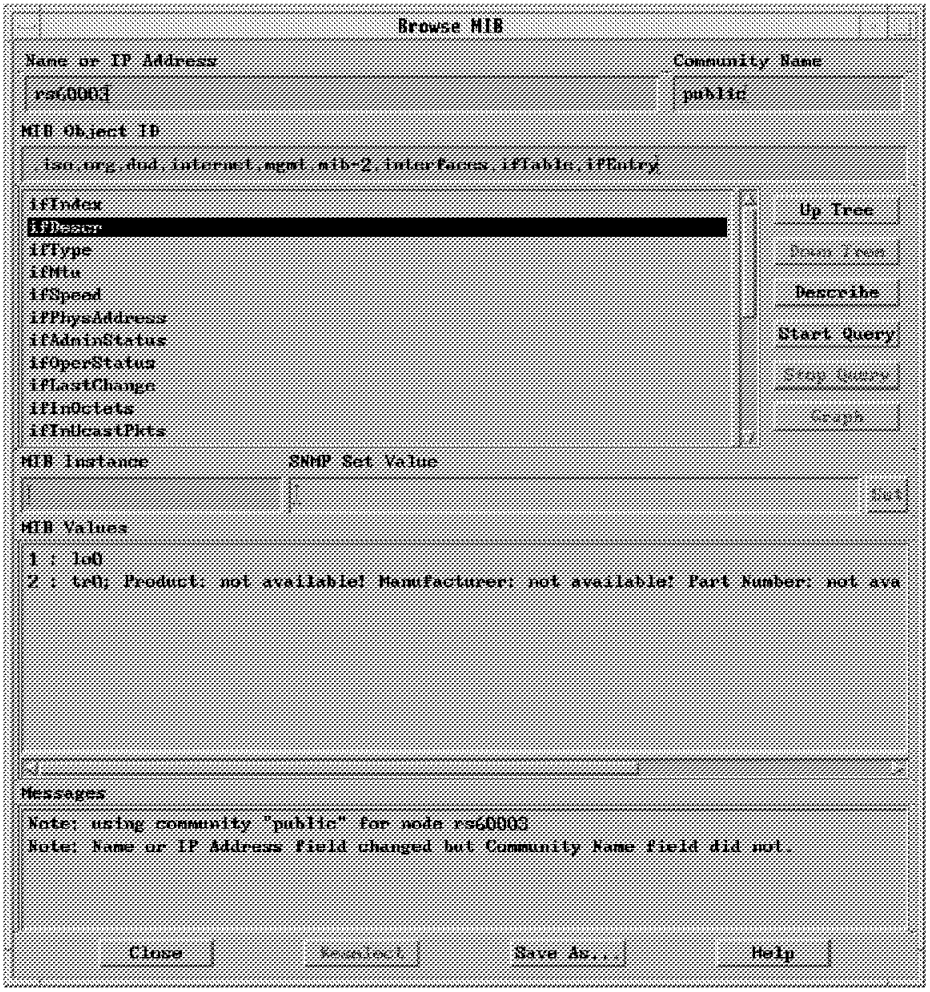


Figure 4. Output from Querying the Interface Description MIB Variable

The node that was queried has two interfaces: the internal *loopback* interface (lo0), and the token-ring interface (tr0). The numbers located to the left of the *":"* represent the MIB instance associated with the variable. Therefore the ifDescr MIB variable for the loopback interface can be accessed using the instance ID 1.3.6.1.2.1.2.2.1.2.1.

Even for variables that only have one value, we still access them using an instance ID. For example, the output of querying the variable sysContact indicates that the MIB instance is 0, as shown in Figure 5 on page 9.

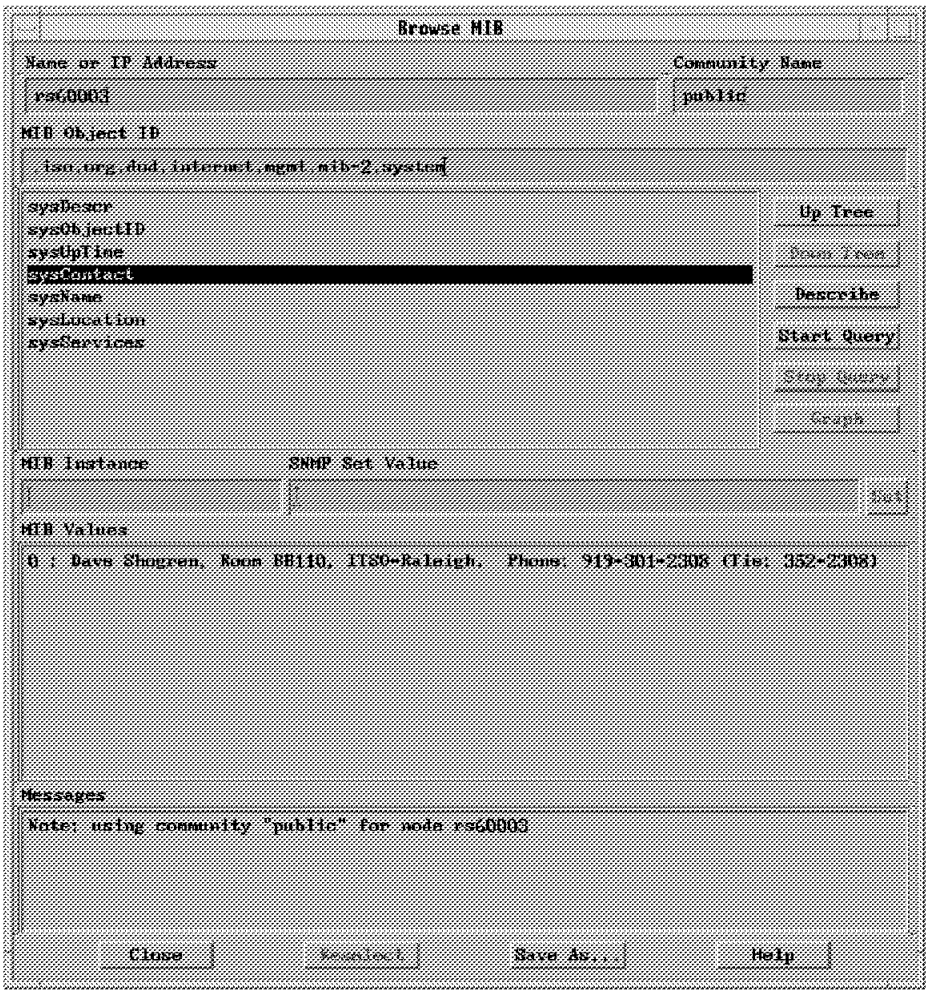


Figure 5. Output from Querying the System Contact MIB Variable

1.1.5 Querying and Setting MIB Variables

Within a manager machine running NetView for AIX, there are the following three ways to determine the value of (that is query) a MIB variable:

1. Use the MIB Browser function on NetView for AIX.

   To use the MIB Browser, select a MIB variable and then the Start Query button. This will return a value for all instances of the selected variable, as shown in Figure 4 on page 8.

   This is the simplest method.

2. The snmpget command, located in the directory /usr/OV/bin can be used.

   The format of the snmpget command is as follows:

   ```
   snmpget -c communityname node MIBvariable.instance
   ```

   As explained earlier, the community name must be the same on both the agent and the manager for the command to be successful. The node is the name of the agent that is being queried.

   The instance must be specified, and the command will return a value for the particular instance that was queried. For example, the snmpget command

required to determine the value for ifDescr for the loopback interface alone, would be as follows:

```
snmpget -c public rs60003 .1.3.6.1.2.1.2.2.1.2.1
```

If the instance is not specified then the snmpget command will fail; for example, entering `snmpget -c public rs60003 .1.3.6.1.2.1.2.2.1.2` will return the following message: `snmpget: This variable does not exist.`

3. The snmpinfo command located in the directory /usr/sbin can be used from both NetView for AIX and the Mid-Level Manager.

   The format of the snmpinfo command is as follows:

   ```
   snmpinfo -m get - c community -h hostname variable.instance
   ```

   As for the snmpget command, the MIB instance must be specified.

To set a MIB variable, use one of the following options:

1. Select a MIB variable, using the MIB Browser, then specify the MIB instance and SNMP set value, and then click on the SET button. The message box at the bottom of the screen will indicate whether or not the SET was successful, via the message SET was successfully completed.

2. Use the snmpset command, with the following format:

   ```
   snmpset -c communityname node MIB variable.instance type value.
   ```

   For example, to set the system location variable to ITSC, Raleigh, for node rs60004, with a community name of ITSC, issue the following command:

   ```
   snmpset -c ITSC rs60004 system.sysLocation.0 octetstringascii ITSC,
   Raleigh
   ```

3. Use the snmpinfo command as follows:

   ```
   snmpinfo -m set -c community -h hostname variable.instance=value
   ```

   There must be no space between the instance, equals sign, or the value to be set, or the command will fail.

   For example, the snmpinfo command to set the system location variable would be as follows:

   ```
   snmpinfo -m set -c ITSC -h rs60004 sysLocation.0=ITSC,Raleigh
   ```

The snmpinfo command uses the file /etc/mib.defs to determine which MIB variables it is authorized to set or get. Therefore in order to set non MIB-II variables, it is necessary to make sure that these variables are listed in the /etc/mib.defs file.

1.1.6 SMUX Protocol

We have discussed extensions to the default MIB tree, whether standard extensions or private ones. However, in order for a manager to be able to get and set these extended MIB variables, the agent also needs to be extended. One method to achieve this is to install a modified SNMP agent, which understands the extended MIB. An alternative is to use a *subagent* approach, where an interface is provided by the standard SNMP agent to allow additional code to register a MIB extension and handle requests for it.

One such subagent API is called the SNMP Multiplexing or SMUX protocol. This protocol is used by the SIA subagent to communicate with the SNMP agent running on the same node. The Systems Monitor for AIX Mid-Level Manager

and Systems Information Agent communicate with each other and with NetView for AIX using the SNMP protocol.

The SMUX protocol is described in detail in RFC 1227.

1.2 History of AIX Systems Monitor/6000

AIX Systems Monitor/6000 was first announced in 1993 and is now available in its third incarnation. In the past two years, AIX Systems Monitor/6000 was changed from a MIB extension for the management of system dependent attributes into a distributed monitor and SNMP manager.

1.2.1 AIX Systems Monitor/6000 Version 1

IBM Systems Monitor provides MIB extensions for the management of machine-specific attributes and is available on a range of different hardware platforms, including IBM′s RISC System/6000.

AIX Systems Monitor/6000 Version 1 provides MIB extensions for the management of machine-specific matters. It is available on different platforms including the following:

- HP-UX
- AIX (now superseded by Version 2)
- NCR UNIX
- OS/2 (as part of NetView for OS/2 and SystemView for AIX Release 2)
- SunOS
- Sun Solaris

The availability of AIX Systems Monitor/6000 across multiple platforms enables network managers such as NetView for AIX to monitor and collect information not provided by standard MIB and MIB-2 extension. In addition, AIX Systems Monitor/6000 provides thresholding, filtering and data collection capabilities. These features can help to off-load some of the work from the central network manager in order to optimize the bandwidth required for management tasks.

1.2.2 AIX Systems Monitor/6000 Version 2.1

AIX Systems Monitor/6000 Version 1 provides both threshold/filter capabilities and MIB extensions in one monolithic application. In AIX Systems Monitor/6000 Version 2.1 this single agent was split up into two separate agents by dividing the product into a management function and a system information function. There is a benefit to the end user in that the size of the executable code that provides the systems information is significantly reduced, and that distributed management can be provided at a lower cost.

AIX Systems Monitor/6000 Version 2.1 consists of the following three main functions:

- System Information Agent (SIA). It provides the system related MIB extensions in the form of an SNMP SMUX agent.

  The SIA provides specific management information about the machine on which it is installed. The MIB can be expanded to monitor files or execute commands on a remote node. Traps can be directed to a Mid-Level Manager instead of sending them directly to NetView for AIX.

- Mid-Level Manager (MLM). It is a distributed SNMP manager, taking over some of the polling function.

  The MLM collects information and traps from SNMP agents and Systems Monitor agents in its subnet. It examines and filters the collected information to decide whether or not to inform a top-level manager, such as NetView for AIX, about the results.

  Other significant functions of MLM are its polling and discovery capabilities. The MLM is able to poll its own subnet and report status changes to a higher level manager (NetView for AIX) via traps, thus reducing the required bandwidth by limiting status polls to local network segments. MLM is able to forward traps both via UDP or TCP. Choosing TCP guarantees a reliable delivery of critical information (refer to 1.3.1, “TCP Port 165 to Prevent Port Conflicts” on page 16).

  A network discovery function completes the MLM so it can act as a distributed network manager.

- CFG - Configuration application. Used to configure AIX Systems Monitor/6000 and retrieve results. The configuration application does not need NetView for AIX. Menu entries are integrated into SMIT or you can use the command line to start CFG. If NetView for AIX is also installed, NetView for AIX menu entries are available for starting CFG for a given node.

The Systems Information Agent provides systems management information relevant to the system on which it is installed, and sends this information to the Mid-Level Manager on request. The Systems Information Agent can also be configured to forward its traps to the Mid-Level Manager rather than NetView for AIX.

The Mid-Level Manager collects information and traps from the Systems Information Agent agents and other Systems Monitor agents in its subnet, and decides whether or not to send then to the top-level manager by applying filters. The Mid-Level Manager is also responsible for collecting statistics from the Systems Information Agent, and thresholding on these statistics. One of the most significant functions of the Mid-Level Manager is to perform status polling for the nodes in its own subnet, and informing NetView for AIX of node status changes via traps.

This is summarized in Figure 6 on page 13 which illustrates the different roles these network elements play in network management.

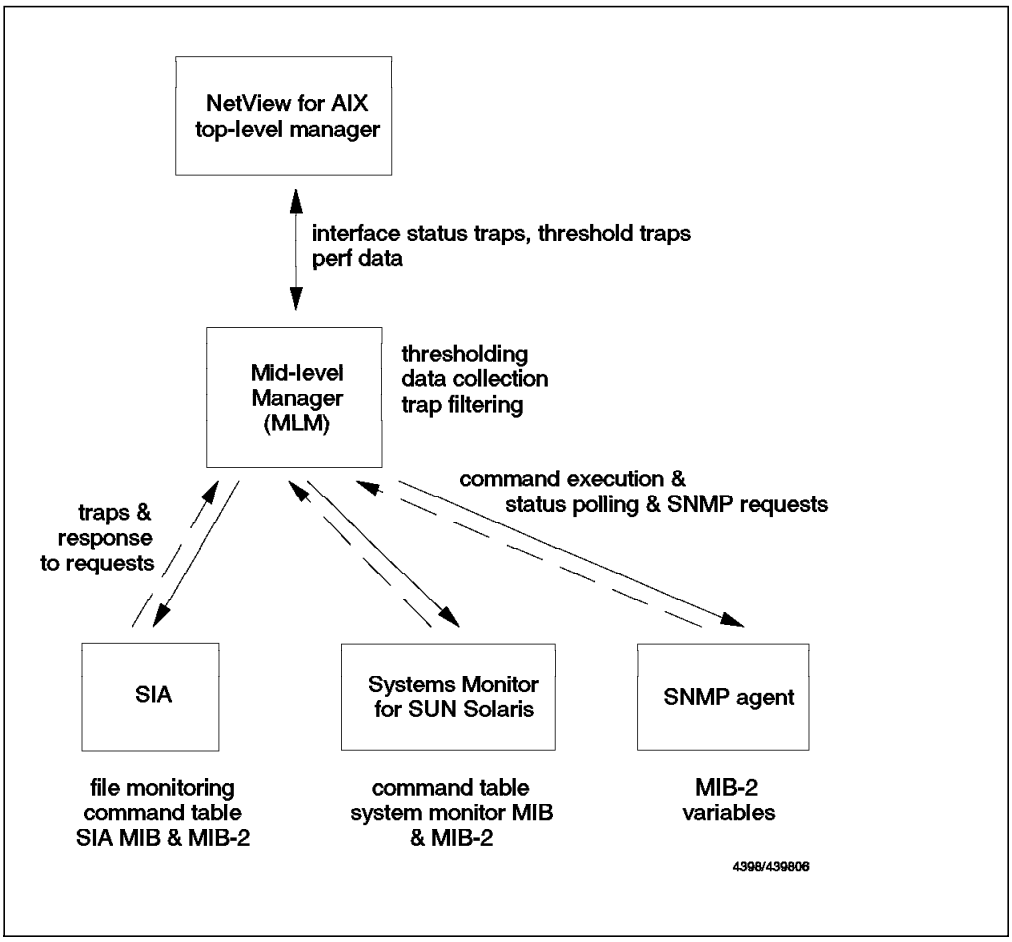


Figure 6. Mid-Level Manager and Systems Information Agent Responsibilities

1.2.3 AIX Systems Monitor/6000 Version 2.2

For the *classic* AIX Systems Monitor/6000 user, Version 2.2 is mostly a service update. One primary reason for Version 2.2 was the integration of AIX Systems Monitor/6000 into AIX Version 4.1. Be aware that AIX Systems Monitor/6000 versions prior to V2.2 do not work correctly with AIX V4.1.

Hint

If you execute the new IBM AIXLINK/X.25 Version 1.1 (5696-868) under AIX V3.2.5 or AIX V4.1, all of the X.25 MIBs you were able to retrieve from the old kernel X.25 support routines are gone. This affects the following MIB groups under the Systems Monitor MIB ...ibmProd.systemsMonitor6000. smSiaSystemInformation.smSiaSystemDevice.smSiaSystemDeviceX25

- smSiaSystemDeviceX25Installed
- smSiaSystemDeviceX25X25Table
- smSiaSystemDeviceX25X25RouteCount
- smSiaSystemDeviceX25X25RouteTable

Because of the new LPP, X.25 related things have been moved and, according to development, are no longer inside the AIX Kernel. Development is working to provide a solution.

Version 2.2 of AIX Systems Monitor/6000 introduces another agent, the System Level Manager (SLM). The SLM runs on a node where an SIA is installed and can be used for thresholding analysis and filtering of information on that local node. Thus, the SLM is similar to the MLM with the following exceptions:

1. Node discovery is not supported.
2. Status monitoring of remote nodes using the Status Table is not available since a Status Monitor Table is not provided by the System Level Manager.
3. Thresholds can only be set against MIB variables on the local node.
4. SLM listens to port 162 for traps only from its own node and, therefore will not filter and forward traps originating anywhere other than the local node.
5. Trap reception from other nodes is not supported.

   Again, SLM listens only to port 162 for traps from its own node. These traps, local to the System Level Manager, *can* be filtered and directed to other destinations in the network.

As with MLM, the System Level Manager is capable of delivering the local node's traps via UDP or TCP for reliable transport of critical information. In other words, SLM resembles the MLM without the ability to manage subnetworks.

Figure 7 shows the roles of the different members of the System Monitor agent family.

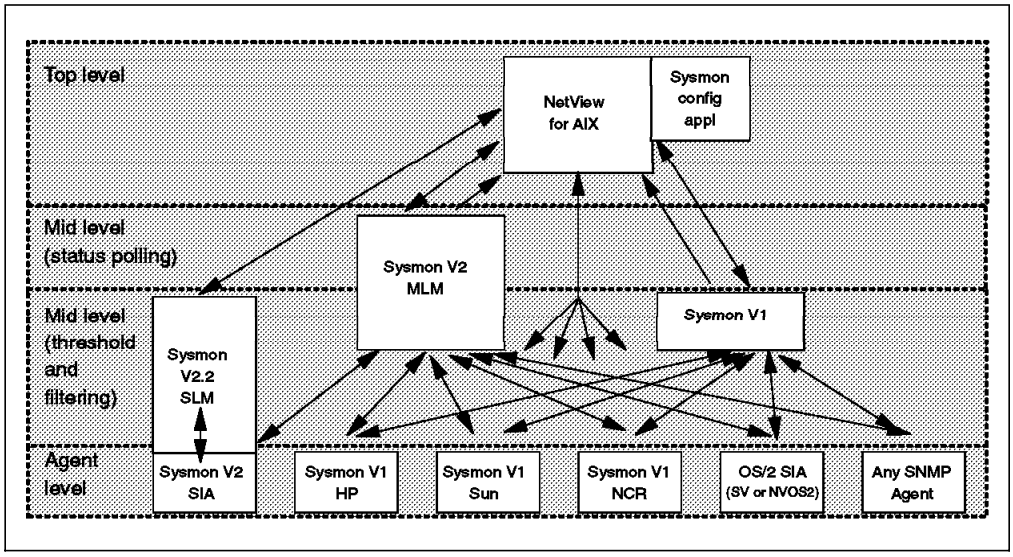


Figure 7. Systems Monitor Agent Roles. This diagram shows the capabilities of each Systems Monitor agent and how they are related to each other. All of the arrows represent SNMP communications.

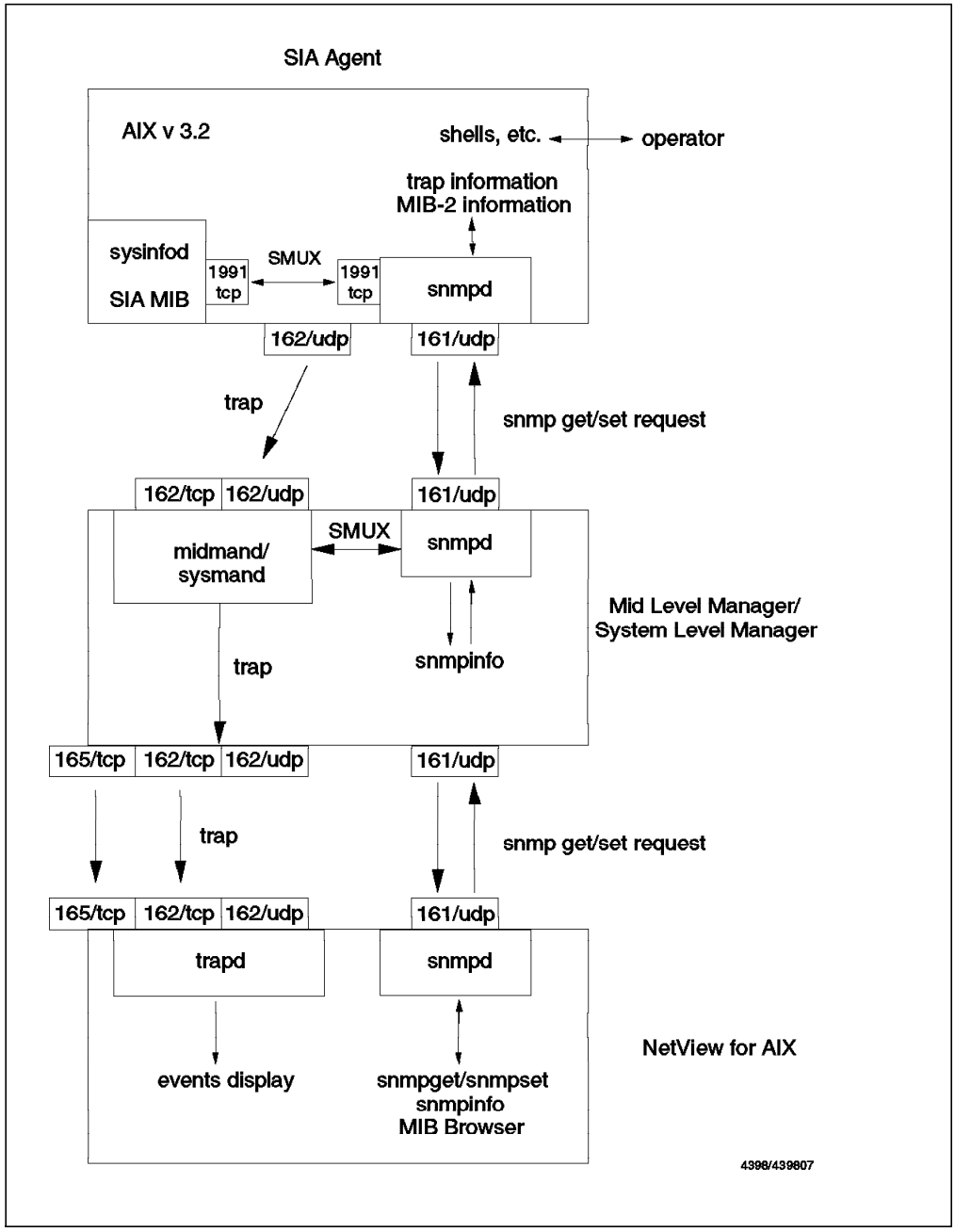


Figure 8. Systems Monitor Agent Responsibilities

Figure 8 shows the relationship between the various parts of a distributed management system using NetView for AIX and AIX Systems Monitor/6000. Note the new port 165 for reliable (TCP) trap delivery. Introduction of this additional port allows the execution of NetView for AIX and Mid-Level Manager on the same node. Configuring your environment to use this feature will be discussed in 1.3.1, “TCP Port 165 to Prevent Port Conflicts” on page 16.

1.3 New Functions in NetView for AIX Version 4 Related to AIX Systems Monitor/6000

NetView for AIX Version 4 implements a number of new functions that greatly improve the integration of AIX Systems Monitor/6000 into a true distributed network management implementation. The functions of &nvaixv4, which affect the operation of AIX Systems Monitor/6000 V2, are as follows:

- A new TCP Port 165 to deliver traps.
- An Agent Policy Manager (APM) to centralize Systems Monitor for AIX functions.
- An APM Console to redirect APM-specific traps to a separate workspace.
- An object collection facility to define groups of nodes that share common attributes.
- A ruleset editor which helps in defining complex filter criteria.
- New icons, which NetView for AIX Version 4 provides in its Root submap to access these facilities.

1.3.1 TCP Port 165 to Prevent Port Conflicts

Both Mid-Level Manager and NetView for AIX listen for traps on ports 162/TCP and 162/UDP. If you install MLM and NetView for AIX on the same workstation, problems will arise if the MLM is started first. The MLM issues a listen() subroutine call upon startup to install its trap receive server on the port 162/TCP to accept incoming TCP connections. Thus, NetView for AIX cannot use its trapd daemon to listen to the same port or trapd will fail. Several other daemons which depend on trapd will also fail to start up correctly. As a consequence, NetView for AIX will not start and exit with the error messages shown in Figure 9.

```
ERROR: The following REQUIRED daemons are not running:
actionsvr nvcorrd nvserverd ovactiond ovtopmd trapd
As root user, start all the daemons by using one of the following methods:
         a) use the smit (SMIT) utility
         b) issue the command:  /usr/OV/bin/ovstart
         c) rerun this command
```

Figure 9. Failing NetView for AIX Daemons

If you need to run both NetView for AIX and Mid-Level Manager or System Level Manager on the same workstation, you should change the port which is used by NetView for AIX to receive traps as discussed in 1.3.1.1, “Changing Trap Reception Port” on page 16.

1.3.1.1 Changing Trap Reception Port

During installation of NetView for AIX Version 4 the installation process adds a few new port definitions to the /etc/services file. Figure 10 on page 17 shows an extract of that file with the port definitions.

```
nvsecd          1663/tcp  # NetView Security daemon port
nvcorrd         1666/tcp  # NetView Correlation daemon port
actionsvr       1670/tcp  # NetView Correlation Action daemon port
nvcold          1664/tcp  # NetView Collection Facility port
nvsecltd        1667/tcp  # NetView Security client daemon
C5_server       1668/tcp  # NetView C5 Consolidated Console and Threshold
nvlockd         1669/tcp  # NetView General Topology Manager lock daemon
nvpagerd        1671/tcp  # NetView Pager daemon
snmp-trap       162/tcp   # NetView snmp-trap port
nvtrapd-trap    162/tcp   # NetView trapd monitor trap port
nvtrapd-trap    162/udp   # NetView trapd monitor trap port
nvtrapd-client 1661/tcp    # NetView trapd client application port
```

Figure 10. Extract from /etc/services

Port 162/TCP is normally defined and named *snmp-trap*. This is the port to receive traps over TCP. The nvtrapd-trap when installed by NetView for AIX uses the same port as the default trap-receive location. You can use SMIT to change this NetView for AIX trap-receive port number. Start SMIT using the `smit nv6000` fast path.

Next, select **Configure**, then **Set options for daemons**, then **Set options for event and trap processing daemons**, and then **Set options for trapd daemon**.

Now, you should be able to access a dialog similar to the one shown in Figure 11 on page 18. We chose port 165 to be the receiving port for traps delivered via TCP, by setting the field Port used to receive traps over TCP to be port 165.

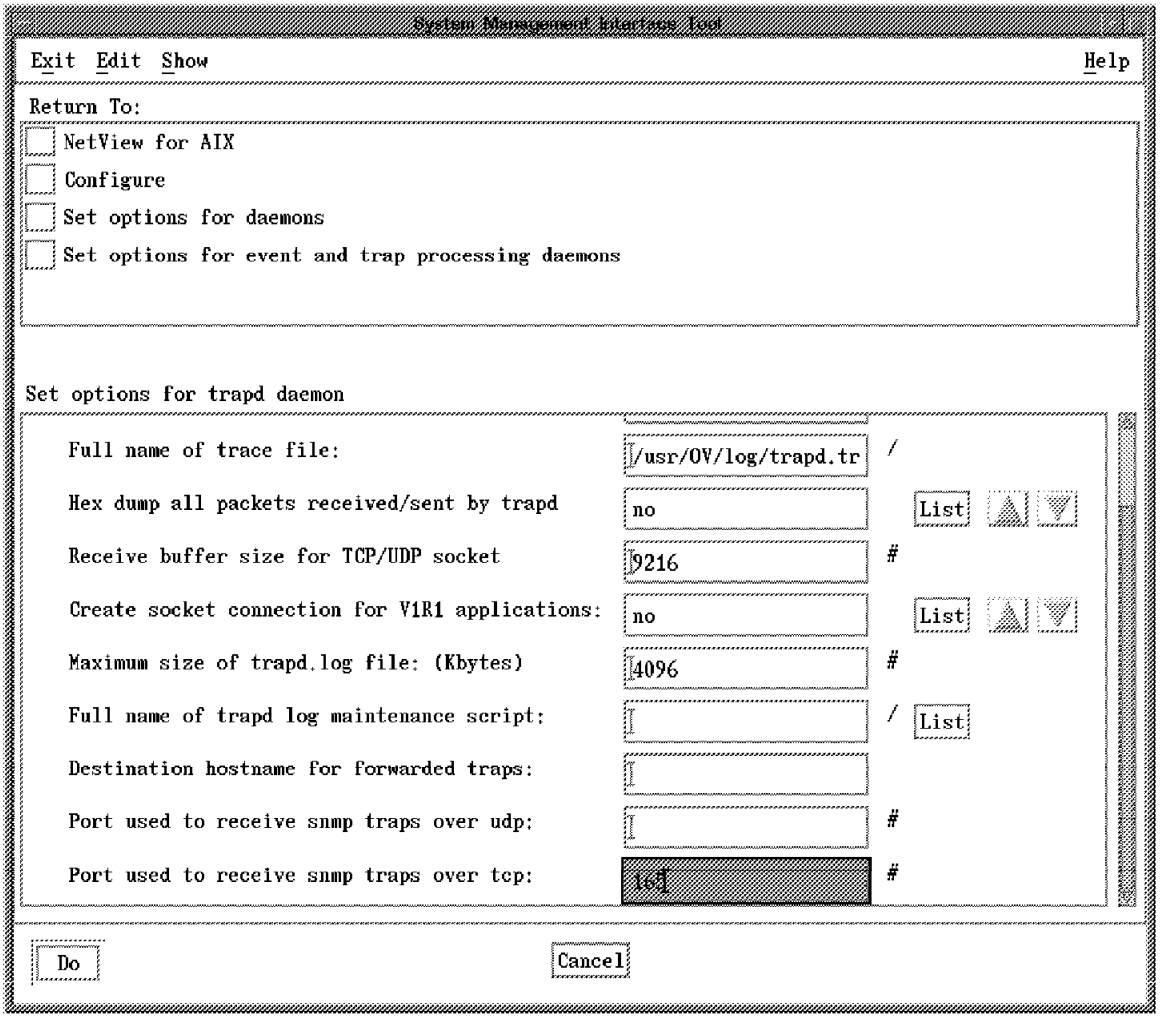


Figure 11. SMIT Dialog to Set the Trap Receiving Port

After successful completion, the /etc/services file will reflect the change as shown in Figure 12.

```
snmp-trap      162/tcp   # NetView snmp-trap port
nvtrapd-trap   162/udp   # NetView trapd monitor trap port
nvtrapd-client 1661/tcp   # NetView trapd client application port
nvtrapd-trap   165/tcp   # NetView trapd monitor trap port
```

Figure 12. Extract from /etc/services After Port Change

You could use any free port to deliver traps via TCP to NetView for AIX as long as both the /etc/services file and the Trap Destination Table of the affected Mid-Level Managers point to the same port. If you stop/start netmon or wait until the next NetView for AIX configuration check, NetView for AIX will update the MLM/SLM Trap Destination Table.

1.3.2 Agent Policy Manager in NetView for AIX Version 4

Systems Monitor V2 is a powerful application. Through the Mid-Level Manager (or MLM), you can off-load some systems management from NetView for AIX with the MLM′s thresholding capability. An MLM can identify a problem, report it to NetView for AIX, and even take some corrective action to solve that problem.

The System Information Agent, or SIA, has an extensible MIB that you can use to monitor not only machine-specific matters. The SIA MIB extensions provide variables to monitor applications and processes. A special SIA table, the File Monitor Table, allows you to monitor a file for a wide variety of conditions, and execute a command or a shell script when the condition occurs.

Systems Monitor is powerful, but requires some new skills and knowledge, especially in the arena of MIBs, community names and SNMP operation. The APM interface, running in conjunction with Systems Monitor and the Collection Facility, allows you to master these tasks more easily in the following ways:

- Simplifies the definition of threshold or file monitoring conditions by eliminating the need to define trap destinations.
- Enables you to distribute a threshold or file monitor configuration to a group of nodes in one single operation.
- Creates submaps and an icon for each submap on your NetView for AIX Version 4 Root submap for MLMs and their managed nodes.
- Creates icons on your Root submap that represent active threshold and file monitoring settings.
- Creates an icon on the node submap to reflect matching conditions on participating nodes. This icon further links to a tool called PDA to assist you in problem determination.
- Allows you to filter file monitor traps, limiting traps being forwarded to the AIX Control Desk to those in which you are interested.

You will find a more in-depth discussion of the Agent Policy Manager in chapter Chapter 8, “Introduction to Agent Policy Manager” on page 261 and in *Examples of Using NetView for AIX V4*, SG24-4515.

1.3.3 APM Console

The APM console works in conjunction with the Dynamic Workspace feature of NetView for AIX. It is a predefined filter you can use to direct File Monitor-related traps sent by the Mid-Level Manager into a separate workspace.

1.3.3.1 Activating the APM Console

The way to activate the APM Console is somewhat hidden in the system. You can activate this console from the NetView for AIX Control Desk via selecting **Create** and then **Dynamic Workspace**. The resulting dialog contains a button marked **Filter Activation**. Selecting this button brings up another dialog which contains yet another button named File List in its top row. Selecting **File List** results in Figure 13 on page 20.

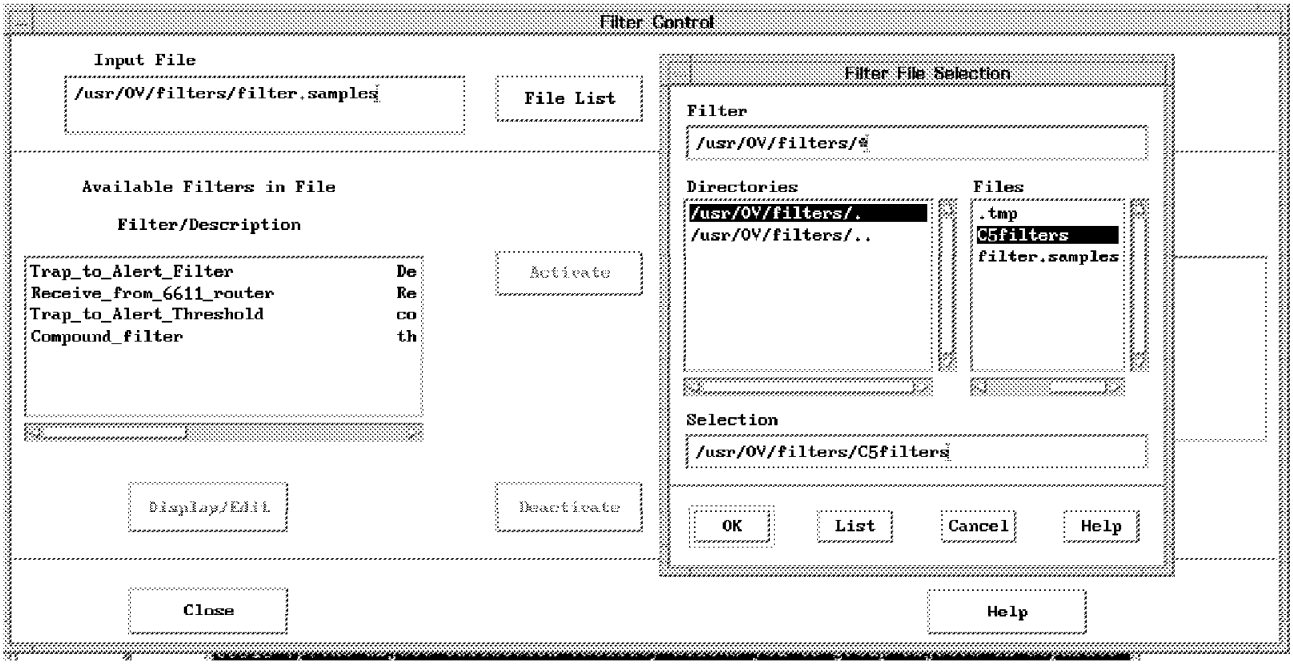


Figure 13. Heading Toward APM Console Filter

Selecting **C5filters** followed by the **OK** button and then selecting **APMConsole** followed by **Display/Edit** results in Figure 14 on page 20.

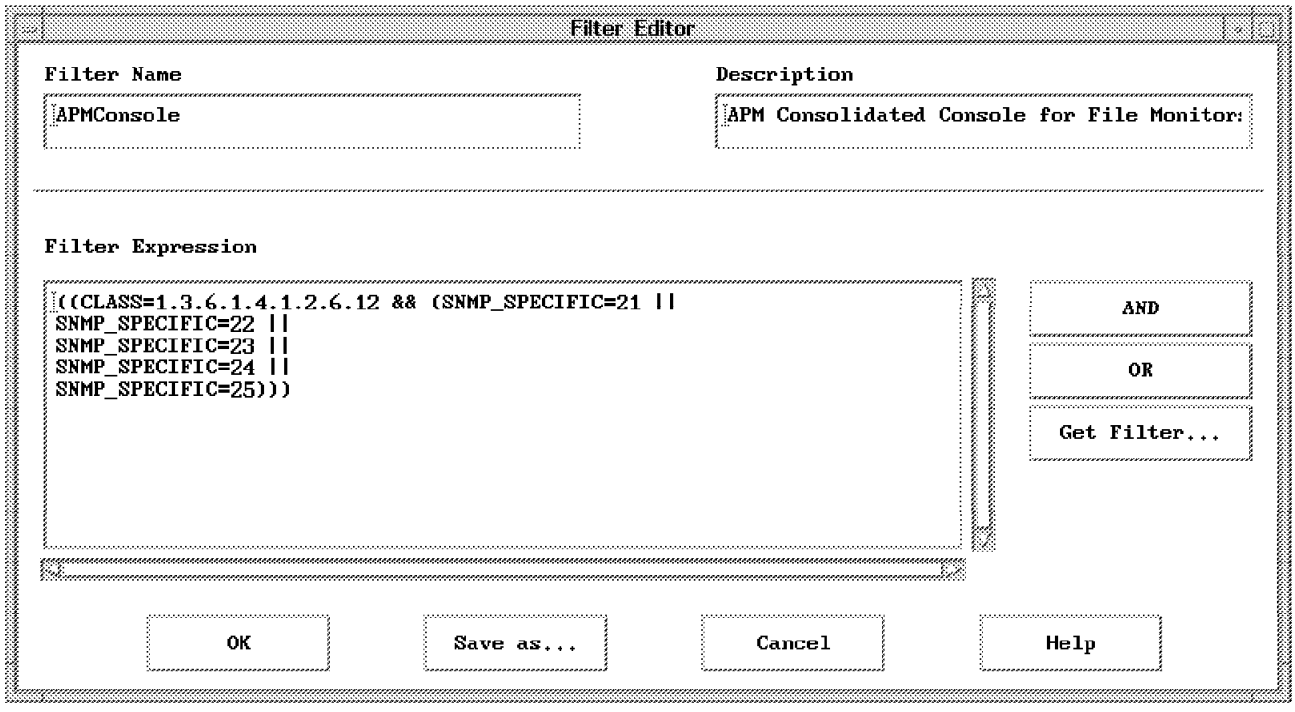


Figure 14. APM Console Definition

Activating this filter by selecting **Activate**, **Close**, and **OK**. This will result in a new dynamic workspace which will now show all File Monitor-related traps sent from the Mid-Level Managers in your network. This filter blocks all incoming traps except SM/6000 Traps 21 to 25 (refer to Appendix B, “Systems Monitor Traps” on page 307).

1.3.4 Object Collection Facility of NetView for AIX Version 4

The NetView for AIX object collection facility allows you to group objects in separate submaps. After grouping objects you are able to perform operations on the defined group of objects including the following:

- Collecting MIB data for defined collections.
- Easily checking the status of the objects in a collection.
- If systems monitor is present in the network, you can distribute configurations to all objects in a collection (refer to 1.3.2, “Agent Policy Manager in NetView for AIX Version 4” on page 19).

Once you define a collection, the status of all objects in that collection will be continuously updated by NetView for AIX. In case an object is found by NetView for AIX which matches the collection criteria, NetView for AIX will automatically add the object to the collection. If AIX Systems Monitor/6000 is running on machines in the network, NetView for AIX automatically detects these nodes and creates collections for the following:

- MLM nodes
- SIA nodes
- SLM nodes

In addition, NetView for AIX examines the nodes where an MLM has been detected and further creates a separate collection containing all the nodes managed by that particular MLM.

1.3.5 Ruleset Editor

The NetView for AIX Ruleset Editor allows the user to construct flow diagrams called rulesets. Access to the Ruleset Editor is via selecting the main NetView for AIX pull-down **Tools** followed by **Ruleset Editor**. This project did not use the Ruleset Editor. Examples can be found in *Examples Using NetView for AIX Version 4*, SG24-4515.

1.3.6 New Root Submap Icons

A number of additional symbols have been added for use on the NetView for AIX Version 4 Root submap.

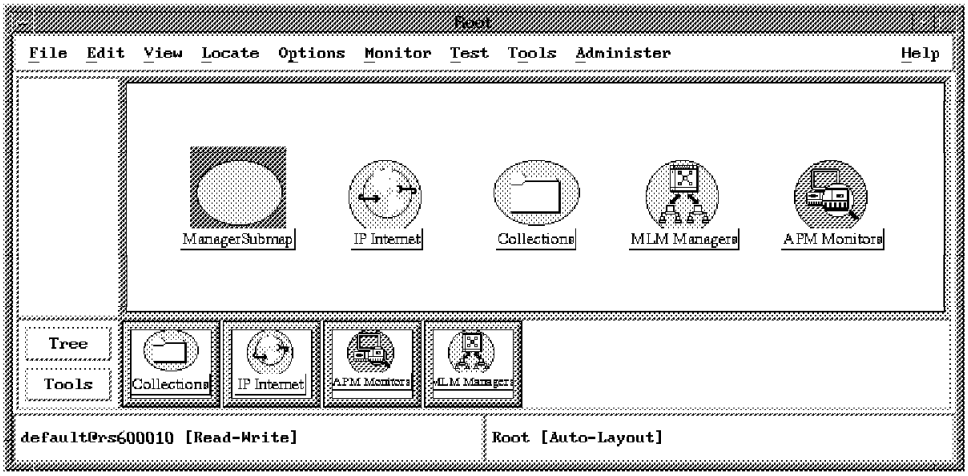


Figure 15. Some Icons in the NetView for AIX Root Submap

In most cases, icons on the Root submap lead to one or more submaps, providing the user with easy access to other applications and submaps.

Some of the new Root submap icons include:

ManagerSubmap Contains node symbols of all other NetView for AIX managers this node discovered in the network. The submap appears in row/column layout.

Collections Contains all defined and automatically created collections in a row/column layout.

MLM Managers Contains symbols of all nodes executing the Mid-Level Manager which were discovered in the network. The MLM nodes are displayed using Software:MLM symbols.

APM Monitors Contains symbols of objects defined as part of the File Monitor and Threshold collection support. The objects are displayed using Software:Collections symbols.

Chapter 2. Installation Notes

In this chapter we list the prerequisite hardware and software for Systems Monitor for AIX. We also describe how Systems Monitor for AIX uses the Simple Network Management Protocol and the configuration that is required for the different agents and managers to successfully communicate with each other. We also provide some information on installation procedures. We do not seek to describe a step-by-step installation process, but instead concentrate on areas that may be confusing and highlight pitfalls to avoid.

2.1 Prerequisites

This section discusses additional machine requirements but please refer to product documentation for complete information and guidelines.

2.1.1 Requirements for the Configuration Application

The Configuration Application is implemented as a graphical user interface and requires the X-Windows System Version 11 Release 5 or later and Motif 1.2 or later. In addition to this, the Configuration Application requires the following:

- No additional memory
- A TCP/IP connection
- 25 MB of free disk space

NetView for AIX is not required in order to execute the Configuration Application. You can start it from an AIX Window Terminal by entering `smconfig`. If NetView for AIX is present on the node where you are going to install the Configuration Application, the installation process will integrate the Configuration Application into the NetView for AIX Menu Tree. Additionally, it loads the relevant Systems Monitor MIBs, ibm-midlevelmgr.mib and ibm-sysinfo.mib, into the MIB database of NetView for AIX and updates the /etc/mib.defs file with Systems Information Agent and Mid-Level Manager definitions. To allow NetView for AIX to process traps sent by Systems Monitor agents, it also updates /usr/OV/conf/C/trapd.conf to reflect the AIX Systems Monitor/6000 traps. Appendix B, “Systems Monitor Traps” on page 307 lists the AIX Systems Monitor/6000 traps.

2.1.2 Requirements for the Mid-Level Manager

In addition to the general prerequisites, the Mid-Level Manager requires the following:

- No additional memory
- A TCP/IP connection
- 7 MB of free disk space

The Mid-Level Manager requires additional space in the /var file system for its configuration files and, of course, for its log files. Figure 16 on page 24 shows configuration of the MLM midmand.log file via the smconfig panel and selection of **MLM Program Logging**.

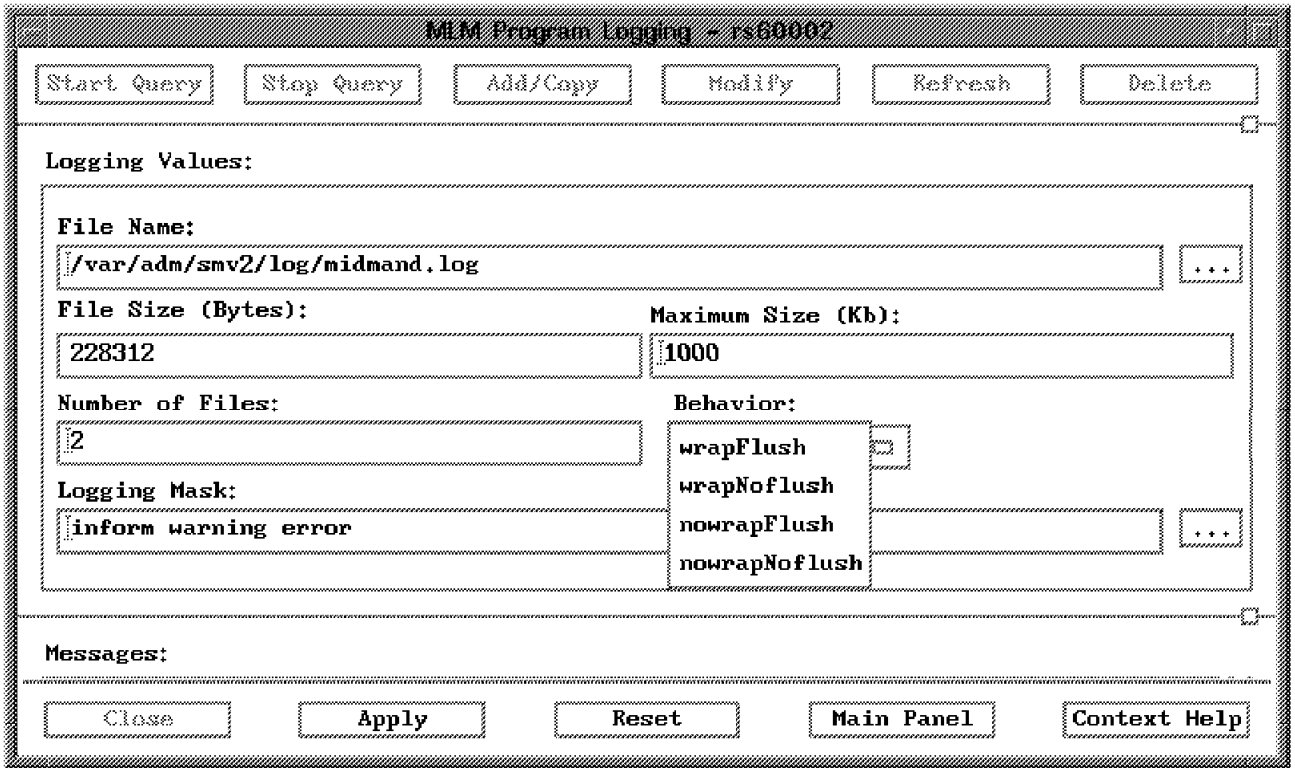


Figure 16. Mid-Level Manager Logfile Configuration

It is a good idea to plan an additional 5 to 7 MB of disk space for logging purposes. By default, the Mid-Level Manager logging facility creates the following two log files under /var/adm/smv2/log:

midmand.log The main log file. It grows to the maximum amount of bytes specified in the MLM Program Logging Table (see Figure 16).

midmand.log1 The second default log file. As soon as midmand.log reaches its configured limit, Mid-Level Manager renames the file to midmand.log1 and opens a new midmand.log file.

Depending on the configured behavior, the oldest file will be removed when the maximum is reached or logging will stop.

The term *flushing* refers to how the log entries are written to the log file. If flushing is active, each log entry is immediately written to the log file. Flushing set to off causes Mid-Level Manager to write log records in larger chunks to the log file, perhaps saving some CPU time.

2.1.3 Requirements for the System Level Manager

In addition to the general prerequisites, the System Level Manager requires the following:

- No additional memory
- A TCP/IP connection
- 5 MB of free disk space

As said for the Mid-Level Manager, the System Level Manager requires additional space in the /var file system for its configuration and log files. You

should plan a reasonable amount of logging space for the System Level Manager. By default, the Mid-Level Manager logging facility creates two log files under /var/adm/smv2/log (sysmand.log and sysmand.log1). The same rules as mentioned in 2.1.2, “Requirements for the Mid-Level Manager” on page 23 apply for the System Level Manager. You can configure the System Level Manager logging behavior via the System Level Manager Program Logging Table.

2.1.4 Requirements for the Systems Information Agent

In addition to the general prerequisites, the Systems Information Agent requires the following:

- No additional memory
- A TCP/IP connection
- 6 MB of free disk space

The Systems Information Agent creates two default log files (sysinfod.log andsysinfod.log1) under /var/adm/smv2.

You can configure the logging capabilities of the Systems Information Agent in the same way as with the other agents via its appropriate SLM Program Logging Table.

2.2 Configuring SNMP

It is necessary to configure SNMP on the agents and the manager before anything useful can be done. Most SNMP agents and managers default to using a community name (or password) of *public*. If no changes are made to the configuration then this is the community name that the manager will use every time it issues a request to the agent. In most circumstances this is sufficient and no changes need to be made to the SNMP configuration; however, when using Systems Monitor for AIX changes will need to be made. This is because by default the community name of public provides read-only access to the MIB, and hence it is not possible to change any of the MIB variables via SET commands. In order to configure Systems Monitor for AIX it is necessary to perform SET commands. Therefore, read/write access to the MIB is a prerequisite.

Read/write access to the MIB can be provided in one of the following two ways:

- Change the *agent* so that the community name of public provides read/write access.
- Change the *manager* to use a new community name which has read/write access to the agent.

In our configuration we have chosen to follow the latter approach, and we are using a community name of ITSC. This community name must be configured on both the manager and the agent, and instructions on how this can be achieved are detailed below.

2.2.1 SNMP Configuration for SNMP Agent

The location of the SNMP configuration file will be different depending on the system involved, for example:

- On the RISC System/6000, the file is called /etc/snmpd.conf.
- On OS2, the source is modified using the **Configuration** icon in the NetView for OS/2 folder on the OS/2 desktop.

Both NetView for AIX and the Mid-Level Manager, although they are SNMP managers, also have SNMP agents that must be configured.

2.2.1.1 SNMP Configuration for RS/6000 Agent

The contents of the snmpd.conf file on our RISC System/6000s are shown below:

```
logging         file=/usr/tmp/snmpd.log         enabled
logging         size=0                          level=0
community       public
community       ITSC    9.24.104.0      255.255.255.0 readWrite
community       private 127.0.0.1 255.255.255.255 readWrite
community       system  127.0.0.1 255.255.255.255 readWrite 1.17.2
view            1.17.2          system enterprises view
trap            public          9.24.104.27     1.2.3   fe    # rs60004
#snmpd         maxpacket=1024 querytimeout=120 smuxtimeout=60
smux            1.3.6.1.4.1.2.3.1.2.1.2         gated_password  # gated
smux
```

Figure 17. Snmp Configuration for RS/6000 Agent on Subnet 9.24.104.0

The entry for community name ITSC will allow any SNMP manager in the 9.24.104 IP subnetwork read/write access to this SNMP agent. Access can be restricted to specific nodes by specifying the IP address of the node rather than the IP address of the subnet.

For example, the entry that would be added to only permit rs60002 read/write access to this agent would be as follows:

```
community       ITSC          9.24.104.28    255.255.255.255  readWrite
```

All other managers are restricted to using a community name of public, with read-only access. Read/write access is permitted only if it is explicitly stated in the configuration file.

The trap statement is also important, because this tells the SNMP agent where to send traps. In the configuration shown above, the agent is configured to send traps to rs60004, the Mid-Level Manager. This should be the trap statement that is used for all agents on the same subnet as the Mid-Level Manager, with the exception of the Mid-Level Manager node itself, which has a trap statement set to its own loopback address (the MLM code itself takes care of routing traps to the NetView for AIX node).

Note

Every time the SNMP configuration file is changed, the snmpd daemon must be refreshed for these changes to take effect. This can be done either by stopping and restarting it, or by using the AIX System Resource Controller `refresh -s snmpd` command.

2.2.1.2 Configuration for Mid-Level Manager

The configuration shown in Figure 18 shows the snmpd.conf file for the node rs60004, before MLM was installed.

```
logging             file=/usr/tmp/snmpd.log       enabled
logging             size=0                        level=0
community       public
community       ITSC    9.24.104.0   255.255.255.0 readWrite
community       private 127.0.0.1    255.255.255.255 readWrite
community       system  127.0.0.1    255.255.255.255 readWrite 1.17.2
view            1.17.2          system enterprises view
trap        public       9.24.104.28     1.2.3   fe # rs60002
#snmpd  maxpacket=1024 querytimeout=120 smuxtimeout=60
smux    1.3.6.1.4.1.2.3.1.2.1.2         gated_password     # gated
smux      1.3.6.1.4.1.2.6.4.1           nv6000  # NetView for AIX:
trapgend
```

Figure 18. snmpd.conf before Mid-Level Manager Install

It can be seen from this configuration that the node is sending traps to the NetView for AIX node, rs60002. When the Mid-Level Manager software is installed on a node, it comments out any trap statements that it finds in /etc/snmpd.conf. It also adds a trap statement for its loopback interface, together with a statement, warning that trap destinations should be configured in the MLM Trap Destination Table, and not /etc/snmpd.conf, as shown in Figure 19.

An entry will also be added to the Mid-Level Manager's Trap Destination Table, to send traps to the node that was specified in the original /etc/snmpd.conf trap statement, which in this case was rs60002.

```
logging   file=/usr/tmp/snmpd.log  enabled
logging   size=0                   level=0
community        public
community        ITSC   9.24.104.0 255.255.255.0 readWrite
community       private 127.0.0.1  255.255.255.255 readWrite
community       system  127.0.0.1  255.255.255.255 readWrite 1.17.2
view            1.17.2          system enterprises view
# trapsmv2    public       9.24.104.28     1.2.3   fe # rs60002
#snmpd   maxpacket=1024 querytimeout=120 smuxtimeout=60
smux    1.3.6.1.4.1.2.3.1.2.1.2    gated_password   # gated
smux      1.3.6.1.4.1.2.6.4.1            nv6000  # NetView for AIX: trapgend
smux      1.3.6.1.4.1.2.6.12             sm6000  # Systems Monitor for AIX: sysinfod
# Systems Monitor (MLM) is installed, trap destinations should be
# configured vai the MLM's Trap Destination Table - NOT IN THIS FILE.
trap      public    127.0.0.1   1.2.3    fe         #loopback

snmpd      maxpacket=16000 smuxtimeout=60       # Systems Monitor for AIX
```

Figure 19. snmpd.conf after Mid-Level Manager Install

2.2.1.3 Configuration for NetView for AIX

The snmpd.conf file for the node running NetView for AIX is shown in Figure 20.

```
logging         file=/usr/tmp/snmpd.log         enabled
logging         size=0                          level=0
community       public
community       ITSC    9.24.104.0      255.255.255.0 readWrite
community       ITSC 127.0.0.1 255.255.255.255 readWrite
view            1.17.2          system enterprises view
trap            public          9.24.104.27  1.2.3   fe    # rs60004
#snmpd          maxpacket=1024 querytimeout=120 smuxtimeout=60
smux            1.3.6.1.4.1.2.3.1.2.1.2         gated_password  # gated
smux
```

Figure 20. Snmp Configuration for RS/6000 Agent on Node Running NetView for AIX

The important difference for the NetView for AIX node is that the community name for the loopback interface must be the *same* as the community name of the LAN interface, which in this network is token-ring.

If the Mid-Level Manager is installed on a node in the same network as NetView for AIX, then you can choose where to send the traps. Either send them to the Mid-Level Manager node, and specify the IP address of that node in the trap statement, as shown above, or specify the loopback address. In the latter case traps will be sent directly to NetView for AIX, via the loopback interface, and will never go out on the LAN. The difference is that an SNMP agent will always send traps using the UDP protocol; therefore, if the trap statement is set to loopback, traps will be sent to NetView for AIX using UDP. However, if the trap statement is set to rs60004, traps will be sent to NetView for AIX using TCP (due to the capability of the Mid-Level Manager to send traps using the TCP protocol).

2.2.2 SNMP Configuration for SNMP Manager

So far we have been discussing the *agent* configuration defining which manager to accept SNMP requests from, the community name to expect, and the destination for traps. There is equivalent configuration to be done at the *manager* end. This is to define which community name to use for which agent.

The Mid-Level Manager, the System Level Manager and NetView for AIX are all SNMP managers, and they will need a configuration file to determine the community name to include in SNMP GETs and SETs. Note that in the special case of the System Level Manager the configuration can be very simple because it only needs a definition for the SNMP agent on its own node (using the loopback address, 127.0.0.1). The manager configuration file is called ovsnmp.conf, and is located in the /usr/OV/conf directory on a NetView for AIX node. On a Mid-Level Manager node it will either be located in /usr/OV/conf or in the /var/adm/smv2 directory.

When the Mid-Level Manager is installed on a node that has NetView for AIX installed, it will use the configuration file in /usr/OV/conf. If the SMIT remote install option is taken on this node, to install the Mid-Level Manager on a remote node, this configuration file can be copied across (using the option to ″Install with Community File″). The ovsnmp.conf file will then be stored in the /usr/OV/conf directory on the Mid-Level Manager node. This directory will be created during the installation process.

If the Mid-Level Manager is installed on a node without NetView for AIX installed, it will be necessary to either copy the ovsnmp.conf file across from a NetView for

AIX node, or create a new one. The latter approach is not recommended, as the syntax of the file is complex, and creating it from scratch could lead to errors. There is an option in SMIT to allow the remote installation/update of ovsnmp.conf to /var/adm/smv2. This is where the Mid-Level Manager will keep its own copy of ovsnmp.conf. When the Mid-Level Manager daemon is started, it will look for the configuration file in /usr/OV/conf first, if it cannot find the file it will look for /var/adm/smv2/ovsnmp.conf.

2.2.2.1 SNMP Configuration File for SNMP Manager (NetView for AIX and Mid-Level Manager)

The format of the configuration file is the same for both the Mid-Level Manager and NetView for AIX. The contents of the file located on rs60002, our NetView for AIX node, is shown in Figure 21.

```
127.0.0.1:ITSC:*:8:3:300::ITSC:
rs60002.itso.ral.ibm.com:ITSC:*:8:3:300::ITSC:
rs60004.itso.ral.ibm.com:ITSC:*:8:3:300::ITSC:
mlmnode:ITSC:*:8:3:::ITSC:
*.*.*.*:public::8:3:300:::
```

Figure 21. SNMP Configuration for NetView for AIX Manager

The first entry in the file is for the NetView for AIX node's loopback interface. If a different community is used for the token-ring interface, then the same community name must be used for the loopback interface. There are three entries in the file for the nodes rs60002, rs60004 and mlmnode, telling the manager to use a community of ITSC when issuing requests to these nodes. The final line in the file instructs the manager to use the community name public when communicating with all other nodes.

The manager can easily be configured by selecting the option **SNMP Configuration** from the Options field on the menu bar, entering the appropriate parameters, and then selecting the **Add** button. This action will update the ovsnmp.conf file. This file *should not* be edited any other way.

It is simple to determine whether or not the agent and manager have been configured correctly, by checking to see if any authentication failure traps appear at the NetView for AIX console. This indicates that the configuration on the manager and the agent do not match.

2.3 How sysmon Uses SNMP

Systems Monitor for AIX presents some special challenges for configuration, because of the fact that it plays several different SNMP roles as follows:

1. All of the agent components are SNMP agents, which respond to SNMP GET and SET requests from both NetView for AIX and the end-user interface.

   This is illustrated in Figure 22 on page 30.

2. The MLM and SLM components also act as SNMP managers, sending SNMP GET requests.

   This is illustrated in Figure 22 on page 30.

3. The MLM and SLM can be a source for SNMP traps, sending them to NetView for AIX.

4. The MLM also acts as a receiver of traps for filtering purposes.

   The MLM and SLM can receive traps from Systems Information Agent agents in the network, and will then apply filter rules, so that only the critical traps are sent to the top-level manager (that is, NetView for AIX). This is illustrated in Figure 23.

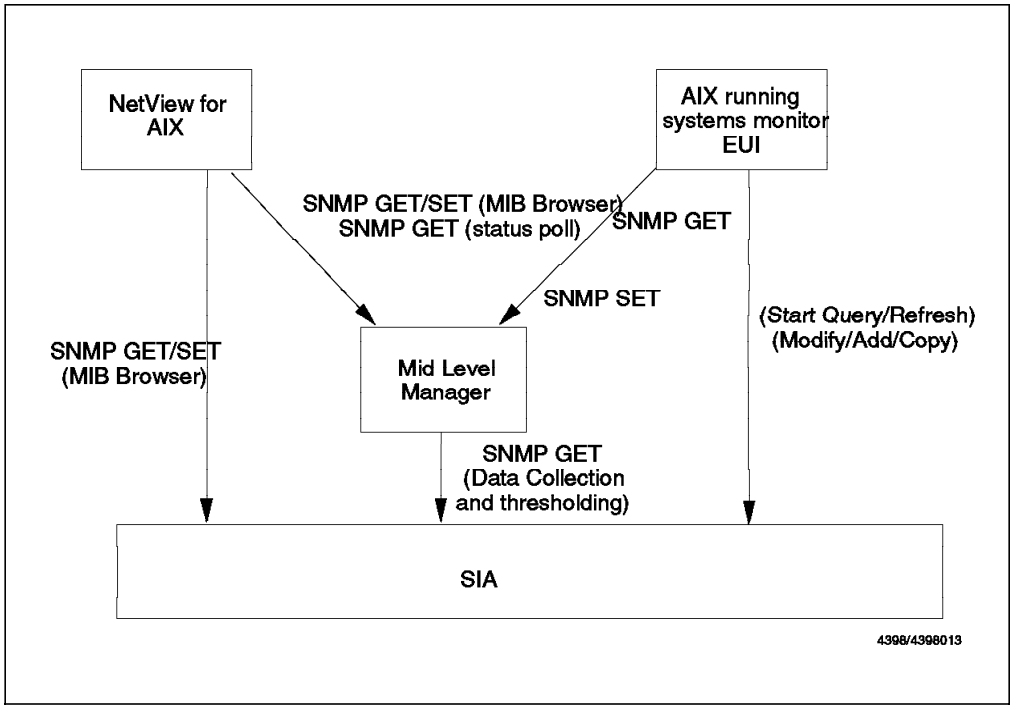


Figure 22. Mid-Level Manager Acting As Sender and Receiver of SNMP GETs and SETs

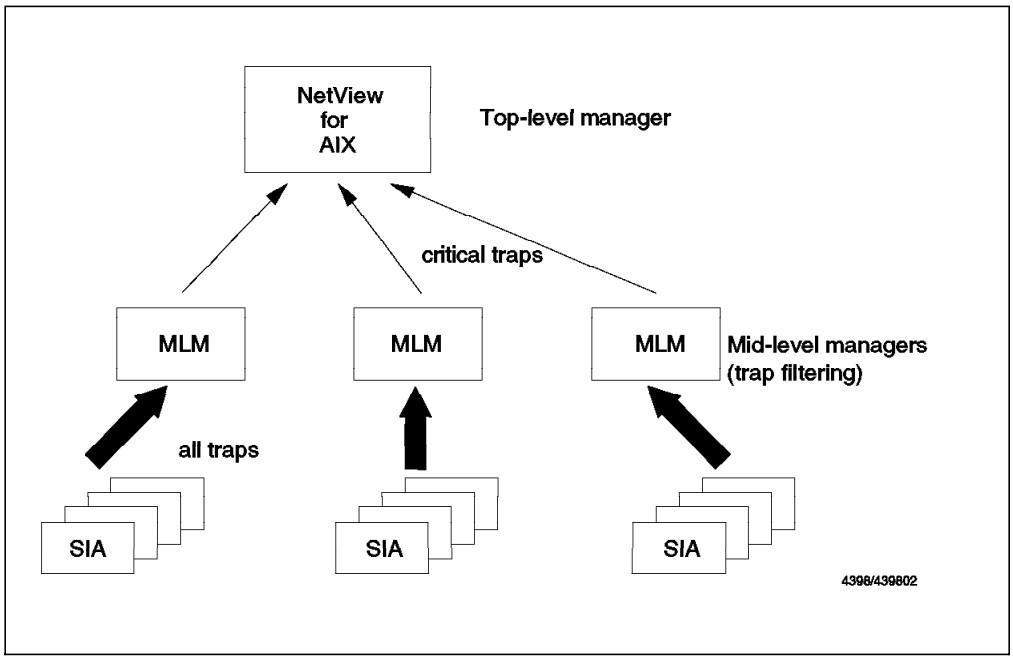


Figure 23. Mid-Level Manager Acting As Sender and Receiver of Traps. Although the lowest layer of agents are shown as SIAs, they can in fact be any kind of SNMP agent that generates traps.

2.3.1 How Systems Monitor for AIX Uses MIB Tables

One of the aspects of Systems Monitor for AIX that can be confusing is that SNMP is not only used for receiving information from it, but also for *configuring* it; that is, whenever we discuss a configuration table in Systems Monitor for AIX (for example, the Alias Table, Command Table) we are really talking about a MIB definition.

What does this mean in practice? Figure 24 shows an example of part of the Alias Table, when viewed via the NetView for AIX MIB Browser.

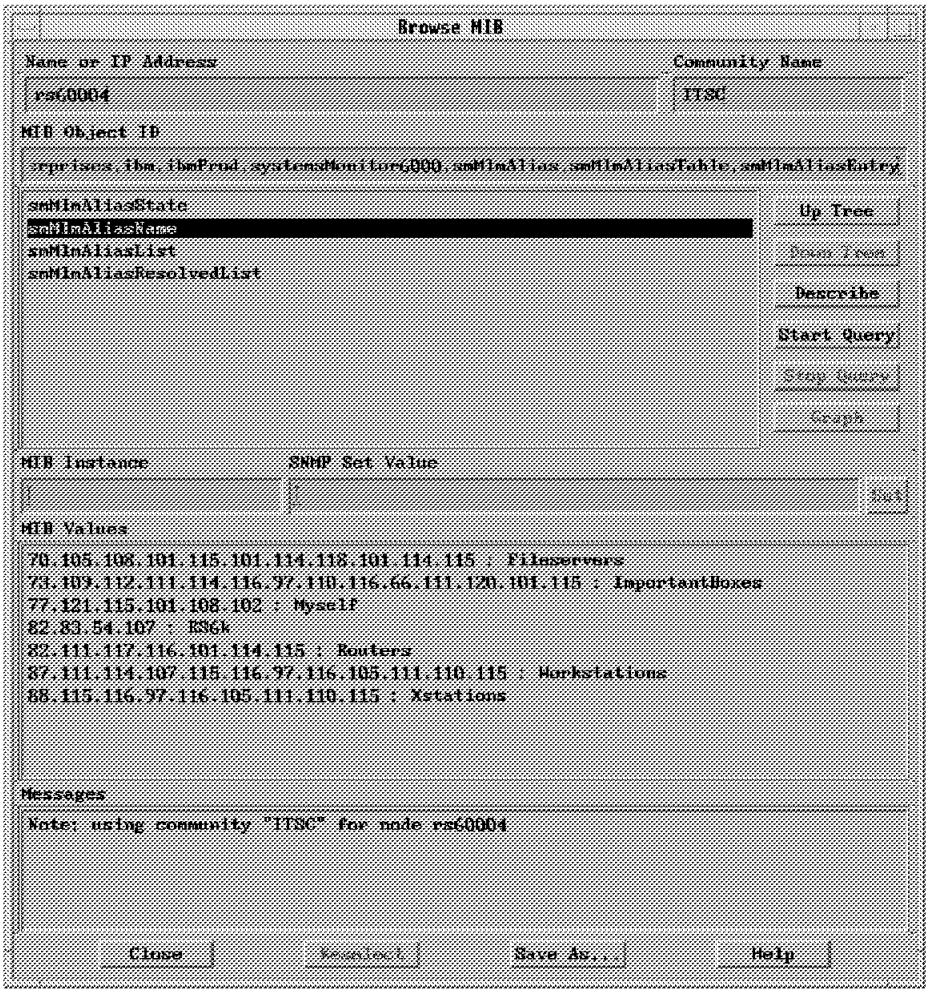


Figure 24. MIB Browser Query of MLM Alias Table

It can be seen that when using the MIB browser to query the variable smMlmAliasName, a list of the entries in the Alias Table is displayed. Note that this is Mid-Level Manager configuration information, but because it is maintained as a MIB table, we are able to retrieve it using SNMP GET requests. Notice also the rather complex instance IDs (the strings of decimal numbers). We will explain these in detail later (2.3.2, “Understanding Systems Monitor MIB Instances” on page 33).

The same list of table entries is displayed when using the Systems Monitor for AIX end-user interface as shown in Figure 25 on page 32. What this means is that when we use the end-user interface we are actually generating a series of SNMP GET requests to learn the Systems Monitor for AIX agent configuration.

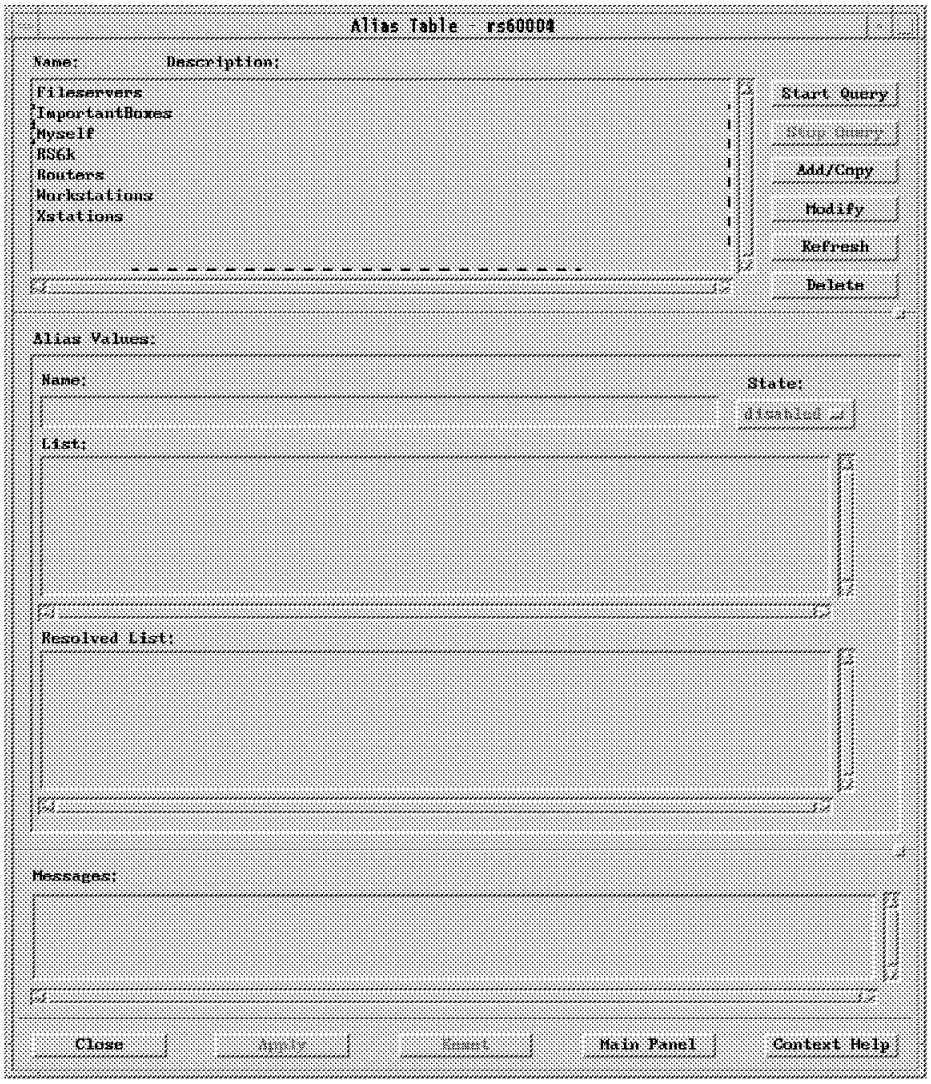


Figure 25. Alias Table Seen from the Systems Monitor for AIX End-User Interface

The MIB instances we are displaying in this way are all part of Systems Monitor MIB extensions. Figure 26 on page 33 shows where the alias entry tables fit in the structure of the MIB tree.

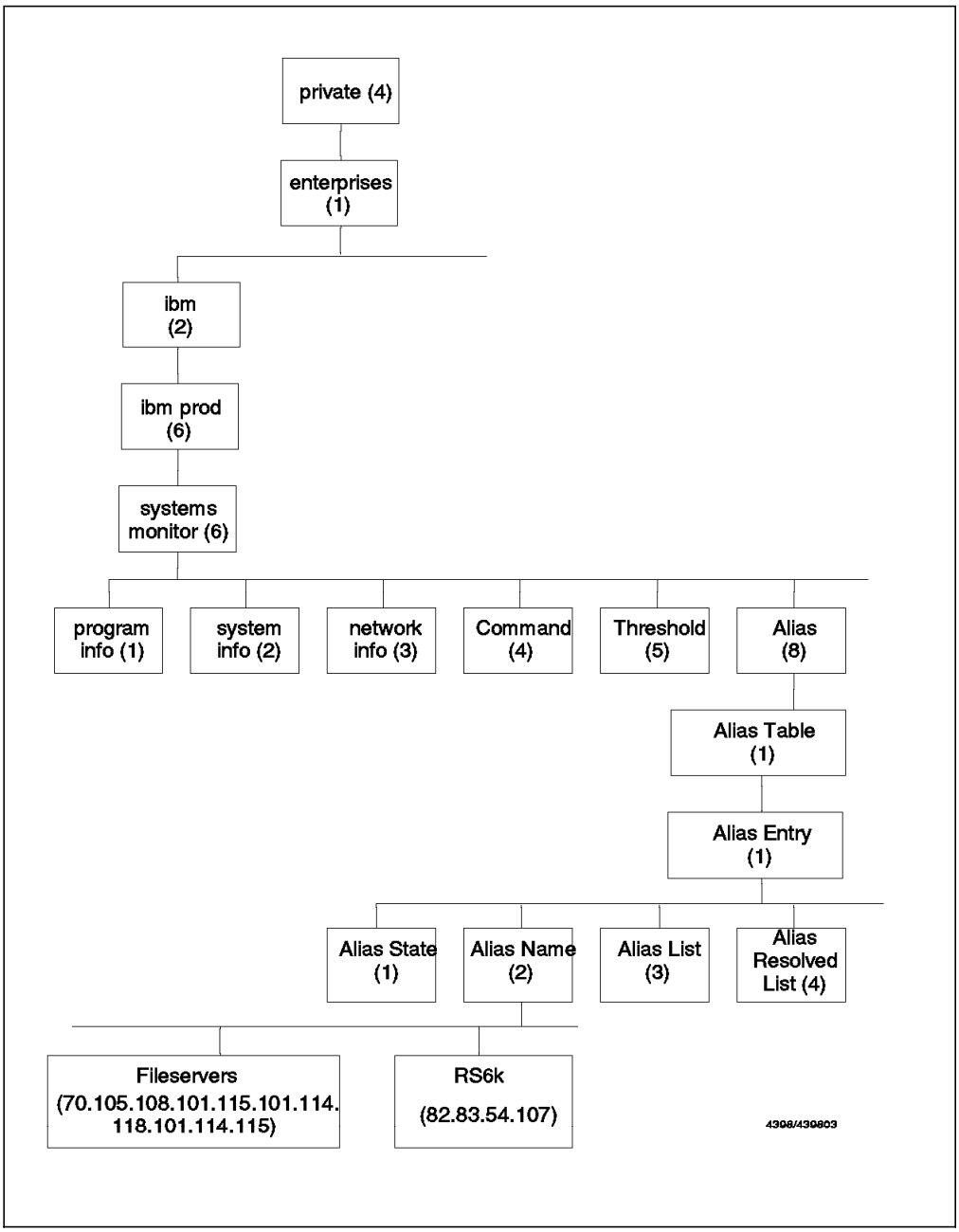


Figure 26. Configuration Tables Are Really MIB Segments

2.3.2 Understanding Systems Monitor MIB Instances

The Systems Monitor MIB is comprised mainly of tabular information. So in the case above, the Alias Table consists of objects defining the name of the alias, the list of nodes it resolves to, the state of the table entry, etc. Just as in any MIB implementation, the individual rows of the table are identified by a unique instance ID (see 1.1.4, “MIB Instances” on page 7). Unlike most SNMP implementations, however, the instances used by Systems Monitor are not single digits.

We have already noticed that the instance is displayed in the MIB Browser as a string of decimal numbers. This is in fact the *label* of the table row, rendered in ASCII. This convention is used throughout the Systems Monitor tables. For

example, in the Alias Table the instance is the name of the alias. From Figure 24 on page 31 it can be seen that for the entry Fileservers, the MIB instance is 70.105.108.101.115.101.114.118.101.114.115.

So, when we select **Fileservers** from the alias table shown in Figure 25 on page 32, and select the **Start Query** button, we are really performing an SNMP GET request for all the Alias Table objects, with the above instance ID.

But what about when we want to *update* the Systems Monitor Configuration Tables? As you would expect, this needs the use of SNMP SET, and whenever we are configuring a remote Systems Monitor for AIX agent using the configuration interface, we are in fact generating a series of SET requests.

2.4 Common Pitfalls When Installing and Configuring Systems Monitor

This section includes a discussion of some of the common areas where problems can occur, and how these problems can be avoided.

2.4.1 Cannot Update Configuration Tables

As we just discussed, whenever you attempt to update one of the Systems Monitor for AIX Configuration Tables, by selecting to **Modify** the entry, and then **Apply**, an SNMP SET command is being performed. The SNMP manager performing the SET command will look at its SNMP configuration file to see if it contains a community name for the target node. If it finds an entry, then it will put the community name in the SET command; otherwise, it will use the default community name (usually public).

When the target node (the agent) receives the SNMP SET request, it will check its own SNMP configuration file. The agent will only accept the SET request if there is an entry for the community name in the request, this entry matches the manager that sent the request, and the entry in the file specifies read/write capabilities.

If any one of these requirements is not met, the agent will return a message similar to `Received NoSuchName error (agent may have this variable in readOnly mode)`. It will also send an Authentication Failure trap (generic trap ID 4) to the manager(s) listed as trap destinations.

The series of checks that the SNMP agent will go through, when it receives an SNMP set, can be seen in Figure 27 on page 35.

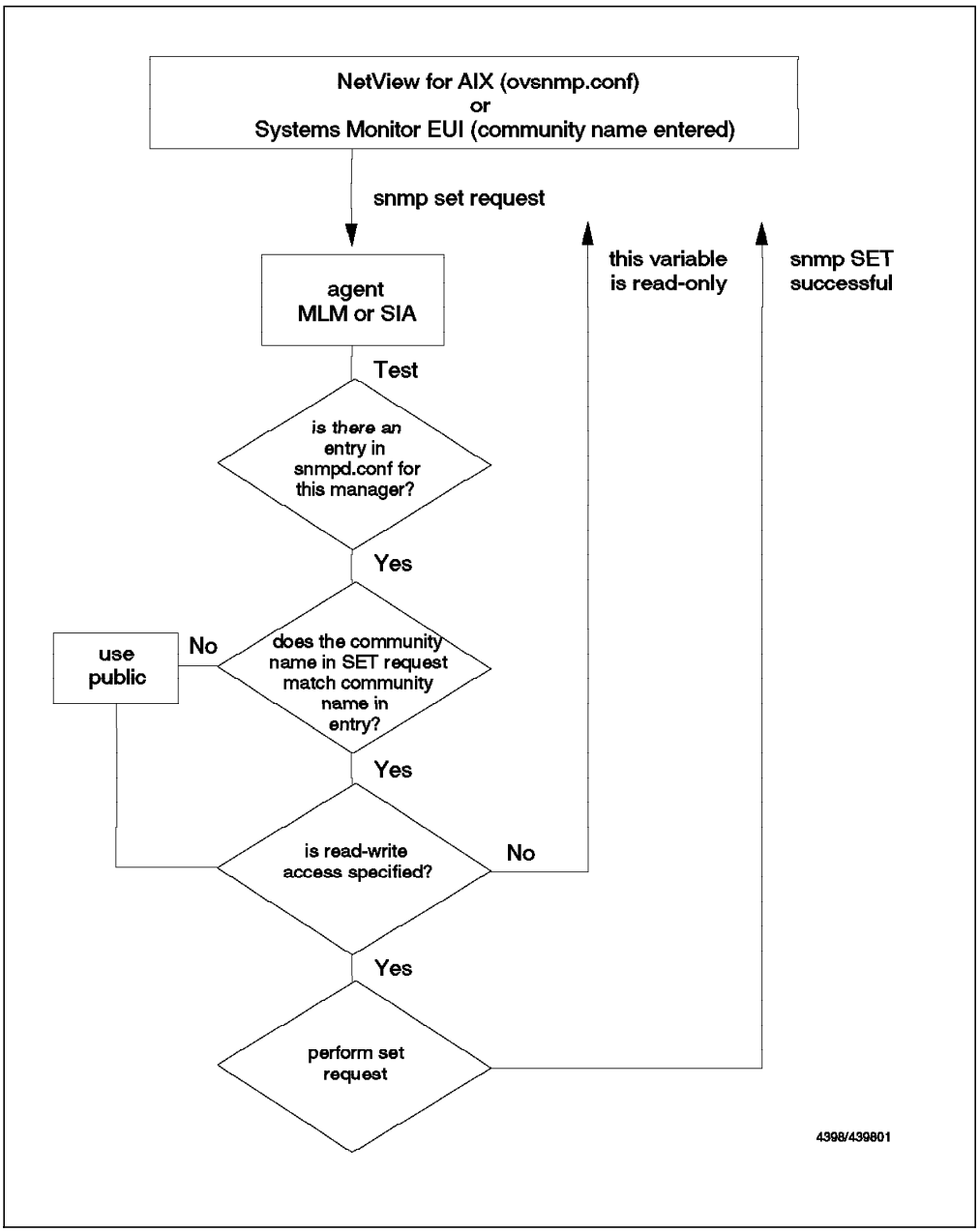


Figure 27. SNMP Community Name Verification Process

2.4.2 Unable to Display Tables

If the Systems Monitor for AIX end-user interface is started for a node that is either running Mid-Level Manager or the Systems Information Agent software, and no tables are displayed when the interface starts, this indicates that the SNMP configuration files are incorrectly configured, as described in the example below.

In our network, we have the Mid-Level Manager software installed on rs60004, and the Systems Information Agent installed on rs60002 and rs60004. The end-user interface is installed on rs60004 and rs60002, which also has NetView for AIX installed. We want to configure the Systems Information Agent agent running on rs60002 from the end-user interface installed on rs60004. In order to

do this, rs60004 must be able to perform SNMP SET and SNMP GET commands on rs60002.

The snmpd.conf file for rs60002 is shown below. This indicates that the manager with an IP address of 9.24.104.28, that is rs60002 itself, is allowed read/write access to the agent's MIB using the community name of ITSC. All other managers are permitted read-only access using a community name of public.

```
logging         file=/usr/tmp/snmpd.log         enabled
logging         size=10000                      level=3
community   public
community ITSC    9.24.104.28   255.255.255.255 readWrite # rs60002
community ITSC  127.0.0.1   255.255.255.255  readWrite
view            1.17.2          system enterprises view
# Avoid collisions with the machine list
# trap  ITSC    127.0.0.1       1.3.1   fe      # loopback
trap    ITSC    rs60002         1.2.2   fe
#snmpd          maxpacket=1024 querytimeout=120 smuxtimeout=60
smux    1.3.6.1.4.1.2.3.1.2.1.2         gated_password  # gated
smux    1.3.6.1.4.1.4.3.1               # unixd
smux    1.3.6.1.4.1.2.6.12            sm6000  # AIX Systems Monitor/6000: sysmond
smux      1.3.6.1.4.1.2.6.12            sm6000  # AIX Systems Monitor/6000: sysmond
smux     1.3.6.1.4.1.2.6.4.1           nv6000  # NetView for AIX:trapgend
smux      1.3.6.1.4.1.2.6.12            sm6000  # Systems Monitor for AIX: sysinfod
snmpd      maxpacket=16000 smuxtimeout=60       # Systems Monitor for AIX
```

Figure 28. Sample SNMP Agent Configuration File

If the end-user interface is started on rs60004 using a community name of ITSC, and the Refresh option is selected to query the agent (as shown in Figure 29 on page 37)

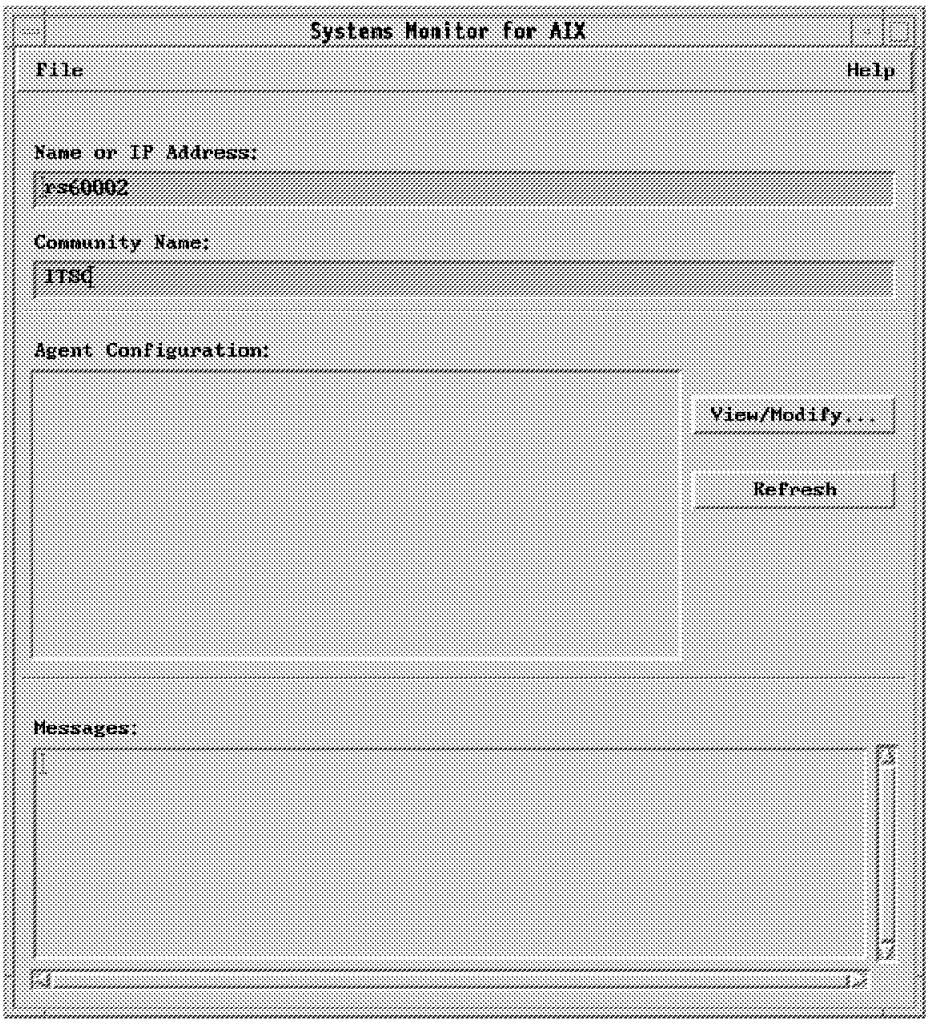


Figure 29. Result of Mid-Level Manager Using an Invalid Community Name

The manager rs60004 is trying to use a community name that has not been authorized for use with that agent, and the agent does not respond, with the result that no table information appears in the window. If the request is repeated using a community name of public, which is valid, then the agent will respond, as shown in Figure 30 on page 38. This can be achieved by replacing ITSC with public and then selecting **Refresh**, to re-query the agent.

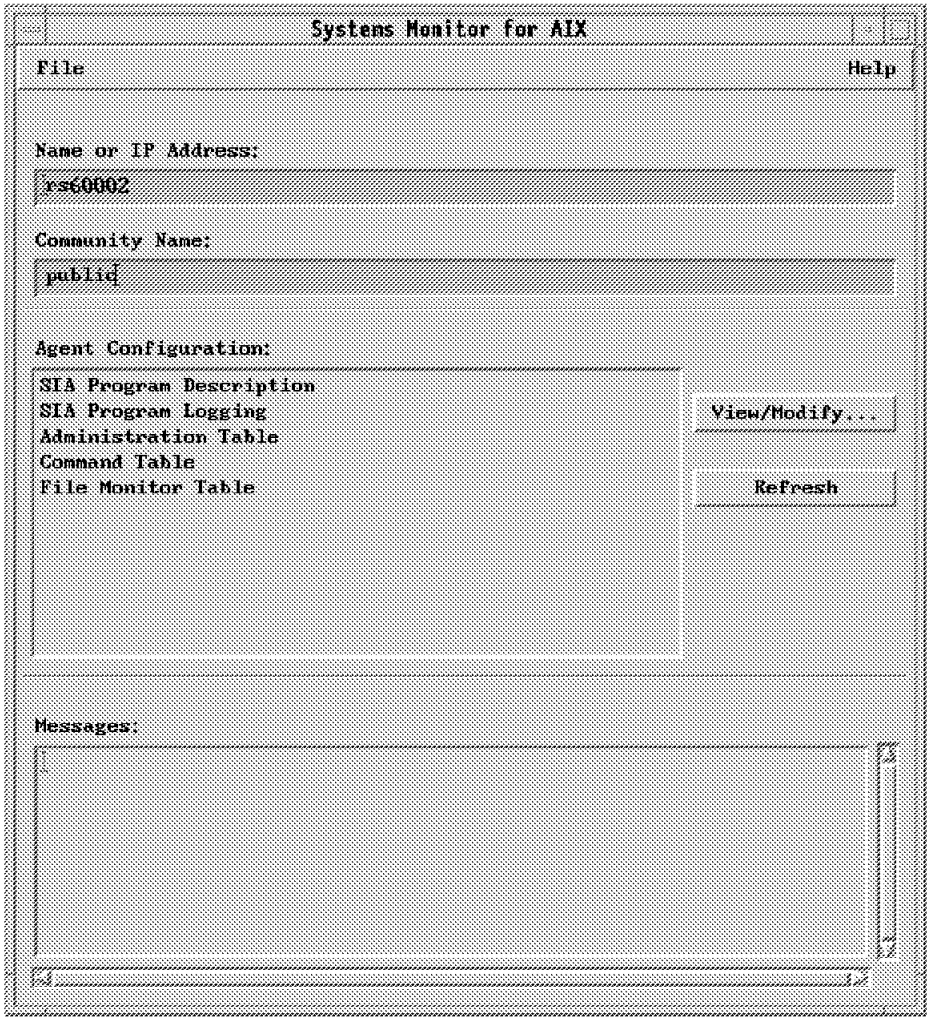


Figure 30. Connecting to Mid-Level Manager Using a Valid Community Name

However, in order to configure the agent, and add or modify entries in the agents table, rs60004 must be able to SET the agent variables. This is not possible with a community name of public, which only permits read-only access; Figure 31 on page 39 shows the results of trying to modify, or in this case, enable, an entry in the Command Table on rs60002, using a community name of public.

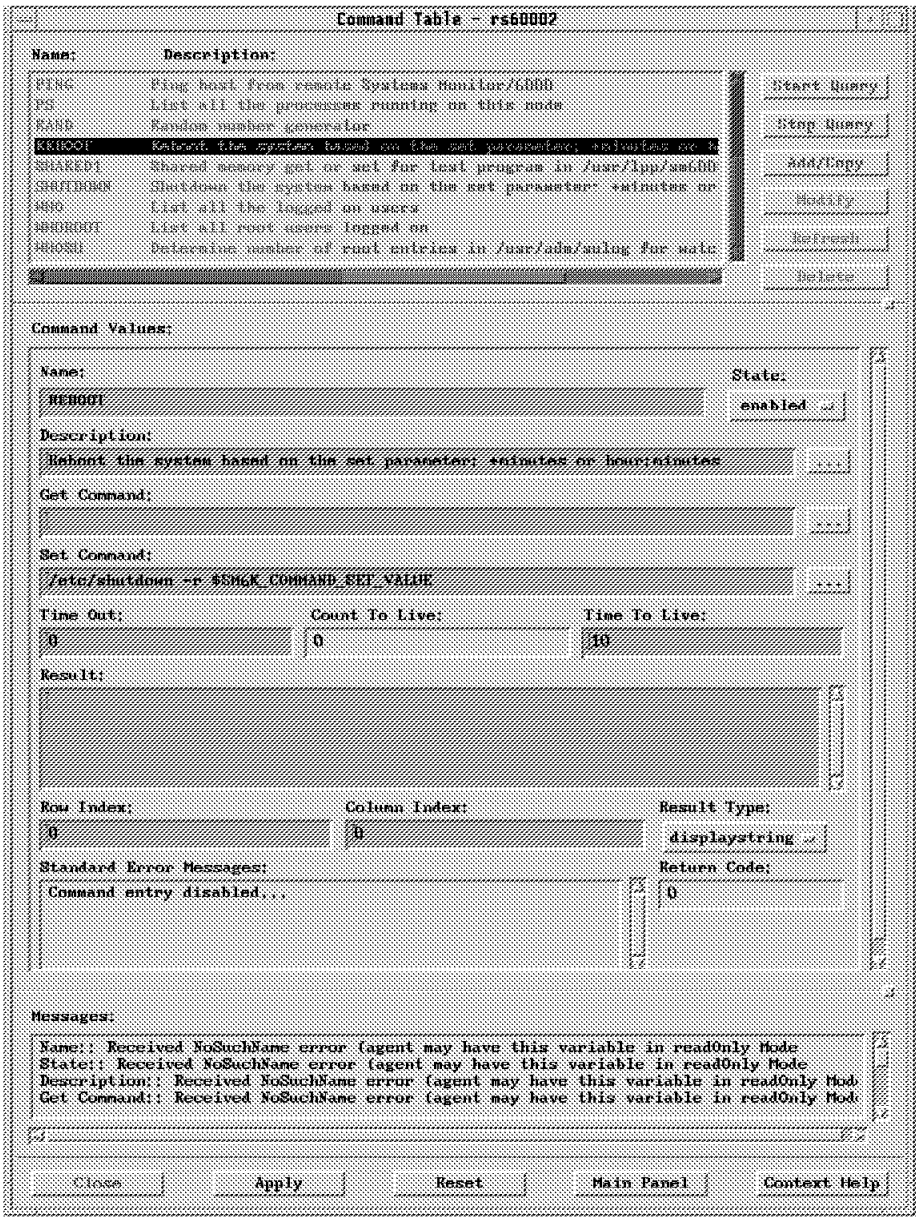


Figure 31. Trying to Change a Variable Using a Read-Only Community Name

The variable is not modified (that is, the entry in the table remains disabled) and a series of messages appear at the bottom of the table, indicating that the variable is in read-only mode.

The agent can be reconfigured to allow the manager read/write access to the MIB, by adding an additional line to snmpd.conf on rs60002, to allow rs60004, whose IP address is 9.24.104.27, read/write access using a community name of ITSC.

```
logging         file=/usr/tmp/snmpd.log         enabled
logging         size=10000                      level=3
community  public
community ITSC    9.24.104.28   255.255.255.255 readWrite # rs60002
community ITSC    9.24.104.27   255.255.255.255 readWrite # rs60002
community ITSC  127.0.0.1   255.255.255.255  readWrite
view            1.17.2          system enterprises view
# Avoid collisions with the machine list
# trap  ITSC    127.0.0.1       1.3.1   fe      # loopback
trap    ITSC    rs60002         1.2.2   fe
#snmpd          maxpacket=1024 querytimeout=120 smuxtimeout=60
smux    1.3.6.1.4.1.2.3.1.2.1.2         gated_password  # gated
smux    1.3.6.1.4.1.4.3.1               # unixd
smux    1.3.6.1.4.1.2.6.12            sm6000  # AIX Systems Monitor/6000: sysmond
smux      1.3.6.1.4.1.2.6.12            sm6000  # AIX Systems Monitor/6000: sysmond
smux      1.3.6.1.4.1.2.6.4.1           nv6000  # NetView for AIX:trapgend
smux      1.3.6.1.4.1.2.6.12            sm6000  # Systems Monitor for AIX: sysinfod
snmpd      maxpacket=16000 smuxtimeout=60       # Systems Monitor for AIX
```

Figure 32. SNMP Agent Configuration File Permitting Read/Write Access

Don′t Forget

You must refresh the snmpd daemon on rs60002 after making any changes to the snmpd.conf file, or these changes will not be implemented.

Figure 33 on page 41 shows that using the community name of ITSC will result in the agent responding to the SNMP GET command, and the MIB tables will be displayed.

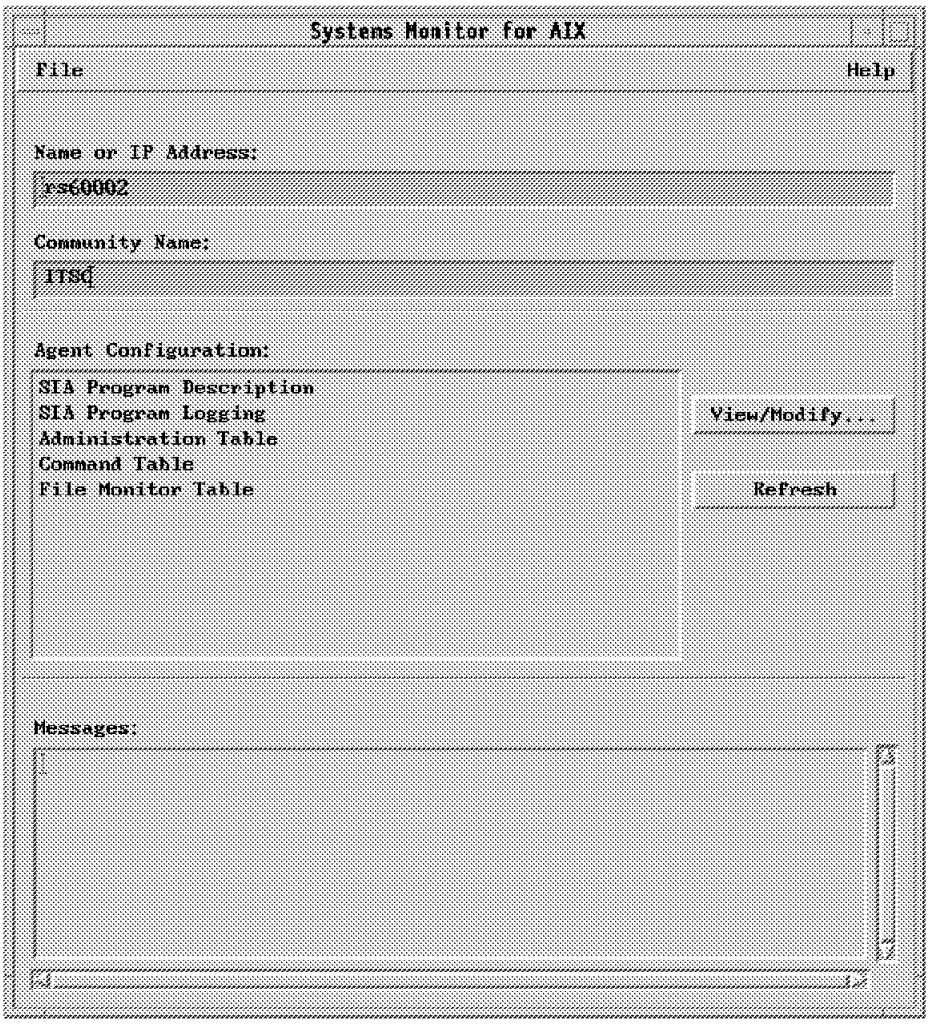


Figure 33. Querying MIB Variables Using a Valid Community Name

Now if you try to enable the entry in the Command Table, it will work, and you will receive a series of Set Successful messages.

2.4.3 The Resume File

The resume file is one of the enhancements in Systems Monitor for AIX Version 2. It is a kind of journal system, which keeps track of any configuration changes you make and re-creates them each time the Systems Monitor for AIX agent is restarted. This means that you do not have to remember to save your changes each time you make a configuration update.

The following are some potential pitfalls that need to be understood regarding use of the resume file:

Always save configurations.

The resume file is designed to be a safety net, so that if the system or the end-user interface fails while you are in the middle of configuring one of the subagents, the configuration details can be recovered, since the SETs are saved in the resume file. The resume file should *not* be viewed as an alternative to saving configuration files. It will keep writing into itself until a certain point, N entries since the dump

of the current configuration, and then dump its current configuration, and clean out redundant entries.

This is important to remember if you are likely to be doing many configuration changes on a regular basis.

The resume file is always used by default.

Don't get caught out by the resume file when you *do* save configurations. When the subagent is started with no flags it uses the resume file by default and will not look at the configuration file at all, not even if it can be found in the default directory /var/adm/smv2/mlm/config.

So, you may find that table entries you expected to find are not there, when you start up the end-user interface. This can be resolved, by starting the subagent with the following flags:

- -i, to prevent use of the resume file
- -c directory, where directory is the name of the directory containing the configuration file or files

2.4.4 Conflicting Use of Trap Ports

The Mid-Level Manager, System Level Manager and NetView for AIX listen for traps on ports 162/TCP and 162/UDP. If you have both of them running on one system and if the midmand subprocess opens the ports before NetView for AIX, then the NetView trap receiver, trapd, will be unable to open the port, and will fail. Several of the other NetView for AIX daemons which are dependent on trapd, will also fail, leading to the error messages shown in Figure 34 on page 43.

```
object manager name: trapd
behavior:            OVs_WELL_BEHAVED
state:               FAILED
PID:                 10557
last message:        A trapd is already running.
exit status:         -

object manager name: ovtopmd
behavior:            OVs_WELL_BEHAVED
state:               UNSTARTABLE
PID:                 (never run)
last message:
exit status:         -

object manager name: snmpCollect
behavior:            OVs_WELL_BEHAVED
state:               UNSTARTABLE
PID:                 (never run)
last message:
exit status:         -

object manager name: ovactiond
behavior:            OVs_WELL_BEHAVED
state:               UNSTARTABLE
PID:                 (never run)
last message:
exit status:         -

object manager name: netmon
behavior:            OVs_WELL_BEHAVED
state:               UNSTARTABLE
PID:                 (never run)
last message:
exit status:         -
```

Figure 34. Failing NetView for AIX daemons

NetView for AIX Version 4 now provides a facility to enable you to modify its trap reception port, as discussed in 1.3, “New Functions in NetView for AIX Version 4 Related to AIX Systems Monitor/6000” on page 16. You may wonder why you would want to place both the Mid-Level Manager and NetView for AIX on the same machine. One possible reason is to make use of the threshold checking capabilities of the MLM, which are superior to those offered by NetView for AIX alone.

2.4.5 Traps Do not Arrive

You may set up the MLM to generate threshold traps, or filter traps from other nodes, but then find that nothing arrives at NetView for AIX. This is probably due to the fact that there are no entries in the Trap Destination Table. Remember that the Mid-Level Manager does not use the trap destinations defined in /etc/snmpd.conf, but has its own Trap Destination Table (a MIB table, like all Systems Monitor tables).

The following are several different types of traps that the Mid-Level Manager must deal with:

- Traps from File Monitor Table, indicating file modifications and status changes
- Threshold traps from the Threshold Table
- Interface status traps from the Mid-Level Manager polling function
- Traps from any other SNMP agent

The Mid-Level Manager is then responsible for deciding what to do with all the traps that it has received. Filters can be added to specify whether traps should be blocked, throttled or sent to NetView for AIX managers, but the Mid-Level Manager also needs to decide where to send the traps. It is essential not to forget the Trap Destination Table. The process that the Mid-Level Manager goes through when deciding whether to send a trap, and where to send it, is as follows:

1. Checks the Filter Table to see if the trap matches a filter.
   - If it does, and the filter is set to blocktraps, no trap is sent.
   - If it does, and the filter is set to sendtraps, and there *is* an entry in the Trap Destination field for that filter, then the trap is sent to the node specified in that entry.
   - If it does, and the filter is set to sendtraps, and there is *no* entry in the Trap Destination field for that filter, then the Trap Destination Table is looked at.
   - If it does not, and the default filter action (specified in the Trap Reception Settings Table) is to send traps, rather then block traps then the Trap Destination Table is looked at.
2. Checks the Trap Destination Table.

   Traps will be sent to all nodes entered in this table, depending on the specified activation and deactivation times.

For information and examples of how to configure these tables, see 6.1, “MIB Processing Tables” on page 147.

2.4.6 Using the Restart Facility

The restart facility allows the subagent (that is, either sysinfod or midmand) to be restarted, either with the same flags, or with different flags. The following example shows how to reinitialize the Systems Information Agent subagent, so that it uses configuration files located in directory /usr/adm/sm6000/sia/config instead of configuration files located in /tmp.

Using the MIB Browser function of NetView for AIX, it is possible to query the MIB variable smSiaProgramControlCurrentFlags to determine what flags sysinfod is currently using. It can be seen from Figure 35 on page 45 that there is only one current flag, which is -c /tmp. This means that the directory containing the current configuration file is /tmp.

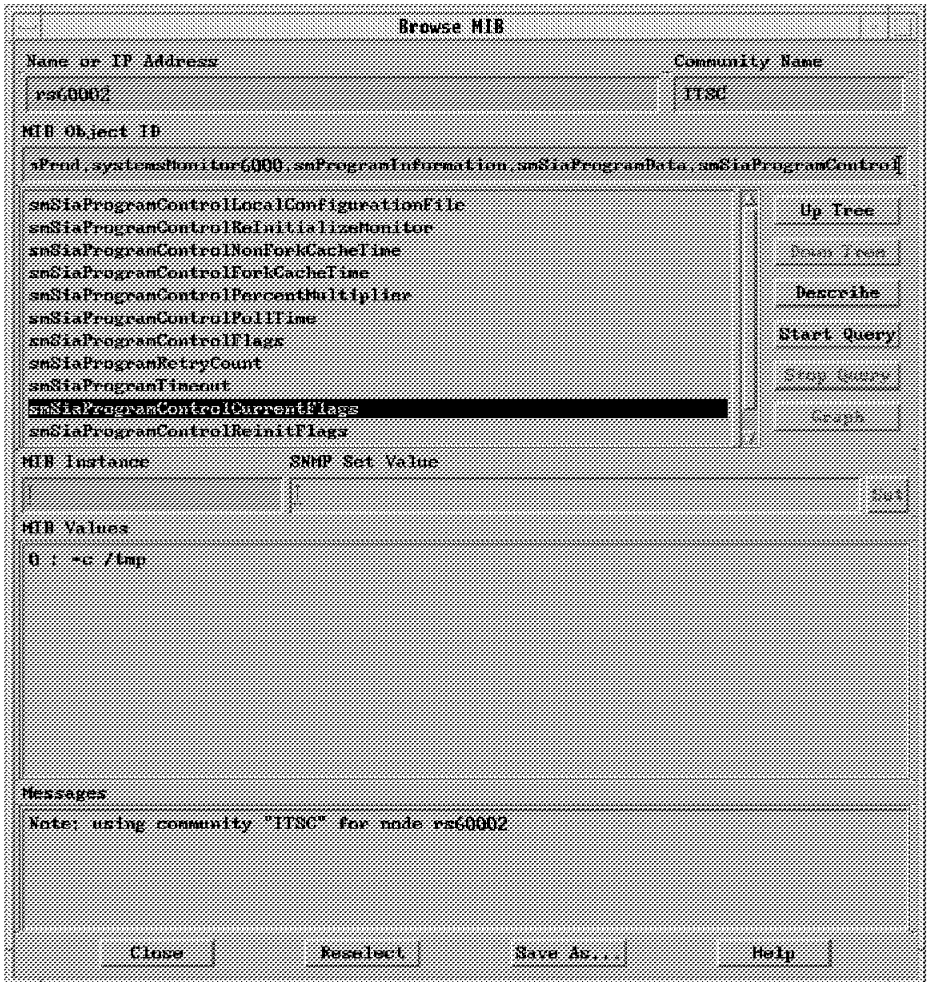


Figure 35. Querying MIB Variable smSiaProgramControlCurrentFlags

There are three options that can be set for reinitializing the subagent. These include the following:

false Reinitialization of the subagent is not permitted.

trueReinit The subagent can be reinitialized, and uses the same flags as before.

trueSaved Reinitialization of the subagent is permitted, and a different set of flags is used.

These options can be seen in Figure 36 on page 46, which is the output from performing a Describe on the MIB variable `smSiaProgramControlReinitializeMonitor`.

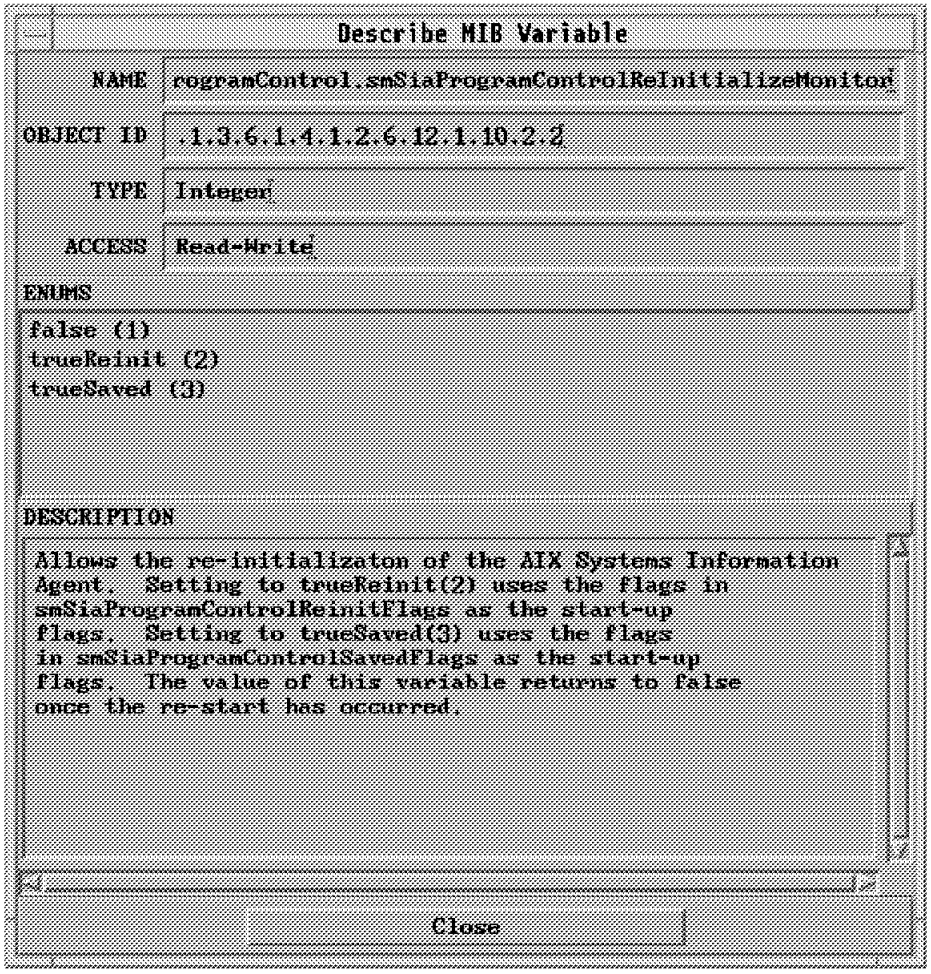


Figure 36. Descripton of the MIB Variable ReinitializeMonitor

If the option trueReinit is chosen, then the subagent will be reinitialized using the flags specified in the MIB variable `smSiaProgramControlReinitFlags`. From Figure 37 on page 47, which is the output from querying this MIB variable, it can be seen that these flags are the same as the current flags.

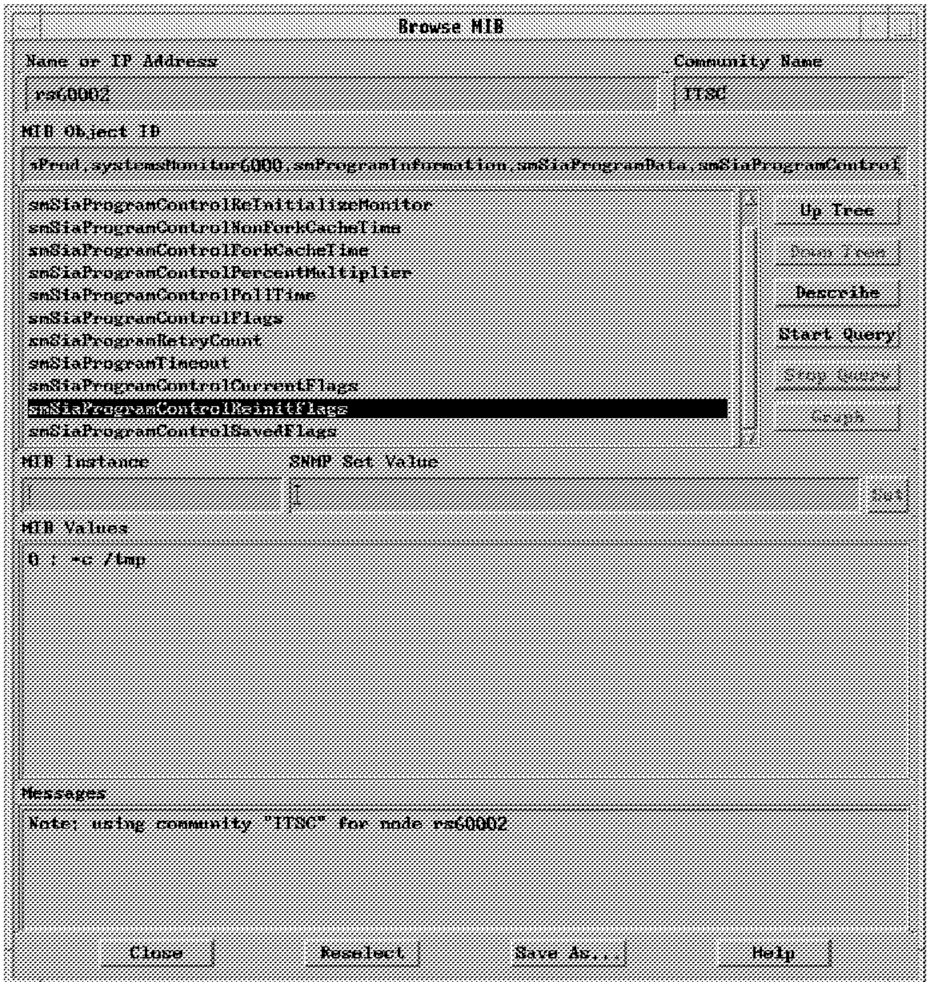


Figure 37. Output from Querying MIB Variable smSiaProgramControlReinitFlags

Therefore, if the subagent is to be reinitialized with the same flags, then `smSiaProgramControlReinitializeMonitor` should be set to trueReinit. However, in our scenario, we would like the subagent to be restarted using a different configuration file, located in directory /usr/adm/sm6000/sia/config. To achieve this, we need to configure the MIB variable `smSiaProgramControlSavedFlags`, and specify the new flags that we want to use. This is achieved by performing a SET on the above MIB variable, as shown in Figure 38 on page 48.

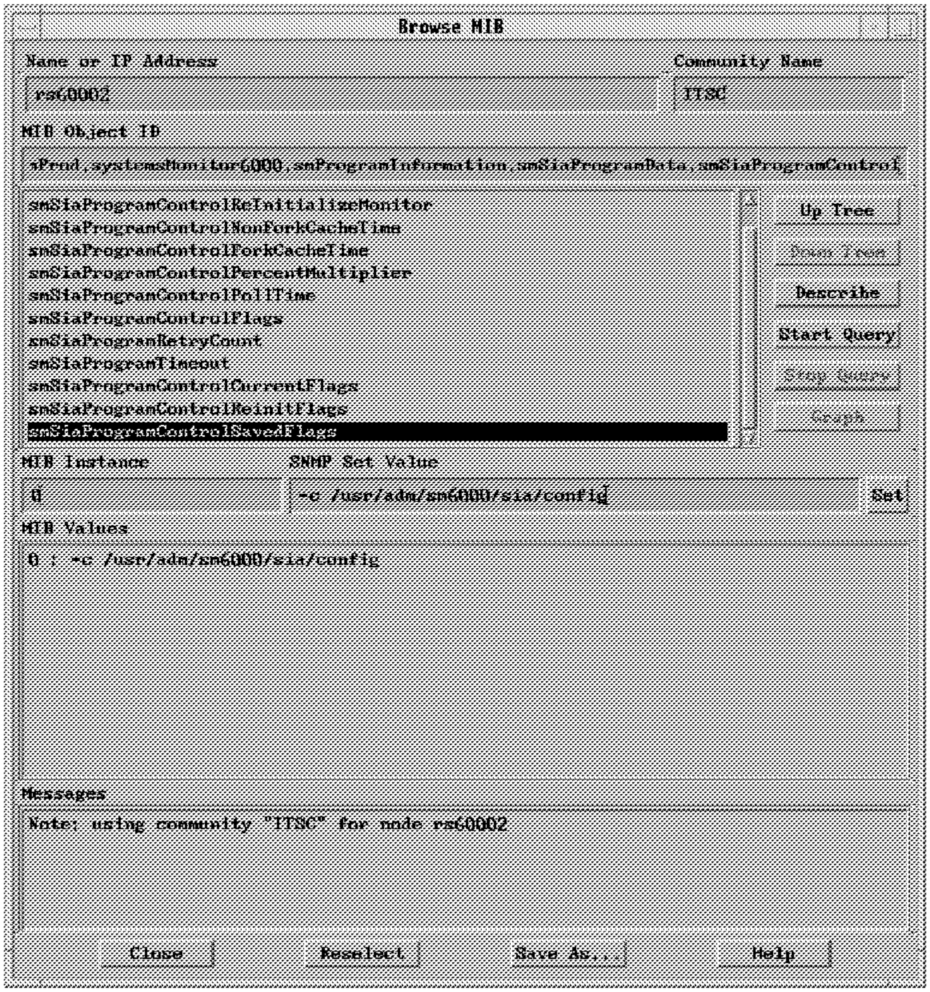


Figure 38. Setting the MIB Variable smSiaProgramControlSavedFlags

Finally, we need to set the MIB variable smSiaProgramControlReinitializeMonitor to trueSaved, so that when the subagent reinitializes, it picks up the new configuration file. This is shown in Figure 39 on page 49.

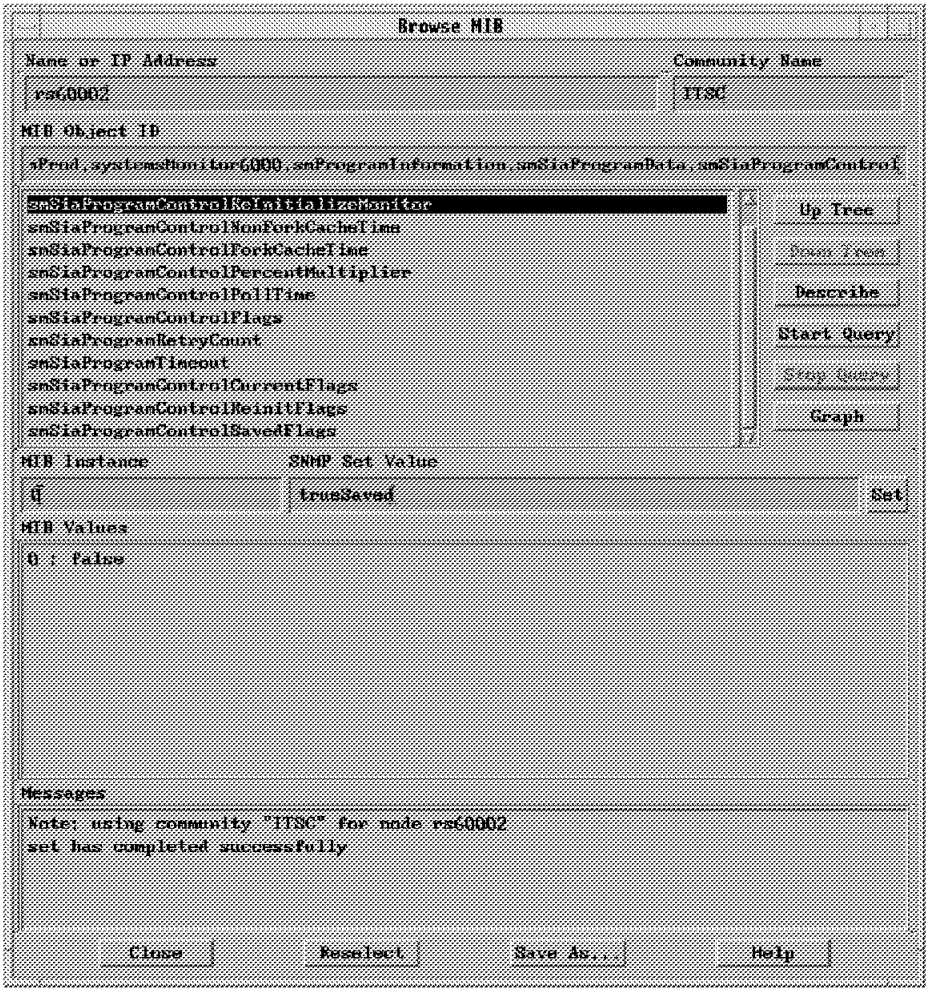


Figure 39. Setting the MIB Variable smSiaProgramControlReInitializeMonitor

In order to prove that the subagent has been reinitialized and is using a different configuration file, we can display the current flags that the subagent is using by querying MIB variable smSiaProgramControlCurrentFlags. The output seen in Figure 40 on page 50 shows that the configuration directory is indeed /usr/adm/sm6000/config.

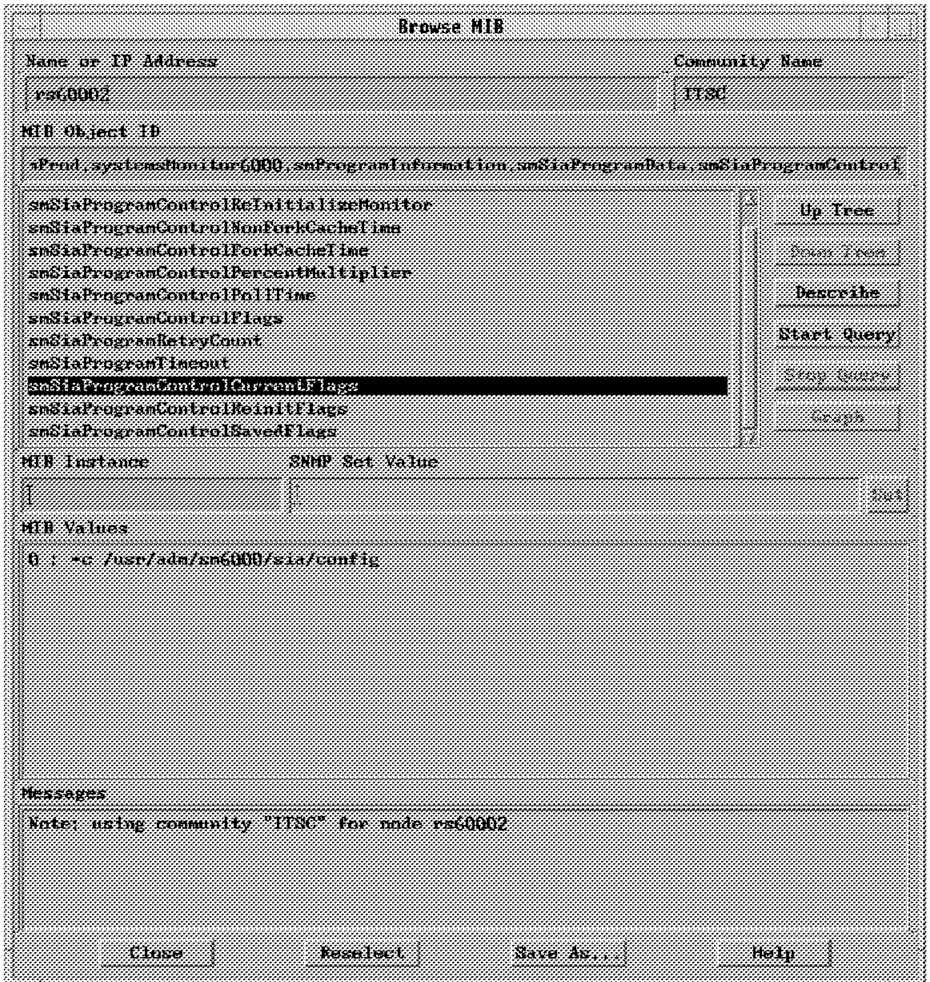


Figure 40. Querying the MIB Variable smSiaProgramControlCurrentFlags

Once the subagent has been reinitialized, it cannot be reinitialized again without the above process being repeated. This can be seen by querying the MIB variable `smSiaProgramControlReInitializeMonitor`, which as shown in Figure 41 on page 51, returns to a value of false.

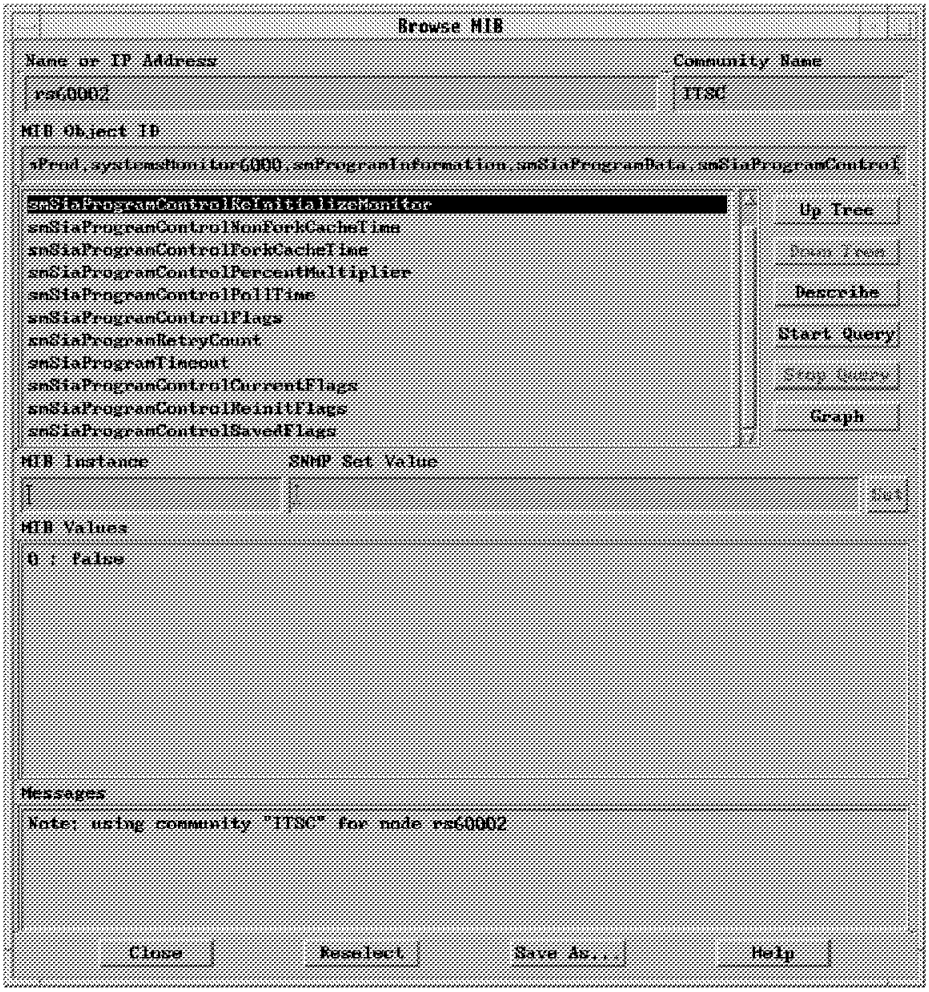


Figure 41. Querying the MIB Variable smSiaProgramControlReInitializeMonitor

Remember the Resume File

The above example assumes that the resume files are empty; otherwise, the change in configuration file will not be effective (see 2.4.3, “The Resume File” on page 41). An alternative is to add the flag -i to the list of flags that the subagent is to use.

An alternative method, is to use the Reinitialize option from the Systems Monitor for AIX end-user interface, as shown in Figure 42 on page 52. It is possible to specify the time of reinitialization, as well as the flags that are to be used, for the subagent.

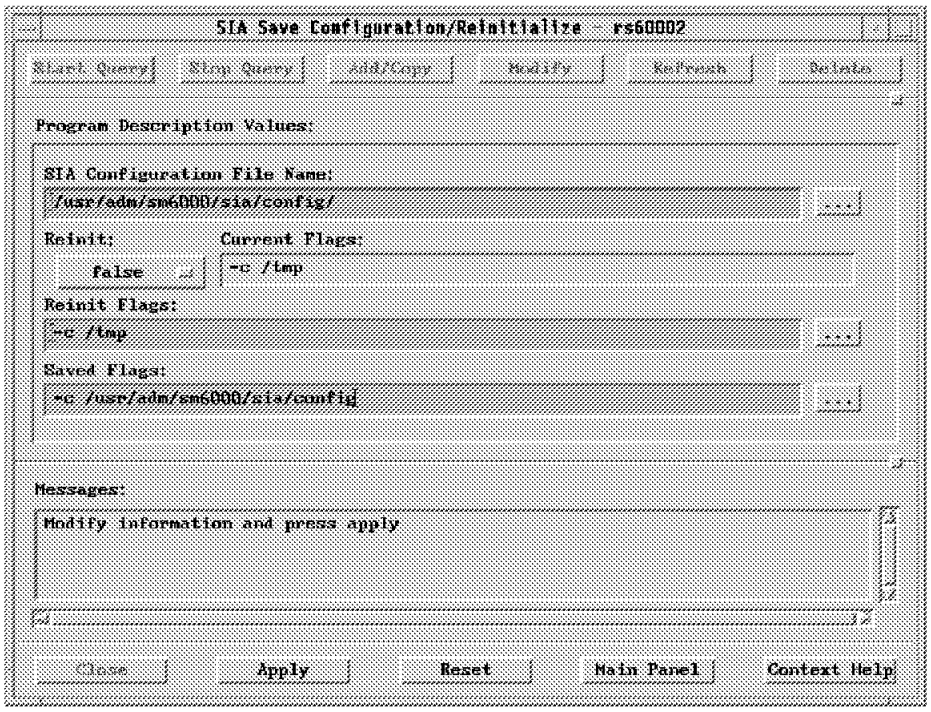


Figure 42. Reinitializing the Sysinfod Subagent from the End-User Interface

Full details of the flag options available for starting the Systems Monitor for AIX agents may be found in *Systems Monitor for AIX User's Guide,* SC31-7150.

2.5 Remotely Installing Mid-Level Manager Software

All the software (that is, the end-user interface, Mid-Level Manager and Systems Information Agent) can be installed directly from the media or remotely from another system that already has the software installed. The RISC System/6000 graphical user interface, SMIT, can be used in both types of installation.

The following example shows how to remotely install the Mid-Level Manager software. Typing `smit commo` takes you directly to the SMIT Communications Applications and Services panel, shown in Figure 43 on page 53.

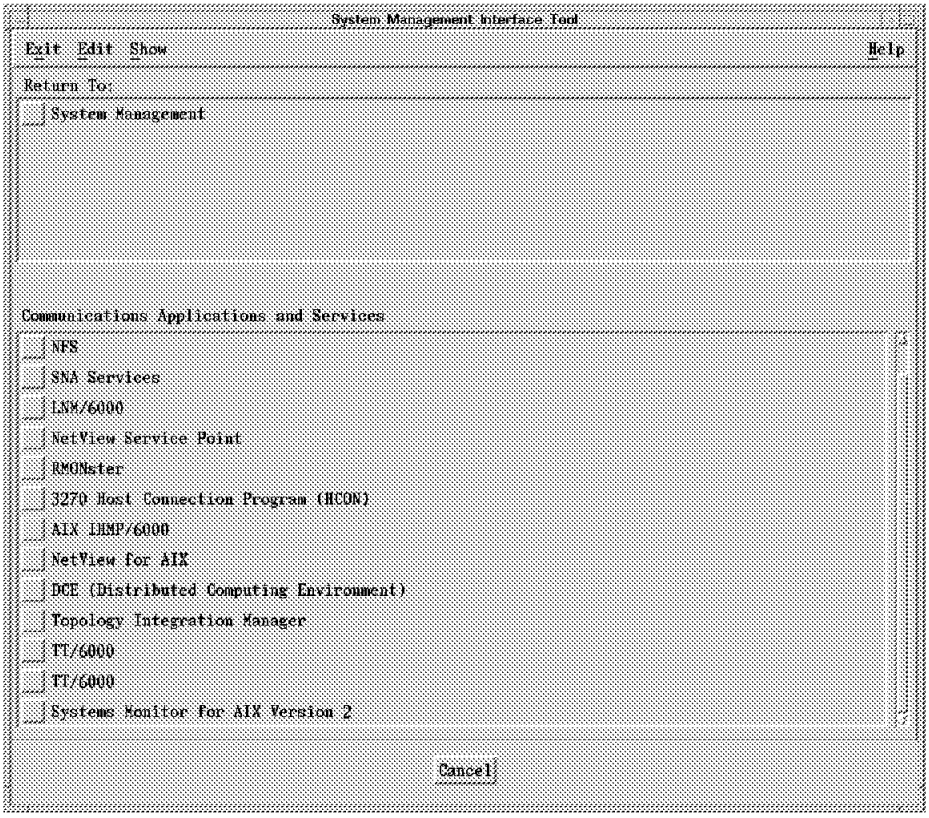


Figure 43. Communication Applications and Services SMIT Panel

Next, select **Systems Monitor For AIX Version 2** and then **Remote Operations** to reach the panel shown in Figure 44 on page 54.

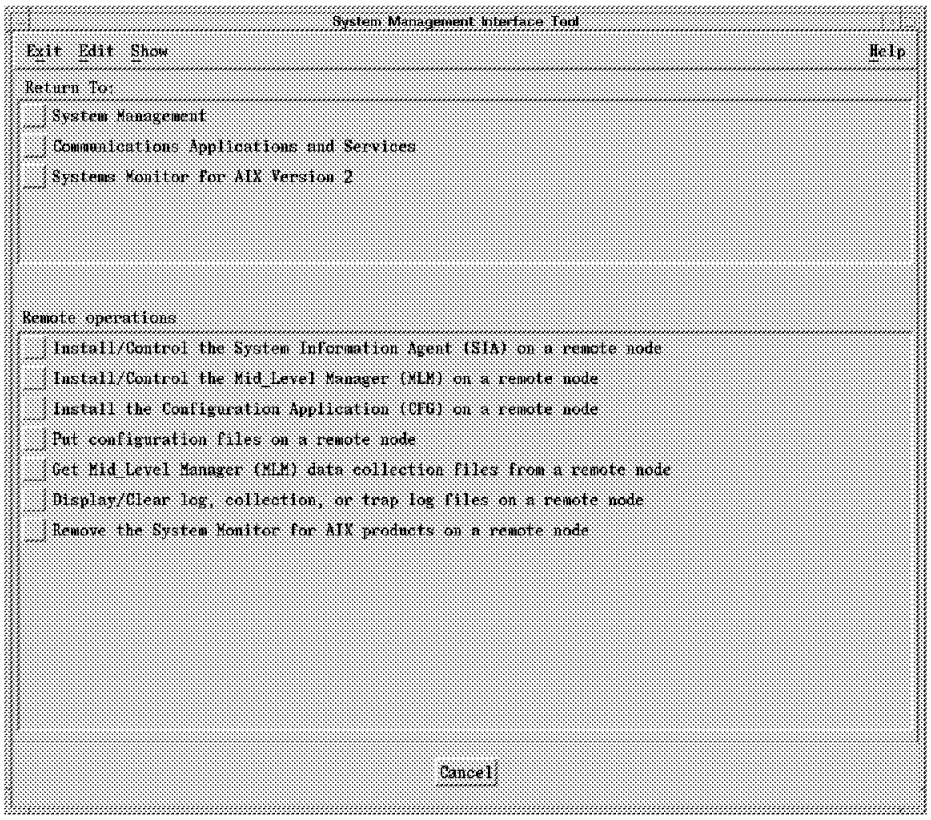


Figure 44. Remote Operations SMIT Panel

Then, select the second option from this screen (**Install/Control the Mid Level Manager (MLM) on a remote node**). The resulting screen shown in Figure 45 prompts you to enter the name or IP address of the remote node on which the Mid-Level Manager software is to be installed, as well as the root password.

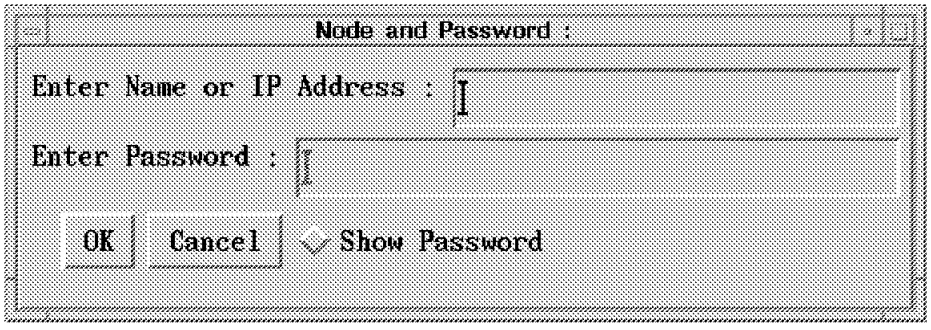


Figure 45. Node and Password Prompt Panel

Enter the host name and root password, and select **OK**. This takes you to the panel shown in Figure 46 on page 55.

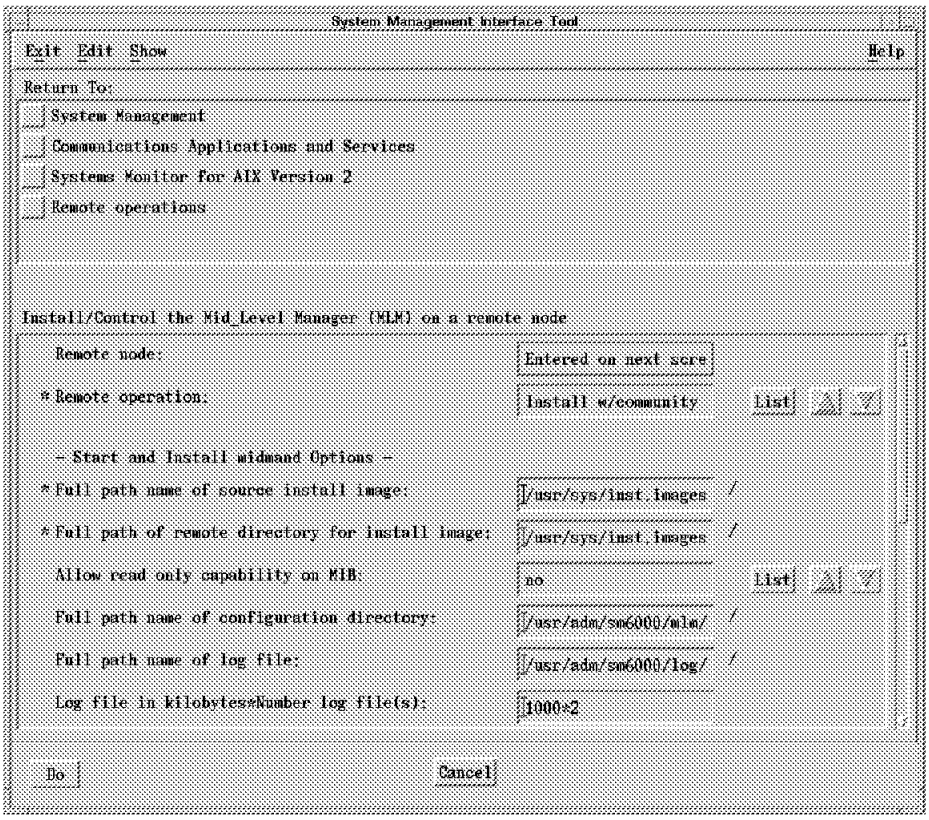


Figure 46. Install/Control Mid Level Manager on a Remote Node SMIT Screen

From here you can select the installation parameters, for example, specifying the path name of the configuration directory and the log file, as well as options to install the community file, and to stop or start various daemons. A list of the remote operations that are possible can be displayed by highlighting Remote Operation and pressing PF4.

In our scenario, we chose the option **Install w/community file,** which caused the MLM code to be transmitted (using FTP) and installp to be remotely invoked to install it. As part of the installation, the community file /usr/OV/conf/ovsnmp.conf will be copied to the remote node.

This community file is important, since it is the file that midmand will check every time it wishes to send an SNMP GET or SET request, to determine the community name that is should use for communicating with the agent. If you want the same community file on all Mid-Level Manager nodes, then choose the option to install with the community file; if not, then it will be necessary to create your own ovsnmp.conf file.

2.6 Installing the End-User Interface

When the Systems Monitor for AIX end-user interface is installed, it will load both the Systems Information Agent MIB and the Mid-Level Manager MIB, which are called ibm-sysinfod.mib and ibm-midlevelmgr.mib respectively. It will also add the Systems Monitor for AIX default traps to trapd.conf.

Therefore, although the end-user interface can be installed on any RISC System/6000 in the network, it should always be installed on the NetView for AIX node.

2.6.1 Using the Systems Monitor for AIX End-User Interface

The end-user interface can be used to configure nodes that have any of the following installed:

- Systems Monitor for AIX Systems Information Agent subagent
- Systems Monitor for AIX Mid-Level Manager subagent
- Systems Monitor/6000 v1
- Systems Monitor for HP-UX
- Systems Monitor for NCR UNIX
- Systems Monitor for SUN Solaris

Before displaying the main menu panel, the end-user interface checks to see which agent is installed on the target node, and build an appropriate list of options. Some examples follow:

- Figure 47 on page 57 is for an SIA node
- Figure 48 on page 58 is for a node with both SIA and MLM installed
- Figure 49 on page 59 is for a Systems Monitor V1 node

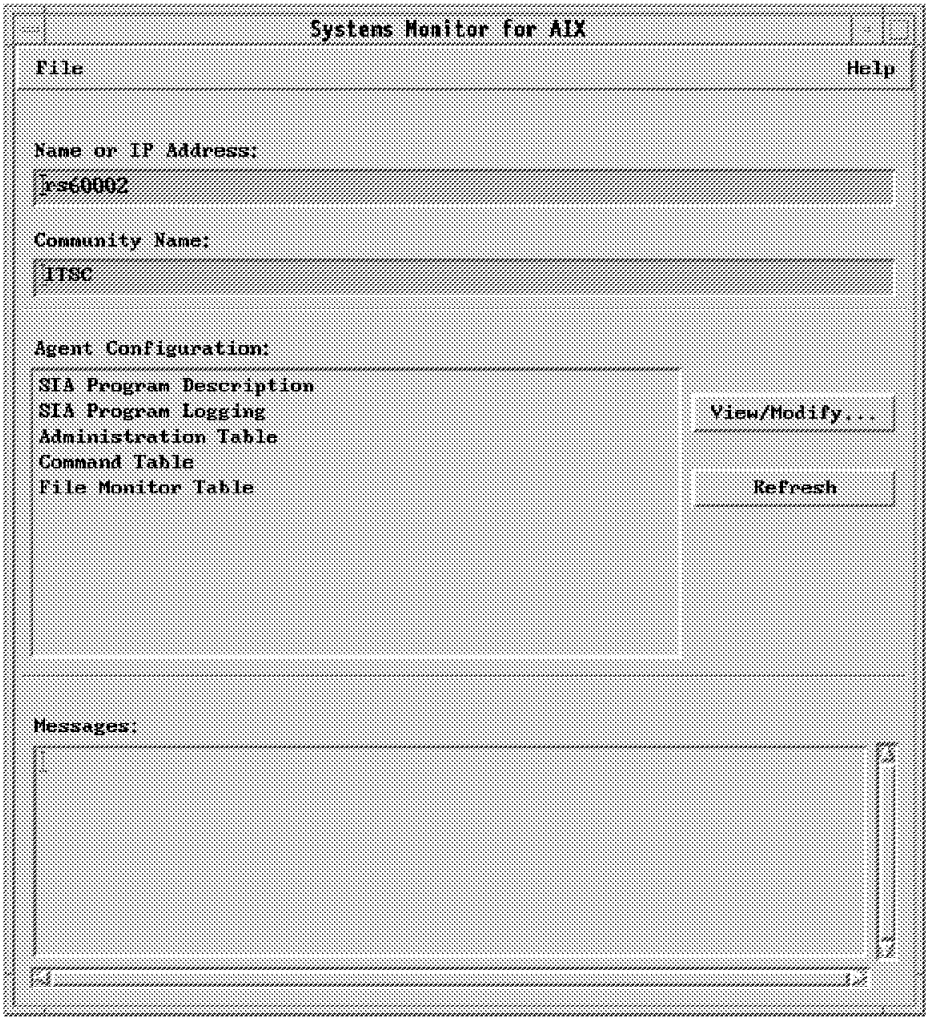
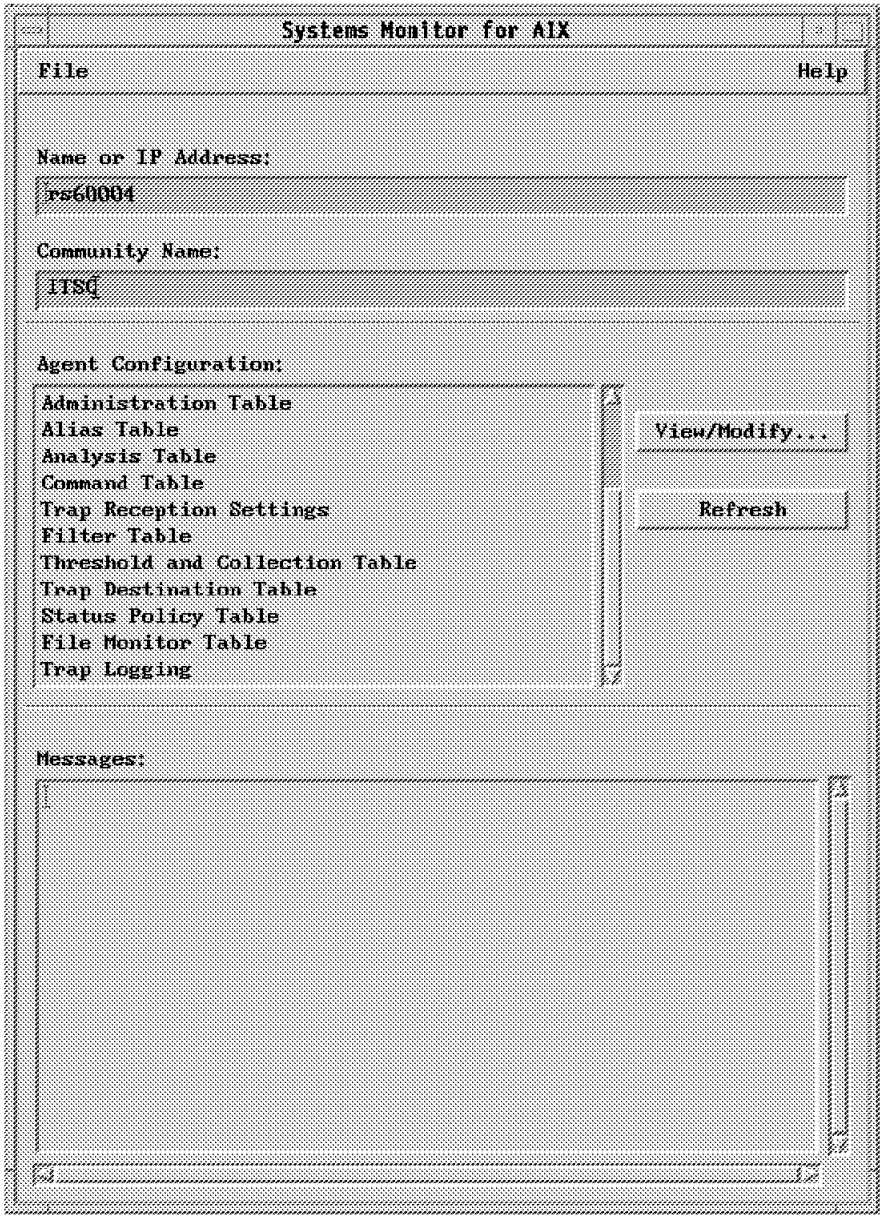


Figure 47. End-User Interface Display for Systems Information Agent Node



Installed

Figure 48. End-User Interface Display for Node with Systems Information Agent and Mid-Level Manager

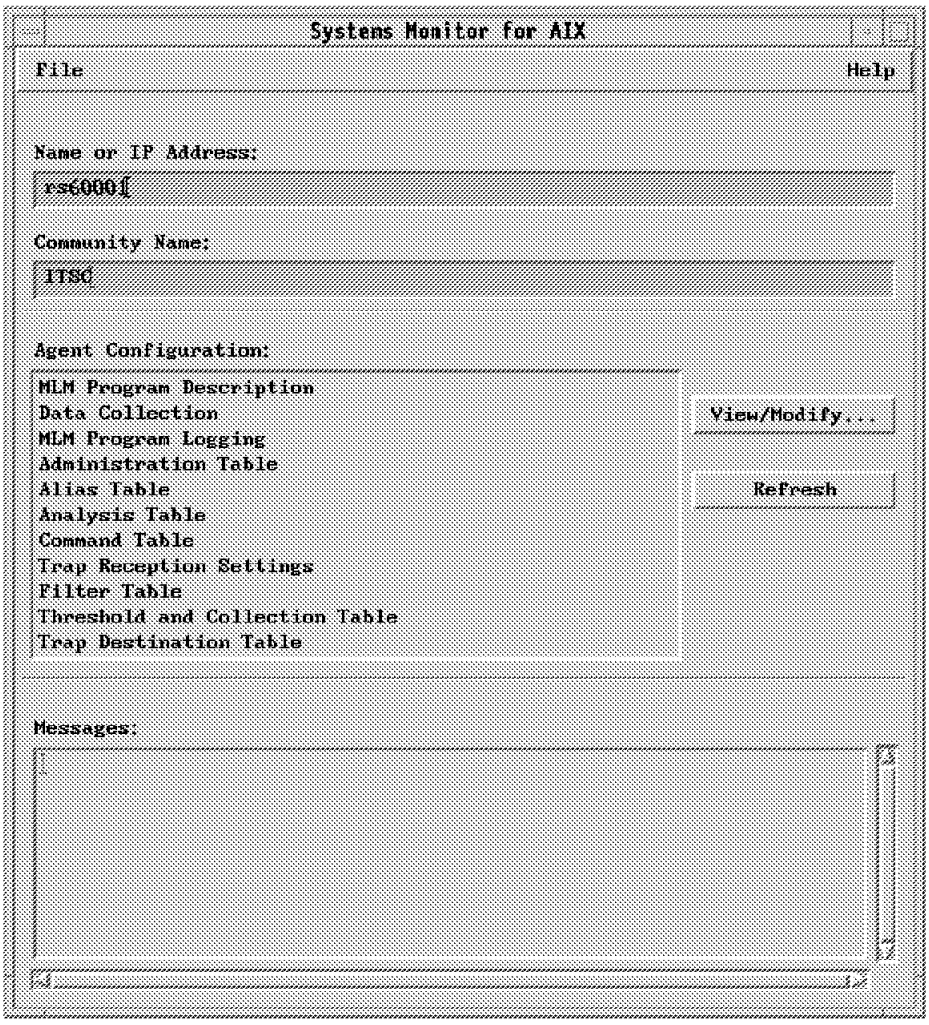


Figure 49. End-User Interface Display for Systems Monitor V1 Node

We will not go into the use of the EUI in a lot of detail here, since it is featured in each of the scenarios in Chapter 7, “Systems Monitor Examples” on page 161. However there are a few general points to be aware of when using the end-user interface for configuring Systems Monitor nodes.

1. Always remember to *enable* the table entry.

   It is very easy to get carried away with all the other fields and forget to enable the entry; you will then be surprised to find that nothing works as you expected.

2. Use the Context-sensitive Help button.

   This provides helpful information on all the fields in the table. Simply click on the button, move the ? prompt that appears to the field you want explained, and depress the mouse button.

3. Check the messages box at the bottom of the panel for messages.

   If you have changed a variable, and the messages do not indicate that the set was successful, the change has not been made.

4. Always remember to save configurations that you want to keep.

5. Beware of minimized windows.

If you select a table and then click on the **View/Modify** button, and nothing happens, check that the table is not already open and minimized.

2.7 Migration from Systems Monitor Version 1 to Systems Monitor for AIX Version 2

If you already have a previous version of Systems Monitor installed, you will want to use your current configurations on Systems Monitor for AIX (the new version). The System Monitor for AIX provides a migration tool for this purpose. This migration tool converts configuration files for Systems Monitor (V1.1 or V1.2) into the format for Systems Monitor for AIX.

2.7.1 Migration from System Monitor V1R1 to V1R2

Systems Monitor V1R1 and V1R2 use the same logic. (System Monitor V1R2 supports Kanji characters, but the logic is unchanged.) Therefore, the configuration files under the /usr/adm/sm6000/config directory have the same format in each case. When you migrate from V1R1 to V1R2, you only need to save the V1R1 configuration files (/usr/adm/sm6000/config) and restore these files on the /usr/adm/sm6000/config directory of the Systems Monitor V1R2 node. So, our recommended procedure for migration from V1R1 to V1R2 is as follows:

1. Back up the /usr/adm/sm6000/config directory of the V1.1 node.
2. Remove V1.1 software from the V1.1 node using SMIT.
3. Install V1.2 software to the target node.
4. Restore the files in the /usr/adm/sm6000/config directory.
5. Check the result of the migration.

At step 2 in the above procedure, an alternative method is simply to install V1R2 as replacement software for V1R1. The result of these two migration procedures should be the same. However, we recommend backing up the V1R1 configuration first, in case of problems.

2.7.2 Migration from System Monitor V1 to V2

System Monitor for AIX V2 is totally different from the previous System Monitor/6000 (V1), both in program structure and capabilities. Therefore, the format of the configuration files is different between System Monitor V1 and V2. To resolve this problem, the Systems Monitor for AIX V2 provides a shell script to convert the configuration files from the older version.

2.7.2.1 Using the Migration Tool

The migration tool is in two parts, one for the SIA and one for the MLM. The tool is in fact a shell script, located as follows:

Table 1. Location of the Migration Shell Scripts

Installation Image	Tool Location
smsia.subagent.obj	/usr/lpp/smsia/original/siaConvertCfg
smmlm.subagent.obj	/usr/lpp/smmlm/original/mlmConvertCfg
smcfg.eui.obj	Not Included

To use the shell scripts, you simply have to specify a Systems Monitor V1 configuration file, or directory containing configuration files. Note that these configuration files reside on the agent node. For example:

```
# siaConvertCfg V1_conf_file_name (or V1_conf_directory_name)
```

Figure 50. Syntax of the Migration Shell Script

2.7.2.2 Migration Experience

We migrated from System Monitor V1.2 to Systems Monitor for AIX V2 using the `siaConvertCfg` command. We found no problems with the migrated configuration profiles. However we recommend backing up the V1.2 profiles prior to performing the migration, in case you wish to revert to the previous version, or check for migration problems.

2.7.2.3 Summary of the Migration Path

The following table is a summary of the various migration patterns of Systems Monitor:

Table 2. Migration of the System Monitor Configuration Files

Version / Release	Description	Migration Method
System Monitor V1R1 → V1R2	Files Accepted	Not Necessary
System Monitor V1R1 → V2R1	Files Modified	Provided Migration Shell (Manual Process)
System Monitor V1R2 → V2R1	Files Modified	Provided Migration Shell (Manual Process)

2.8 Preparing NetView for AIX Version 4 to Work with AIX Systems Monitor/6000 V2

To gain full advantage of the new functions implemented in NetView for AIX Version 4, you need to prepare NetView for AIX to use the AIX Systems Monitor/6000 capabilities. This includes the following:

- Setting daemon options with the help of SMIT
- Registering daemons to the NetView for AIX main application, ovw

2.8.1 Setting Daemon Options

You can use SMIT to configure NetView for AIX. To go directly to the NetView for AIX menus, enter the SMIT fast path `smit nv6000`. Then select **Configure**, then **Set options for daemons**, then **Set options for topology, discovery, and database daemons**, and then **Set options for netmon daemon**.

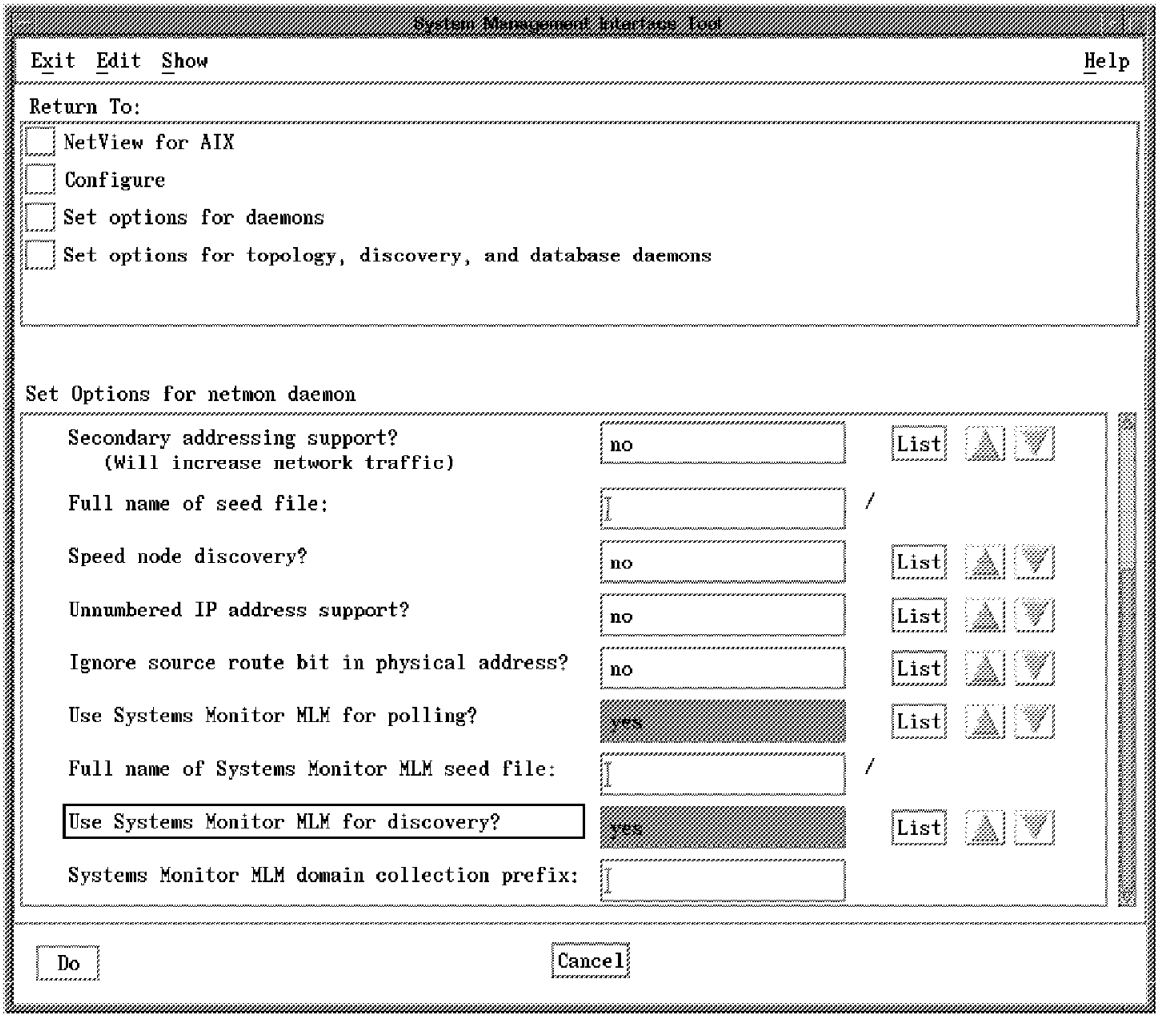


Figure 51. Parameters to Configure NetView for AIX for SYSMON

Advance to the end of the displayed dialog. There you will find the AIX Systems Monitor/6000 related parameters:

Use Systems Monitor MLM for polling

Set this parameter to yes if you wish to exploit the MLM as a distributed manager. This will cause NetView for AIX to automatically distribute status checking to the MLM. The netmon daemon can distribute status checking to MLMs in either of the following two ways:

- Automatically distribute status checking
- Using MLM domain collections

If a network segment contains more than one node running MLM and no collections are defined, NetView for AIX will distribute all the nodes in the network segment equally to the MLMs. This means, if two MLMs will be discovered by NetView for AIX in a subnet which contains 80 more nodes, NetView for AIX will assign 40 nodes to each MLM. This behavior of netmon changes dramatically if you use MLM domain collections together with the nvcold daemon. We discuss how NetView for AIX uses MLMs in detail in Chapter 4, “Using the Mid-Level Manager for Status Polling” on page 109.

Full name of Systems Monitor seed file

Specifies a file which contains IP addresses of MLM nodes. If you specify this file, NetView for AIX will use only the nodes in the seed file for status checking.

Use Systems Monitor MLM for discovery

Set this parameter to yes. This will cause NetView for AIX delegate node discovery to the MLM. NetView for AIX will not discover nodes in the network where the MLM resides, as long as the MLM is active; instead it reads the MLM discovery tables.

Systems Monitor MLM domain collection prefix

The last entry in the dialog is of some interest if you use collections to distribute management information to MLMs using APM. NetView for AIX automatically creates collections for discovered MLMs. By default, these collections have a prefix of mlmDomain_.

If you provide a name in this field, all netmon-generated MLM collections will be prepended by this prefix.

These configuration steps will set NetView for AIX to normal behavior. However you can modify this normal behavior by specifying collections (see chapters 1.3.2, “Agent Policy Manager in NetView for AIX Version 4” on page 19 and 1.3.4, “Object Collection Facility of NetView for AIX Version 4” on page 21). Changing the way NetView for AIX discovers the network and performs status checking is important if you use MLM to check nodes outside the MLMs subnetwork or, in other words, for checking nodes NetView for AIX *thinks* reside outside the MLMs subnetwork. This is done by modifying the particular MLMs collection as defined within NetView for AIX.

One reason for doing this would be if NetView for AIX is supposed to manage networks connected via interfaces that don’t support ARP and we choose to have the MLM discover and status poll particular nodes or subnetworks.

Chapter 3. The System Information Agent

This chapter provides more detail about the Systems Information Agent. The SIA has two main roles:

- It provides instrumentation for the agent node - information about resource and process utilization and configuration.
- It provides an extensible agent capability, by means of the Command Table and File Monitor Table.

We will deal with each of these roles in turn. First we discuss the MIB variables that are available in the Systems Information Agent MIB, and list those that are the most useful. Later in this chapter is an introduction to the Systems Information Agent tables.

3.1 Understanding SIA MIB Information

The System Information Agent (SIA) has a lot of useful information for managing a system. All of this information is defined as MIB extensions, so any SNMP manager may access it. There are over 600 MIB objects defined in the SIA MIB, subdivided into the following components:

- Program Information
- System Information
- Command Table
- File Monitor Table

We will discuss the Command Table and File Monitor Table in more detail later (see 3.3, “Command Table” on page 79 and 3.2, “File Monitor Table” on page 75).

In our environment, NetView for AIX can retrieve SIA MIB information directly, using the MIB Browser or menu bar applications. Alternatively the information can be used effectively by the threshold polling capabilities of the Mid-Level Manager (MLM). Figure 52 on page 66 explains these possibilities.

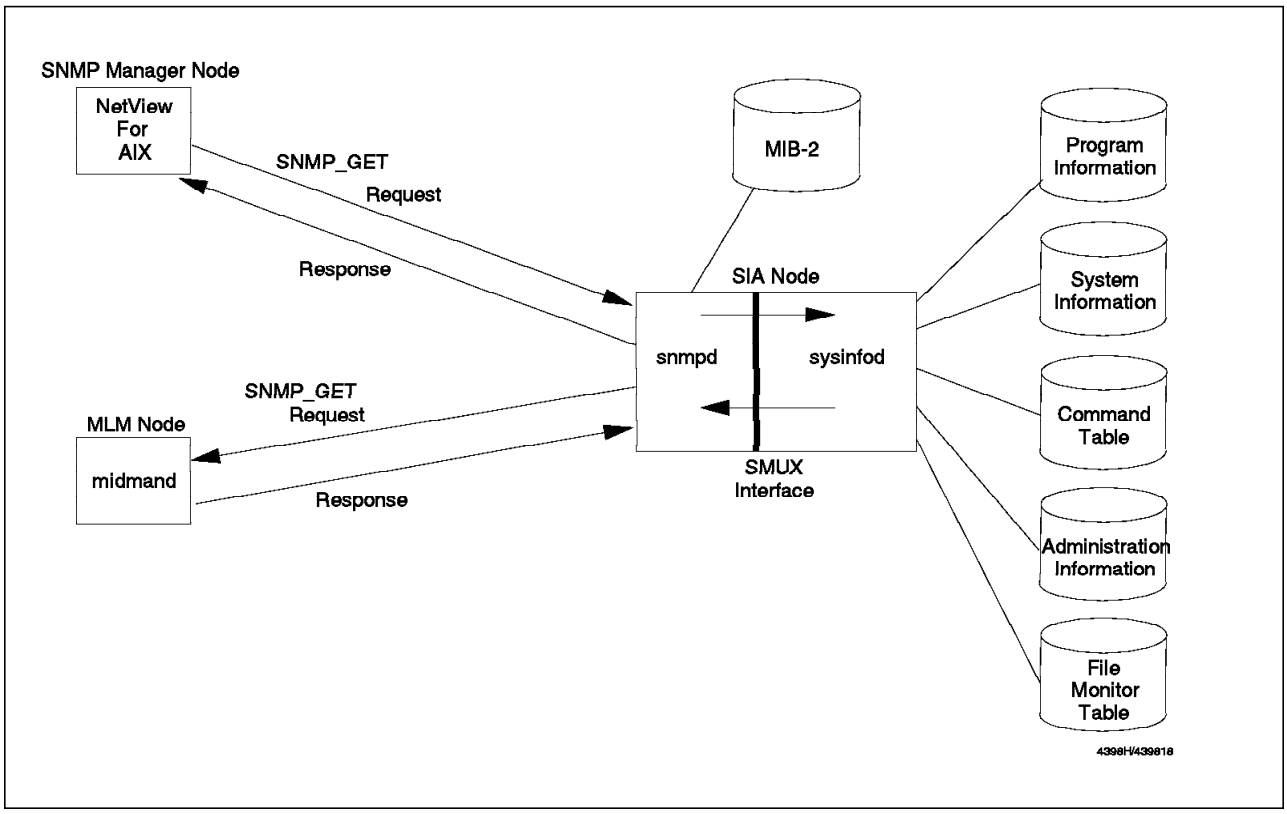


Figure 52. Role of the System Information Agent

As we have said, the SIA agent will provide us with a lot of information which we can process in various ways. However if we want to manage the SIA node effectively, we need to understand what pieces of this information are most useful.

In the following paragraphs we highlight some of the more useful parts of the SIA MIB and give suggestions of threshold values you could use to monitor them against. We also provide the summary of the complete SIA MIB in Appendix A, "SIA MIB Field Meanings" on page 305.

3.1.1 The System Information Agent MIB Extension

This part of the SIA MIB provides many objects to help manage systems in detail:

```
.iso.org.dod.internet.
                private.
                  enterprise.
                    ibm.
                      ibmprod.
                        systemMonitor6000.
                          smSiaSystemInformation.
```

Figure 53. The System Information MIBs Location in the MIB Tree

This MIB branch has the following sub-branches:

- System Description
- System Configuration Information

- System Device Information
- System Paging Information
- System File System Information
- System Subsystem Information
- System Process Information
- System User Information
- System Utilization Information
- System Miscellaneous Information

In the following paragraphs, we will highlight some of the problem areas that you may want Systems Monitor to help you with, and suggest SIA MIB objects to monitor:

3.1.2 Case 1: Performance Monitoring

In general, we use System Monitor/6000 for monitoring performance of an agent node. In this case, we consider four typical performance areas. In each area, we consider the most useful SIA MIB for management.

- CPU
- Disk
- Paging space
- Network

3.1.2.1 CPU Utilization Monitoring

It is important to know the system activity status and utilization. CPU utilization monitoring provides some useful information. For example we can learn when the system is under peak load, and which applications are contributing most to the load.

If the CPU utilization indicates 100%, it may not necessarily indicate a problem. All it shows is that all the available resource is being used, which may be leading to delays for CPU-dependant applications. If the high utilization is only for a short period it may be acceptable. A more useful measure may be the long-term trend for CPU utilization. For example, if we see the figure growing over time from 50% average utilization to 70%, we know that either a new application is exercising the system or an old application is working harder.

To help with short- and long-term monitoring, Systems Information Agent provides the following collection intervals as SIA MIB instances:

15 Seconds Instance 1

30 Seconds Instance 2

1 Minute Instance 3

5 Minutes Instance 4

15 Minutes Instance 5

1 Hour Instance 6

The following table shows some of the most useful MIB objects for monitoring CPU utilization, and a recommendation of an appropriate threshold value to use:

Table 3. CPU Utilization Monitoring MIB Objects and Thresholds		
Object Name	**Object ID**	**Suggested Threshold**
smSiaSystemUtilizationCPUUser	.1.3.6.1.4.1.2.6.12.2.9.1.3.1.4	30
smSiaSystemUtilizationCPUSystem	.1.3.6.1.4.1.2.6.12.2.9.1.3.1.5	30
smSiaSystemUtilizationCPUBusy	.1.3.6.1.4.1.2.6.12.2.9.1.3.1.8	95
smSiaSystemProcessCPU	.1.3.6.1.4.1.2.6.12.2.7.2.1.18	10

Notes on the above MIB objects:

smSiaSystemUtilizationCPUUser

This object indicates average percent CPU in user mode over the interval (CPU utilization may be divided into *system* and *user* modes. Generally speaking, the system component is the time spent processing kernel code, system calls, I/O requests and operating system functions. The user component is the time spent in program code). The same information can be obtained locally from the `vmstat` or `iostat` commands. Our suggested threshold of 30% should be applied to a longer-term sampling instance, five minutes or more, since it is normal for the figure to peak up to 100% for several seconds on a system with many users and processes.

smSiaSystemUtilizationCPUSystem

This object indicates average percent CPU in system mode over the interval. As for the user case (above), the same information can be obtained locally from the `vmstat` and `iostat` commands. The same argument applies to the threshold, too: monitor the five minute, or longer, interval.

smSiaSystemUtilizationCPUBusy

This object is the total percent CPU utilization, and as such is equal to the two previous objects (user % and system %) added together. The recommended threshold of 95% applies to relatively short intervals (one minute or less). For longer intervals a suitable value will depend on the type of work the machine is doing.

smSiaSystemProcessCPU

This object is from a different part of the MIB tree to the total system values listed above. It is in the smSiaSystemProcess table, which contains information about each process running in the system. The scheme for allocating instances in this table is different too. The instance ID is the name of the process, converted to ASCII, with the process ID appended. So for example, if we had a process called foobar, which was process ID 54321, the MIB instance used in the process table would be 102.111.111.98.97.114.54321.

The smSiaSystemProcessCPU indicates percentage of CPU used by each process. The same information is available locally from the `ps aux` command. Unlike the system-wide values, which have several sampling intervals, this object returns the average percentage utilization since the last

time it was polled. Therefore, to eliminate short-term utilization spikes, it should be polled on a five minute or longer interval. The recommendation (no process using greater than 10% CPU) would need to be modified depending on what the system was being used for. We show a practical example of using the Threshold Table to monitor this MIB object in 7.3.6, “Monitoring Excessive CPU Utilization for Processes” on page 242.

3.1.2.2 Disk Space Monitoring

File system free space monitoring is important. If there is no free space in a file system, some fatal error may occur (such as, I/O error, system hang, etc.). In particular, /var and /tmp file systems are used for work areas, therefore if there is not enough free space on these file systems, applications may fail. The following SIA MIB objects can help us do effective management of file system space:

Table 4. File System Space Monitoring MIB Objects and Thresholds

Object Name	Object ID	Suggested Threshold
smSiaSystemFileSystemFree	.1.3.6.1.4.1.2.6.12.2.5.2.1.3	1024
smSiaSystemFileSystemPercentUsed	.1.3.6.1.4.1.2.6.12.2.5.2.1.4	90
smSiaSystemFileSystemInodesPercentUsed	.1.3.6.1.4.1.2.6.12.2.5.2.1.6	90

All of these objects are in the smSiaSystemFileSystemTable part of the MIB tree. The table uses the name of the file system in ASCII as an instance ID. For example, if we wanted to check any of the above objects for the /var file system, we would use an instance ID of 47.118.97.114.

The above MIB objects have the following meanings:

smSiaSystemFileSystemFree

This object indicates the free file system space on the system in kilobytes. If there is not enough free space, errors might occur in some applications. So we need to monitor this SIA MIB value and if this value is less than threshold, we have to extend the file system as soon as possible. The same information is available locally using the `df` command. The suggested threshold value of 1 MB should be adequate in most cases, but may need adjusting if the system is running an application that generates files very quickly.

smSiaSystemFileSystemPercentUsed

This object indicates the percentage of the space allocated to a file system that is actually used by files. This object should be treated with caution when performing threshold monitoring, since the size of file systems varies enormously and hence the number of free bytes implied by a given percentage varies too.

The same information is available locally using the `df` command.

See 7.3.3, “Monitoring File System Utilization” on page 227 for an example using this MIB variable.

smSiaSystemFileSystemInodesPercentUsed

This object indicates the percentage of a file system's inodes used. Inodes are needed to provide directory entries for new files, so if there are no free inodes, we can't create any new files in the file system. This is typically a problem where applications generate a lot of very small files, which exhaust the available inodes before filling the space in the file system.

3.1.2.3 Paging Space Monitoring

Paging space is very important in an AIX system. If free space for paging is less than 512 pages, the AIX kernel sends SIGDANGER (signal number 33) to all processes on the system, and when it is less than 128 pages, the kernel sends SIGKILL (signal number 9) to the largest (in terms of virtual memory) processes until it gets back to more than 512 pages. This process is designed to prevent a complete system crash, but since the largest process is likely to be a very important process the effect can still be catastrophic.

Ideally, then, we want to be warned before the free space for paging approaches 512 pages. The following SIA MIB objects help us do effective management for the system paging subsystem:

Table 5. Paging Space Monitoring MIBs and Thresholds

Object Name	Object ID	Suggested Threshold
smSiaSystemFreePagingSpace	.1.3.6.1.4.1.2.6.12.2.4.1	5
smSiaSystemFreePagingSpaceUntilKill	.1.3.6.1.4.1.2.6.12.2.4.2	5
smSiaSystemPagingSpacePercentUsed	.1.3.6.1.4.1.2.6.12.2.4.4.2.1.5	80
smSiaSystemProcessDataVirtualMemorySize	.1.3.6.1.4.1.2.6.12.2.7.2.1.16	various

The first two of these objects are in the smSiaSystemPagingSpace part of the MIB tree. These are system-wide variables which therefore only have one instance ID, 0. The third object is at a lower point in the same part of the tree, in the smSiaSystemPagingSpaceTable table. This table contains information about individual paging spaces, and uses the name of the paging space in ASCII as an instance ID. For example, if we wanted to check smSiaSystemPagingSpacePercentUsed for the hd6 paging space, we would use an instance ID of 104.100.54. The last object is in the smSiaSystemProcess table (see page68 for a description of the organization of this table).

The following are the meanings of the above MIB objects:

smSiaSystemFreePagingSpace

This MIB indicates that the free paging space in the system in megabytes. If this value is less than 2 MB, AIX kernel sends SIGDANGER to all processes. So we should set a threshold value which includes a safety margin. This information is available locally using the `svmon`, `vmstat` and `lsps` commands.

smSiaSystemFreePagingSpaceUntilKill

This object indicates that the free paging space until the kill threshold in megabytes. Normally this value is one less than smSiaSystemFreePagingSpace. If this MIB value

becomes 0, AIX kernel sends SIGKILL to the largest process on the system.

smSiaSystemPagingSpacePercentUsed
This MIB indicates the percent used of a given paging space. In general, if this MIB value indicates more than 80, the performance of system might suffer. So we may want to monitor this value in be warned of this status before it occurs.

smSiaSystemProcessDataVirtualMemorySize
This object is the total virtual memory size, in kilobytes, of a given process. The MIB objects above can give us warning of when paging usage is about to become a serious problem, but often we may want to monitor key processes in the system to make sure they are not growing too large, thereby preempting the disruption that a paging shortage would cause.

3.1.2.4 Network Performance Monitoring

We can get some basic performance information about network interfaces from MIB-2 data (for example, ifInUcastPkts, ifOutUcastPkts, etc.). However, this data is mostly about the *network* side of the interface. That is, we can find out about data and errors going in and out of the interface, but we cannot see the effect this traffic is having on the internal processing of the system. In the SIA System Device Information MIB tree we can get a lot of more detailed information. This information is restricted to token-ring, Ethernet and X.25 interfaces.

For TCP/IP communications, some key indicators are the mbuf and buffer queue size. If these resources become exhausted the result may be overflows of the network interface and consequently rejected frames and retries. We will not discuss the use of mbufs and buffer queues in detail. If you would like to know more please refer to *AIX System Management Guide: Communications and Networks* (GC23-2487).

The following MIB objects may be used for network performance monitoring of a token-ring interface (there are also equivalent objects for Ethernet and X.25).

Table 6. Network Performance Monitoring MIB Objects and Thresholds

Object Name	Object ID	Suggested Threshold
smSiaSystemDeviceTokenRingTxQueHigh	.1.3.6.1.4.1.2.6.12.2.3.2.2.1.17	20
smSiaSystemDeviceTokenRingRxQueHigh	.1.3.6.1.4.1.2.6.12.2.3.2.2.1.18	20
smSiaSystemDeviceTokenRingRxQueNoMbuf	.1.3.6.1.4.1.2.6.12.2.3.2.2.1.25	0
smSiaSystemDeviceTokenRingRxQueNoMbufExt	.1.3.6.1.4.1.2.6.12.2.3.2.2.1.26	0

The above objects are found under the smSiaSystemDevice part of the MIB tree, in the smSiaSystemDeviceTokenRing table. The instance ID for this table is a single numeric digit, so if two token-ring interfaces were installed, each object would have instances .1 and .2. There is an object in the table smSiaSystemDeviceTokenRingNumber that maps this instance ID to the token-ring interface card ID (for example, if tr0 is instance .1, the .1 instance of smSiaSystemDeviceTokenRingNumber would return a value of 0).

These objects have the following meanings:

smSiaSystemDeviceTokenRingTxQueHigh

This indicates maximum transmits ever queued for the token-ring adapter. In each adapter, the default value of buffer queue size is 30. So if this MIB value exceeds 20, we should increase buffer queue size to prevent overflows. This status information can be obtained locally using the `netstat -v` command.

smSiaSystemDeviceTokenRingRxQueHigh

This indicates maximum receives ever queued for the token-ring adapter. On each adapter, the default value of buffer queue size is 30. So if this MIB value exceeds 20, we should increase buffer queue size to prevent overflows.

smSiaSystemDeviceTokenRingRxQueNoMbuf and smSiaSystemDeviceTokenRingRxQueNoMbufExt

These MIBs indicate that packet lost due to the system having exhausted mbufs and mbuf clusters. If these values are not 0, we should extend mbuf as soon as possible. mbufs are very important to network communication, because every byte that passes through the adapter has to be copied between adapter and mbuf. So a lack of mbufs causes a significant impact on the system (for example, communication application failures, lost data and potential system hangs). This status information can be obtained locally using the `netstat -m` and `netstat -v` commands.

Notes

The mbufs and mbuf cluster are pinned in real memory. If we increase mbuf size, so the rest of real memory for applications is decreased. This may result in increased paging and degraded performance. If we increase mbuf size, we should be careful to take this into account. Similarly, it may be that on a machine with little network traffic we would reduce mbufs to free up memory for application use.

3.1.3 Case 2: Process Monitoring

To understand what is happening in an AIX system we need to know about the processes that are running. The SIA MIB gives us the ability to monitor some key performance and status indicators for each process. However, if our objective is to track process status in detail (for example, Return code, Detecting abend subroutine, etc.), we may need to also use other methods (for example, Job Scheduler for AIX).

We have already (3.1.2.1, “CPU Utilization Monitoring” on page 67 and 3.1.2.1, “CPU Utilization Monitoring” on page 67) shown examples of performance monitoring for processes. Here we concentrate on monitoring for whether a process is running or not. The following is a list of some useful objects for monitoring process status.

Table 7 (Page 1 of 2). MIB Objects for Process Monitoring and Thresholds

Object Name	Object ID	Suggested Threshold
smSiaSystemProcessPID	.1.3.6.1.4.1.2.6.12.2.7.2.1.2	doesNotExist

Table 7 (Page 2 of 2). MIB Objects for Process Monitoring and Thresholds		
Object Name	**Object ID**	**Suggested Threshold**
smSiaSystemSubSystemsStatusCode	.1.3.6.1.4.1.2.6.12.2.6.2.1.5	11

The first object is from the smSiaSystemProcessTable table, which contains information about each process running in the system. The instance ID in this table is the name of the process, converted to ASCII, with the process ID appended. So for example, if we had a process called foobar, which was process ID 54321, the MIB instance used in the process table would be 102.111.111.98.97.114.54321.

The second object is from the smSiaSystemSubSystemsTable which interfaces with the AIX System Resource Controller (SRC) function. The instances used in this table are the name of the subsystem, converted into ASCII. So, for example, the MIB instance to use for the sna subsystem would be 115.110.97.

The MIB objects listed above have the following meanings:

smSiaSystemProcessPID

This MIB indicates process PID. So if we can't find out the PID of the monitored process, it means that this process died. This gives us an easy way to monitor whether a critical process is running in the system or not. The same information is available locally using the `ps` command.

smSiaSystemSubSystemStatusCode

This MIB indicates the status of the subsystem, whether active or inoperative. If the status is active, this instance returns 1, and if it is inoperative, it returns 11. For processes that are defined as AIX subsystems, this is much more convenient to monitor than the process ID, since one subsystem may be comprised of many processes. The same information may be obtained locally using the `lssrc` command.

Notes

We can also manage the status of the process using another SIA MIB smSiaSystemSubSystemStatusText. The difference between smSisSystemSubSystemStatusCode and smSiaSystemSubSystemStatusText is the type of MIB value. One is Integer (1 or 11) and the other is DisplayString (active or inactive), but the meaning of them is the same.

3.1.4 Case 3: Accounting

The Systems Information Agent MIB gives us some capability for doing simple system accounting, but if you want to do accounting managing in detail, you should use the AIX system accounting function (for example, the `runacct` command).

The following SIA MIB objects may be collected and used for accounting purposes:

Table 8. Accounting-Related MIB Objects		
Object Name	**Object ID**	**Suggested Threshold**
smSiaSystemProcessCPUTime	.1.3.6.1.4.1.2.6.12.2.7.2.1.4	6000
smSiaSystemProcessUserTime	.1.3.6.1.4.1.2.6.12.2.7.2.1.5	3000
smSiaSystemProcessSystemTime	.1.3.6.1.4.1.2.6.12.2.7.2.1.6	3000

The above objects are all in the smSiaSystemProcessTable which was described above (3.1.3, “Case 2: Process Monitoring” on page 72). The meanings of the MIB objects are as follows:

smSiaSystemProcessCPUTime

This object indicates the total CPU time used since startup. Therefore the value is a sum of CPU system time and user time. (smSiaSystemProcessCPUTime = smSiaSystemProcessUserTime + smSiaSystemProcessSystemTime)

The same information can be obtained locally using the ps aux command.

smSiaSystemProcessUserTime

This indicates the CPU time used in user mode since startup. This MIB value is a portion of the total CPU time.

smSiaSystemProcessSystemTime

This indicates the CPU time used in system mode since startup. This MIB value is a portion of the total CPU time.

Notes

If you do comparisons for MIB values which describe time periods, you should take care that the units match. The type of units are defined in the ASN.1 format for the MIB object. In the cases above, the data type of the objects are TimeTicks, so we can do direct numerical comparisons. However, if a MIB value which describes time has a DisplayString data type (for example, smSiaSystemUsersTime), we cannot do numeric comparisons. When the data type of the MIB is TimeTicks, it counts the time in hundredths of a second since 1970. They are defined in detail in RFC1155.

3.1.5 Case 4: Security Monitoring

Security is a very important aspect of systems management. The Systems Information Agent MIB gives us some information that we can monitor for security management purposes. However, to do a thorough job you should use the facilities built in to AIX itself and consider other IBM products such as Distributed SMIT, DCE and Network Security Program.

The following SIA MIB object can be useful to monitor system security:

Table 9. Security Monitoring MIBs and Thresholds		
Object Name	**Object ID**	**Suggested Threshold**
smSiaSystemUserRemoteHost	.1.3.6.1.4.1.2.6.12.2.8.2.1.5	exists

The meaning of this MIB object is:

smSiaSystemUserRemoteHost

This MIB object indicates the host name from which a remote user has logged-in. If this MIB value indicates an unknown host name, it might mean illegal access to the system. So we may want to perform a threshold check to trigger a trap when this value exists (dependent on our system environment).

Notes

The File Monitor Table (see 3.2, “File Monitor Table”) can also be used for some aspects of security management. Two of the scenarios in Chapter 7, “Systems Monitor Examples” on page 161 show example of this.

3.2 File Monitor Table

The File Monitor Table is a feature of Systems Monitor for AIX V2 for monitoring file status on the agent node. The aspects of files that it can monitor are:

- Existence - whether the file is there or not
- Attributes - file mode, size, ownership
- Contents - checking for specific strings

In the event of the File Monitor detecting changes in any of these, it can take action, such as generating a trap or a command.

The following diagram shows how these things are related.

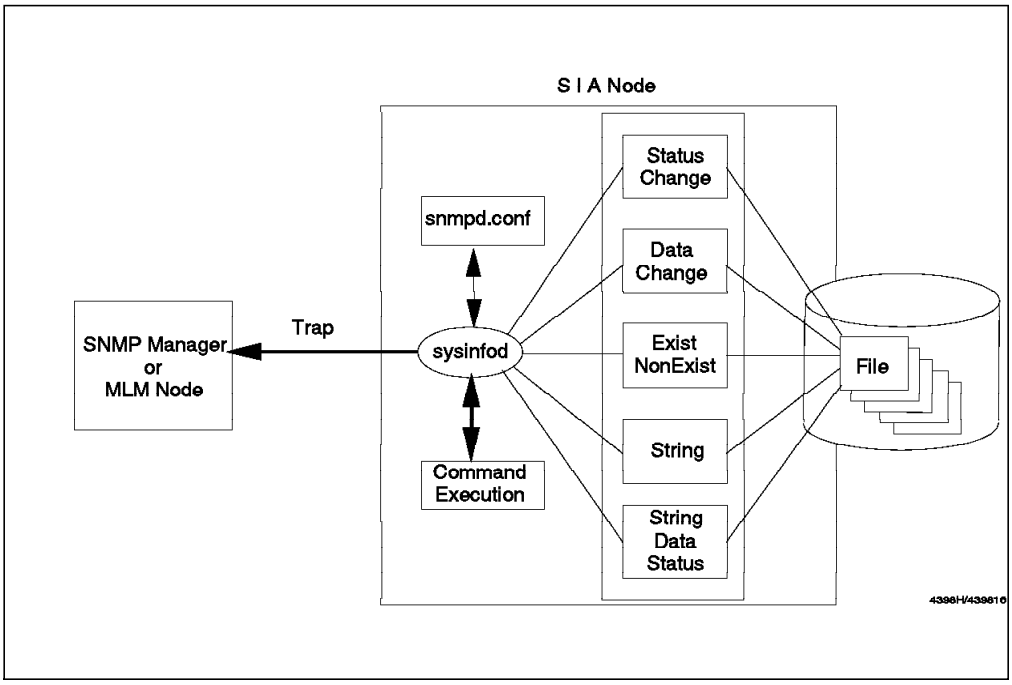


Figure 54. The Role of the File Monitor Table

3.2.1 How the File Monitor Table Differs from Other SIA Tables

The File Monitor Table is different from the other SIA tables in that it contains its own thresholding mechanism. If you want to check information from any of the other tables (including the Command Table) you have to do an SNMP GET for it. If you then want to perform threshold monitoring, you have to set up a regular SNMP poll, to retrieve the information to be checked. The Systems Monitor MLM Threshold Table, or the NetView for AIX Data Collection and Thresholding function may be used to do this.

By contrast, the File Monitor Table can perform local checking and send a trap to the SNMP manager or MLM when the monitor condition is met. For this reason you must define the appropriate destination node(s) for traps by editing /etc/snmpd.conf on the sysinfod (SIA) node.

3.2.2 Overview of the File Monitor Table

This table can be used for monitoring files, in the following different ways:

statusChange Changes in file status, including changes in file permissions, and file ownership.

An example of this can be seen in 7.2.2, “Monitoring /etc/passwd for Status Changes” on page 213.

dataChange Changes in file data.

An example of this can be seen in 7.1.6, “Monitoring /etc/hosts for Data Changes” on page 195.

exist / notExist Existence or non-existence of files.

An example of this can be seen in 7.1.7, “Monitoring /etc/resolv.conf to Verify It Exists” on page 198.

String Existence of a particular string in a file.

An example of this can be seen in 7.2.3, “Monitoring for Failed Login Attempts” on page 217.

StrDataStatus This will monitor the file for changes in file status and data changes, as well as the existence of a specified string.

Figure 55 on page 77 shows the File Monitor Table with the list of possible file monitor types described above.

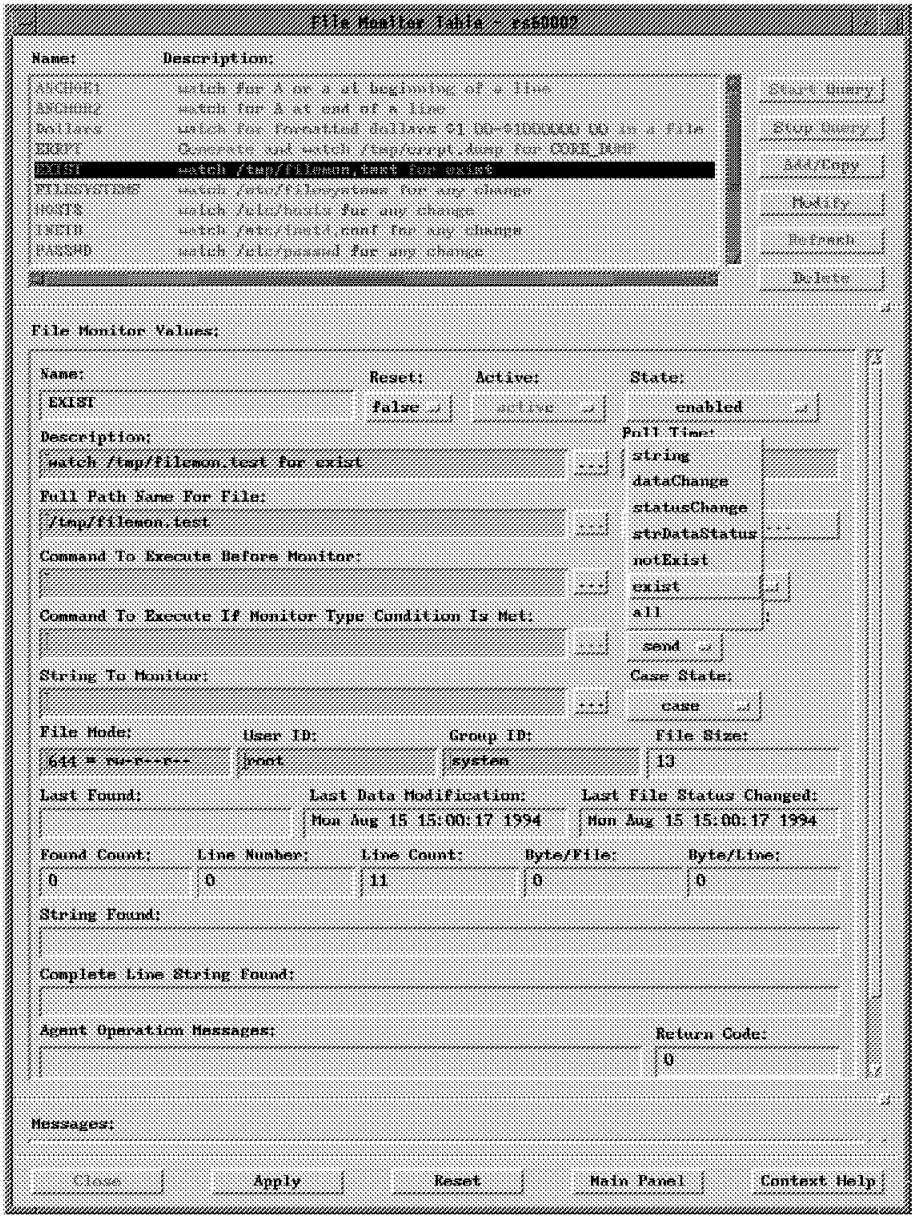


Figure 55. File Monitor Table

As is always the case in Systems Monitor for AIX, the File Monitor Table is in fact a MIB table, and the fields in the configuration screen (above) map directly to MIB objects. Figure 56 on page 78 shows the file monitor table configuration screen with an entry called EXIST, as well as the part of the MIB tree that contains the values for that entry.

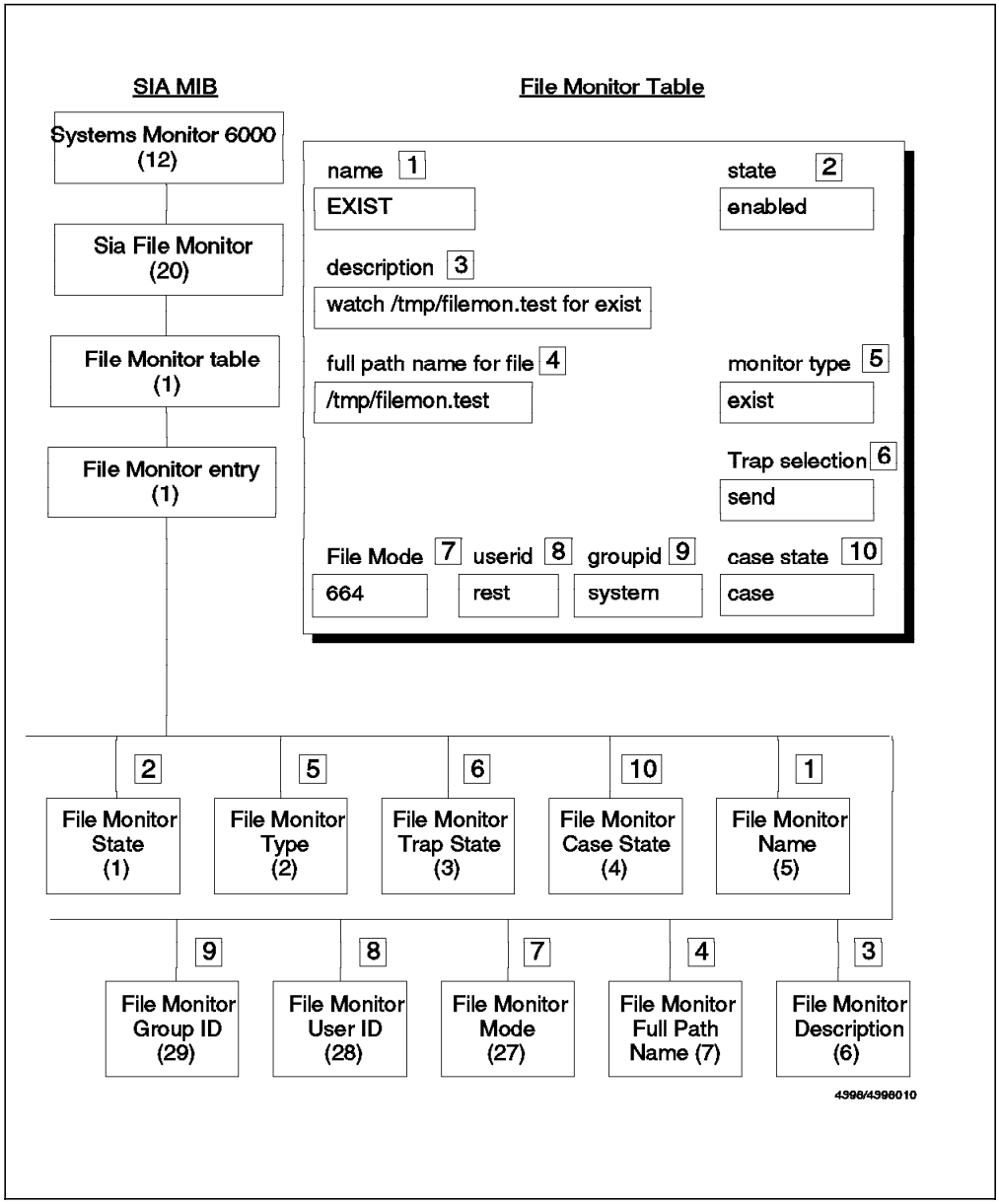


Figure 56. File Monitor Table and MIB Tree

We can see from this figure, that the name of the File Monitor Table entry (that is, EXIST) is an instance of MIB object smSiaFileMonitorName in the smSiaFileMonitor section of the MIB tree. Similarly the description in the table for that entry, can be found as an instance of the MIB object smSiaFileMonitorDescription, and so on.

The MIB instance IDs for variables in this table are the name of the table entry (in the above case, EXIST) converted into ASCII. In Figure 57 on page 79 we have used the NetView for AIX MIB Browser to query the value of the File Monitor Type for all File Monitor Table entries on the SIA agent. You can see the MIB instances in dotted-decimal notation in the MIB Values section of the screen.

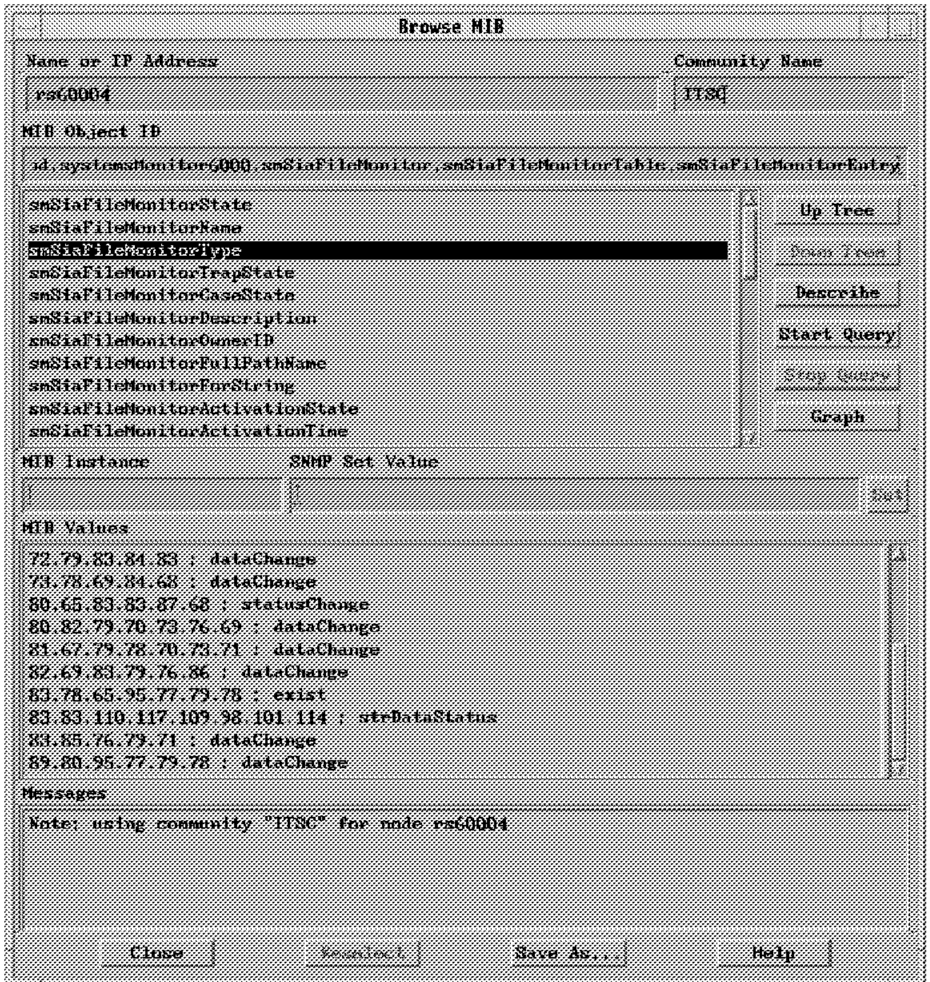


Figure 57. The File Monitor Table MIB in the MIB Browser

Using the NetView for AIX MIB Browser with the SIA

If you want to see the SIA extended MIB using the MIB browser, you need to first load the Systems Monitor for AIX V2 MIB files into NetView for AIX using the MIB loader function. If the end-user interface is installed onto an NetView for AIX system, this loading is done for you automatically. However, if you are loading the MIB manually, you may first have to unload the Systems Monitor V1 MIB before loading the V2 MIB files.

If you are planning to make use of the File Monitor Table, we recommend you read the examples in Chapter 7, "Systems Monitor Examples" on page 161, that we referred to previously.

3.3 Command Table

The Command Table is one of the most useful functions in the Systems Monitor for AIX product. Figure 58 on page 80 shows how it operates.

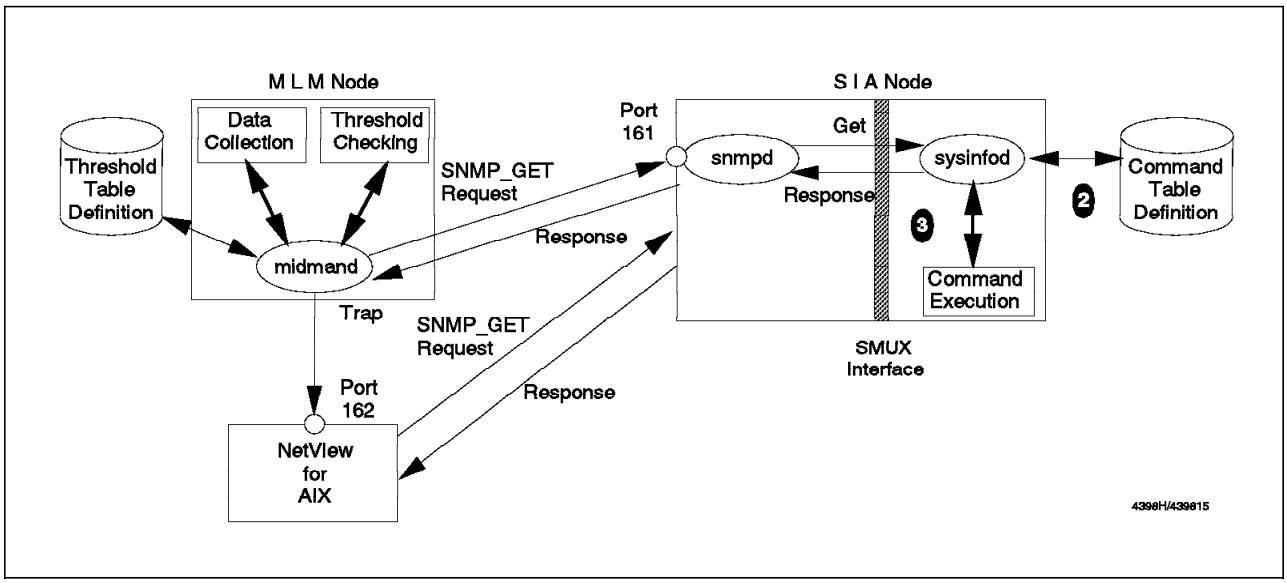


Figure 58. Concept of the Command Table

There are four steps to the Command Table operation:

1. The manager (in our case, either the Mid-Level Manager or NetView for AIX) sends an SNMP GET or SET request for a given MIB instance.
2. In the Systems Information Agent, the MIB instance is associated with a command. The SIA looks up that command.
3. The SIA opens a shell and executes the command.
4. The result of the command (stdout) is returned to the manager as the response to the SNMP GET or SET.

The overall effect is that an invocation of a command or shell script occurs as a result of querying a MIB value. Therefore, the Command Table is an extremely useful way of extending the system monitor MIB. For example, it may be that system or application data that we want to monitor is not contained within any MIB. The Command Table gives us a way to access it from an SNMP manager without complex coding.

Any line command can be invoked by an SNMP GET request, regardless of whether it is a display command or an action command. Why, then, would we invoke the Command Table with an SNMP SET? The reason is that often the command to be executed requires a variable argument. Using SNMP SET gives us a way to pass a variable to the command.

The Command Table can also be used to access shared memory. This gives us a very efficient way to add monitoring to a distributed application. It requires that the application is written to place status information in shared memory. The Command Table can then be used to access the shared memory and read the status information without the overhead of creating a new shell and executing a command.

As always with Systems Monitor, the configuration data that links the MIB instance to a command to execute is, itself, stored as a MIB table. Figure 59 on page 81 explains the relationship between the fields on the Command Table configuration screen and these MIB objects:

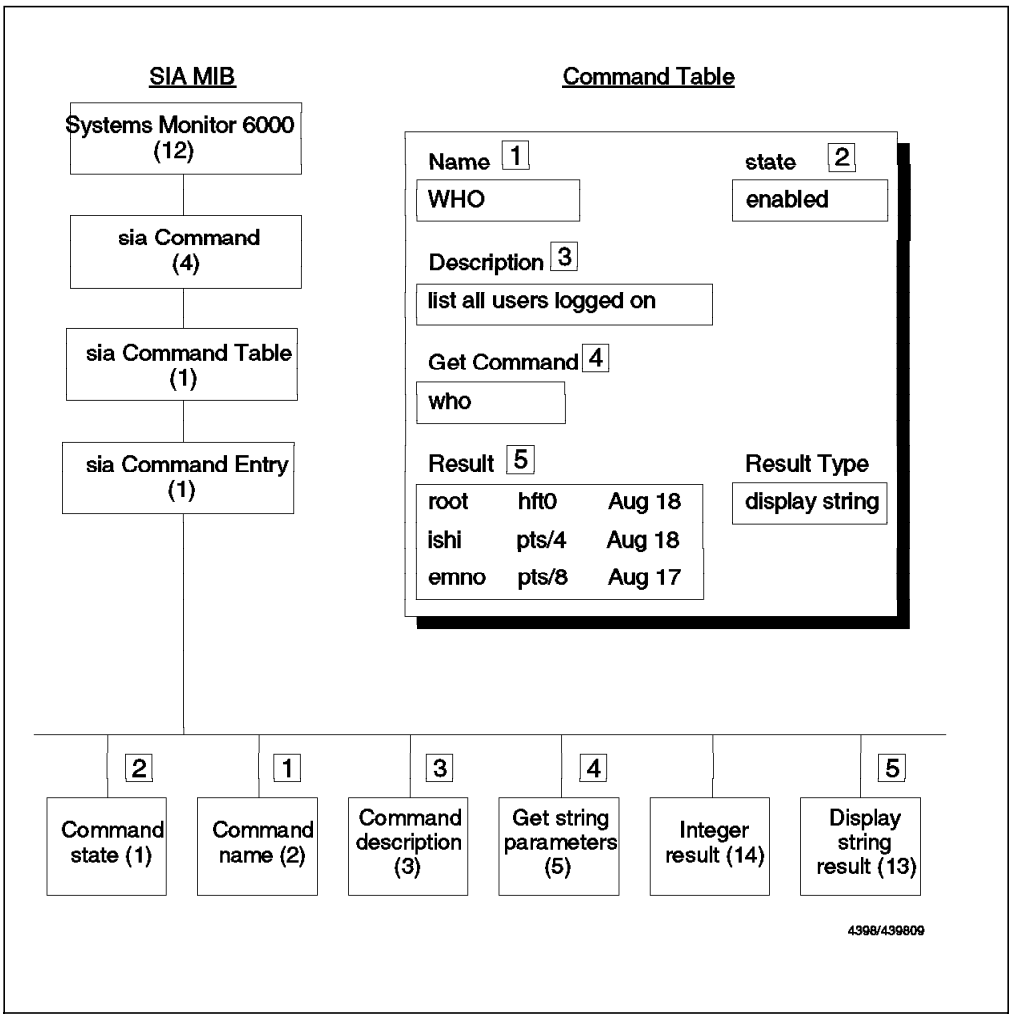


Figure 59. Command Table and MIB Tree

Referring back to the Command Table operation diagram (Figure 58 on page 80), the MIB instance that the manager performs a GET for is the Display String Result (**5**). The command associated with it is in the Get Command field (**4**). When the command has executed, stdout is passed back as the response to the original GET.

The Command Table MIB is located in the following position in the MIB tree.

```
.iso.org.dod.internet.
               private.
                 enterprise.
                   ibm.
                     ibmprod.
                       systemMonitor6000.
                         smSiaCommand.
                           smSiaCommandTable
```

Figure 60. The Command Table MIB in the MIB Tree

The instance IDs used in the Command Table are the name of the Command Table entry, converted into ASCII. So, the instance ID for entry WHO (above) would be 87.72.79. We can use the NetView for AIX MIB Browser to view the

command table entries. Figure 61 on page 82 shows an example where we have queried the smSiaCommandGetStringAndParameters object. We see that there are eight Command Table entries, each identified by their unique instance ID.

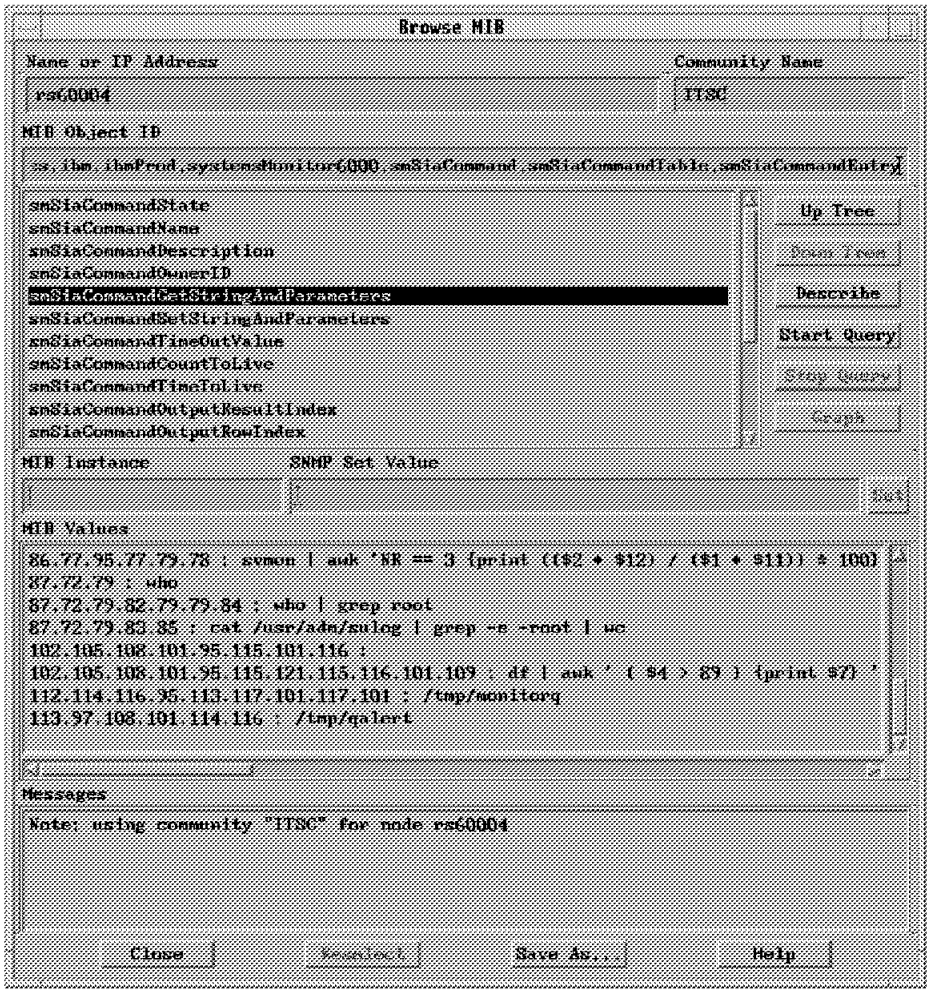


Figure 61. The Command Table MIB in the MIB Browser

We have included several examples of use of the Command Table in Chapter 7, “Systems Monitor Examples” on page 161. In particular:

- 7.1.3, “Monitoring Status of Print Queues” on page 180
- 7.1.4, “Monitoring Number of Jobs in the Queue” on page 186
- 7.3.8, “NFS Performance Monitoring” on page 250
- 7.3.4, “Automatic Response to File System Full Errors” on page 231
- 7.1.8, “SNA Session Status Monitoring” on page 201

3.4 Command Table Performance Issues

There are some performance issues surrounding the operation of the Command Table. In this section we describe the query mechanism (in particular the time-out mechanism) of the Systems Monitor for AIX Command Table.

In general, when we invoke a command from the console we get the result of the command immediately. However, when we query the result of a command which is defined in the SIA Command Table, sometimes we cannot get the result at once and a timeout occurs. Therefore it is important to set appropriate timeout values for Command Table entries.

3.4.1 Definition of Time-out Values

There are several timeout values involved when using the Systems Monitor for AIX Command Table. This is further compounded if we are using the Command Table in conjunction with the Mid-Level Manager Threshold Table (at this point you may wish to preview the function of the Threshold Table by reading 6.1.3, “Threshold and Collection Table” on page 152).

Here we discuss the role of each time-out setting:

3.4.1.1 Poll Time in the Threshold Table

We define the Poll Time in the Threshold Table as shown in Figure 62 on page 84. In this example, we define 5m (= 5 minutes) in the Poll Time field.

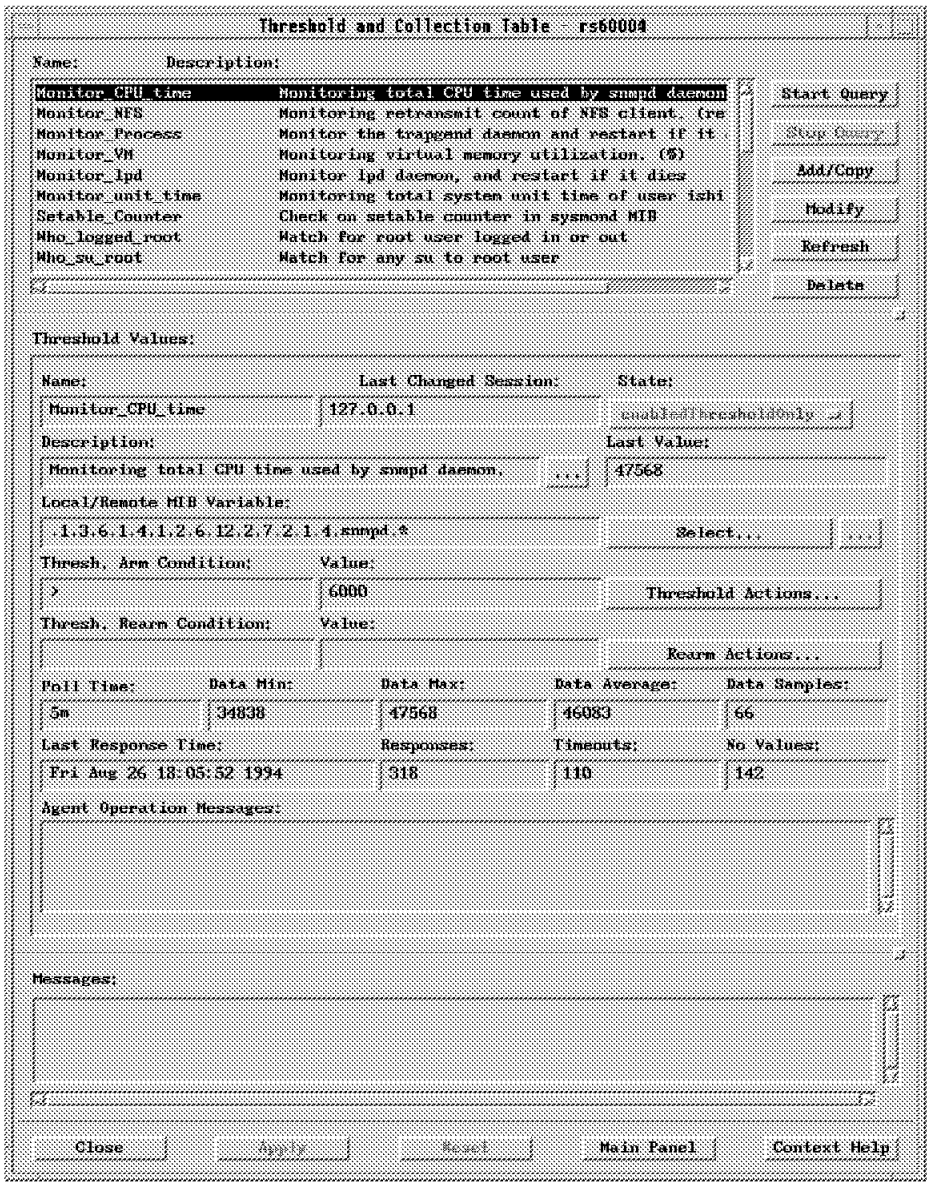


Figure 62. Definition of Poll Time

This tells the MLM to poll the given MIB instance every five minutes. That is, an SNMP GET will be sent each five minutes to retrieve the result of the Get Command in the Command Table on the SIA. If the MLM has not received a response from the polled node and it is time to check the node again, a timeout error occurs. This means that the response time is longer than the polling interval defined in the poll time field of the Threshold Table.

3.4.1.2 SNMP_GET Timeout

Whenever we invoke a Command Table function (whether using the Threshold Table, the snmpget command or any other way) an SNMP_GET request is sent to the SIA. The SNMP support built into NetView for AIX and Systems Monitor provides a timeout/retry mechanism defined in the ovsnmp.conf file:

```
rs60002.itso.ral.ibm.com:ITSC:*:8:3:::ITSC:
rs60004.itso.ral.ibm.com:ITSC:*:8:3:::ITSC:
rs60001.itso.ral.ibm.com:ITSC:*:8:3:::ITSC:
*.*.*.*:public::8:3:300:::
```

Figure 63. The SNMP_GET Timeout Definition in /usr/OV/conf/ovsnmp.conf File

The timeout field is in position 4 (colon-delimited) and retries is in position 5. Each SNMP_GET is retried N times with an exponential timeout M. So, the first try times out in M seconds, then M*2, then M*(2**2), etc. (the values in our sample ovsnmp.conf are 3 times and 0.8 seconds respectively, these are also the defaults). Figure 64 shows a representation of this behavior.

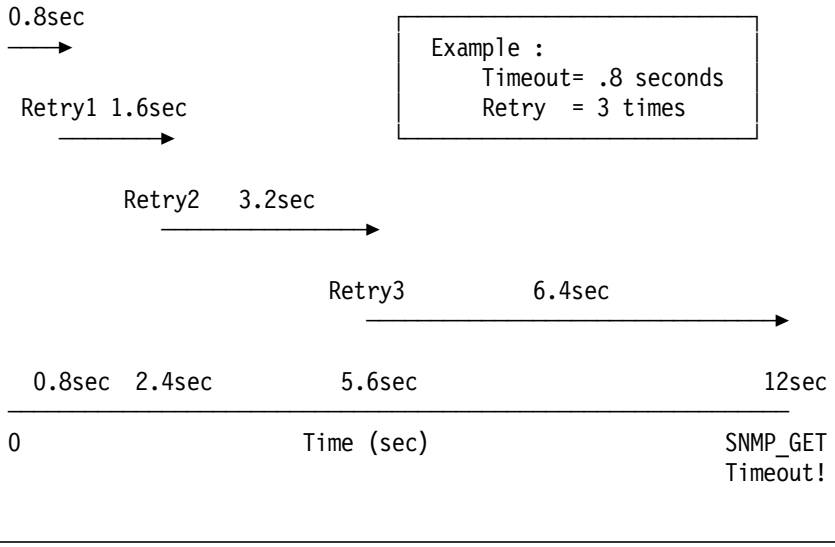


Figure 64. How to Calculate the SNMP_GET Timeout Value

So the default timeout is 12 seconds.

Refer to 2.2.2, “SNMP Configuration for SNMP Manager” on page 28 for a description of the different ways that an ovsnmp.conf file may be installed on a MLM node.

Important

If you are using the MLM Threshold Table to retrieve MIB information the SNMP_GET timeout can not be changed dynamically. To activate the new value you must stop and restart the midmand daemon. When midmand is restarted a Threshold Entry_Name initialization complete message is displayed in the Agent Operation Messages field of the threshold table configuration panel (Figure 65 on page 86). This message means that the change of the SNMP_GET timeout has completed.

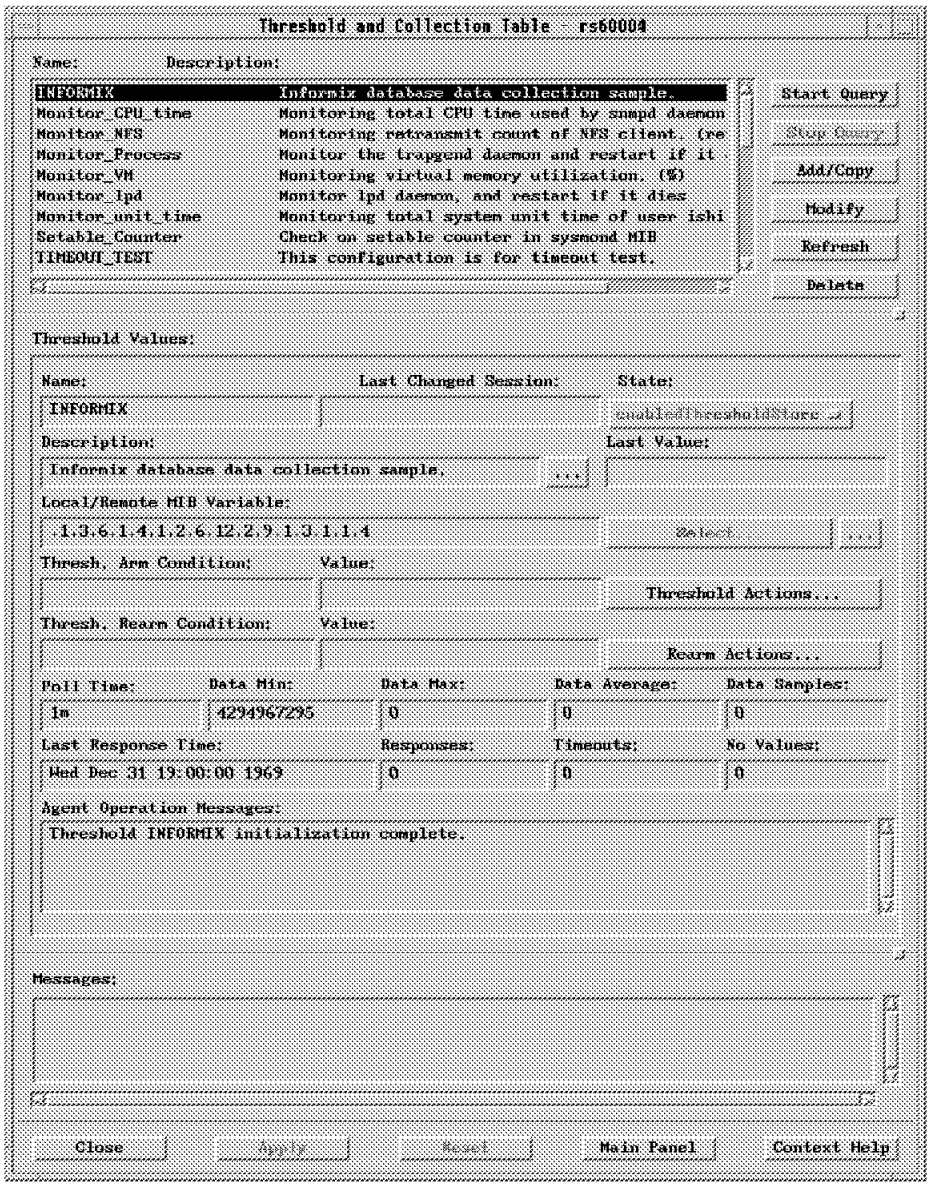


Figure 65. midmand Initialization Complete Message

3.4.1.3 smuxtimeout

As described in 1.1.6, “SMUX Protocol” on page 10, Systems Monitor is in fact two SNMP *subagents* that use the SMUX API to communicate with the snmpd agent. This API, too, has timeout features.

smuxtimeout is defined in /etc/snmpd.conf as shown in Figure 66.

```
snmpd       maxpacket=16000 smuxtimeout=60       # Systems Monitor for AIX
```

Figure 66. The smuxtimeout Definition in /etc/snmpd.conf File

smuxtimeout is applied to requests from the snmpd daemon to any SMUX subagents attached to it. The default value if nothing is specified in

/etc/snmpd.conf is 15 seconds. When the snmpd daemon receives a get request, it decides if it has to be passed to a SMUX agent based on the registration information in /etc/snmpd.conf and /etc/snmpd.peers. At this point the smuxtimeout timer starts running and if the subagent does not respond within the defined time, an error will be returned to the manager.

Updating the smuxtimeout value requires a refresh of the snmpd daemon, using the command, `refresh -s snmpd`.

When you first install snmpd (as part of the AIX base operating system) the default value of smuxtimeout is 15 seconds. However when you later install Systems Monitor for AIX the smuxtimeout is changed to 60 seconds by the install process.

3.4.1.4 Command Table Timeout

The timeout of the Command Table is defined in the Timeout field of the Command Table (actually an instance of MIB object smSiaCommandTimeOutValue) as shown in Figure 67 on page 88.

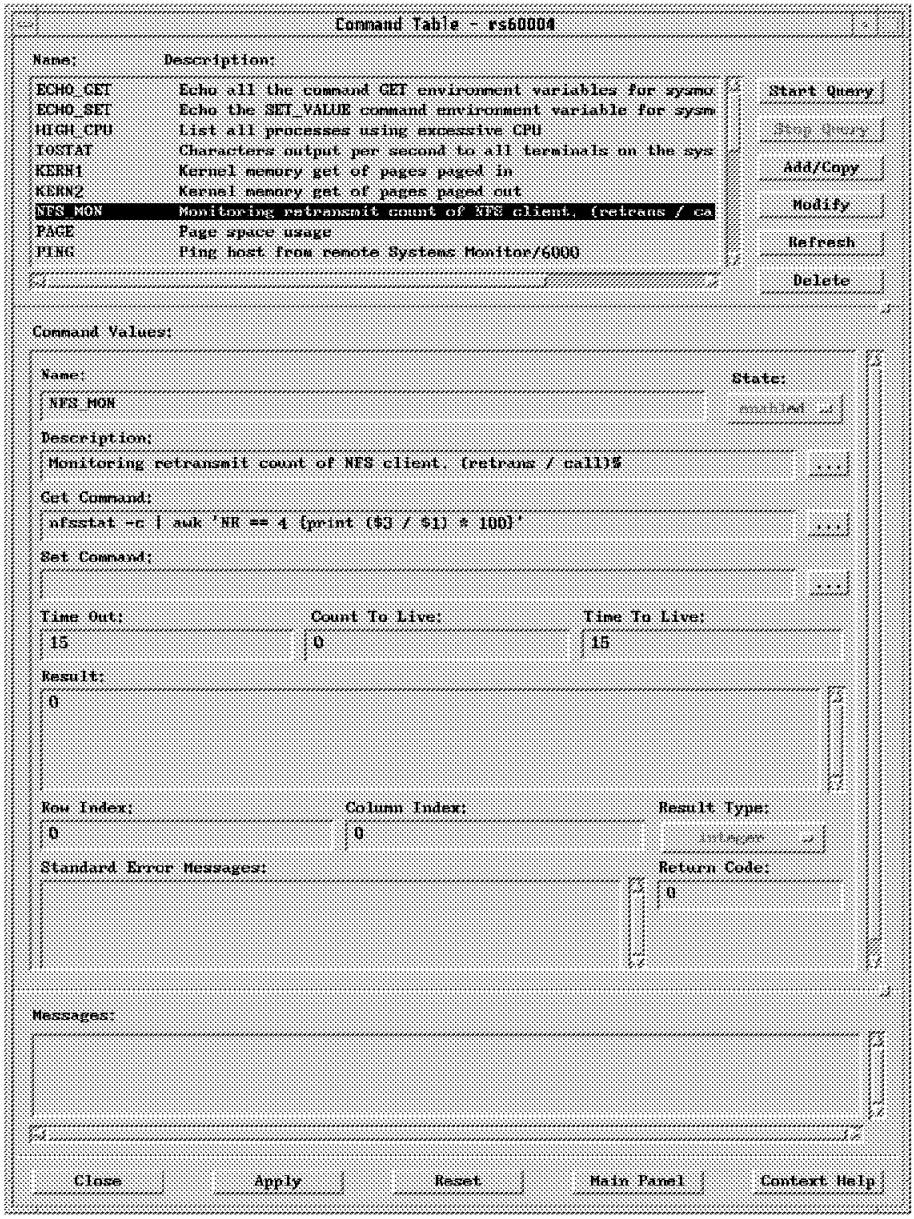


Figure 67. Definition of Command Table Timeout. The Time Out field, in the middle of the panel, is in seconds. It defines the maximum time that a command is allowed to execute for. Do not confuse it with the Time To Live field, which is the time for which the result of a command is considered valid. That is, if a second request for a Command Table entry is received within Time To Live seconds of the first one, the command will not be executed again.

In our case (Figure 67), we define 15 (seconds) to the Time Out field of the Command Table. If the elapsed time of the command execution exceeds this timeout value, a command table timeout occurs.

You may be tempted to assign long timeout values to avoid this problem. Do so with care, since only one command will be executed at once, so if you assign timeout values that are too long (>20 seconds) you may cause subagent performance degradation due to queueing.

Table 10 on page 89 is a summary of each timeout.

Table 10. The Various Timeouts

Timeout Name	Description	Location	Default Value
Poll Time	The definition of the polling interval.	Threshold Table (smMlmThresholdPollTime)	1 Minute
SNMP_GET Timeout	The definition of the SNMP_GET requests timeout. It consists of the retry times and the timeout value.	/usr/OV/conf/ovsnmp.conf	Retry=3, Timeout=0.8 seconds (12 seconds)
smuxtimeout	The definition of the SMUX requests timeout from the snmpd to the SMUX agent.	/etc/snmpd.conf	60 Seconds
querytimeout	The definition of the querying MIB-2 data timeout from the snmpd.	/etc/snmpd.conf	60 Seconds
Command Table ″Time Out″ field	The definition of the timeout for command execution.	Command table (smSiaCommandTimeOutValue)	3 Seconds

3.4.2 Understanding the Query Procedure

When we access the Command Table MIB value from the MLM threshold table, the flow is as shown in Figure 68 on page 90. Although this shows the MLM as the polling manager, the timeout characteristics would be the same if NetView for AIX were performing the poll.

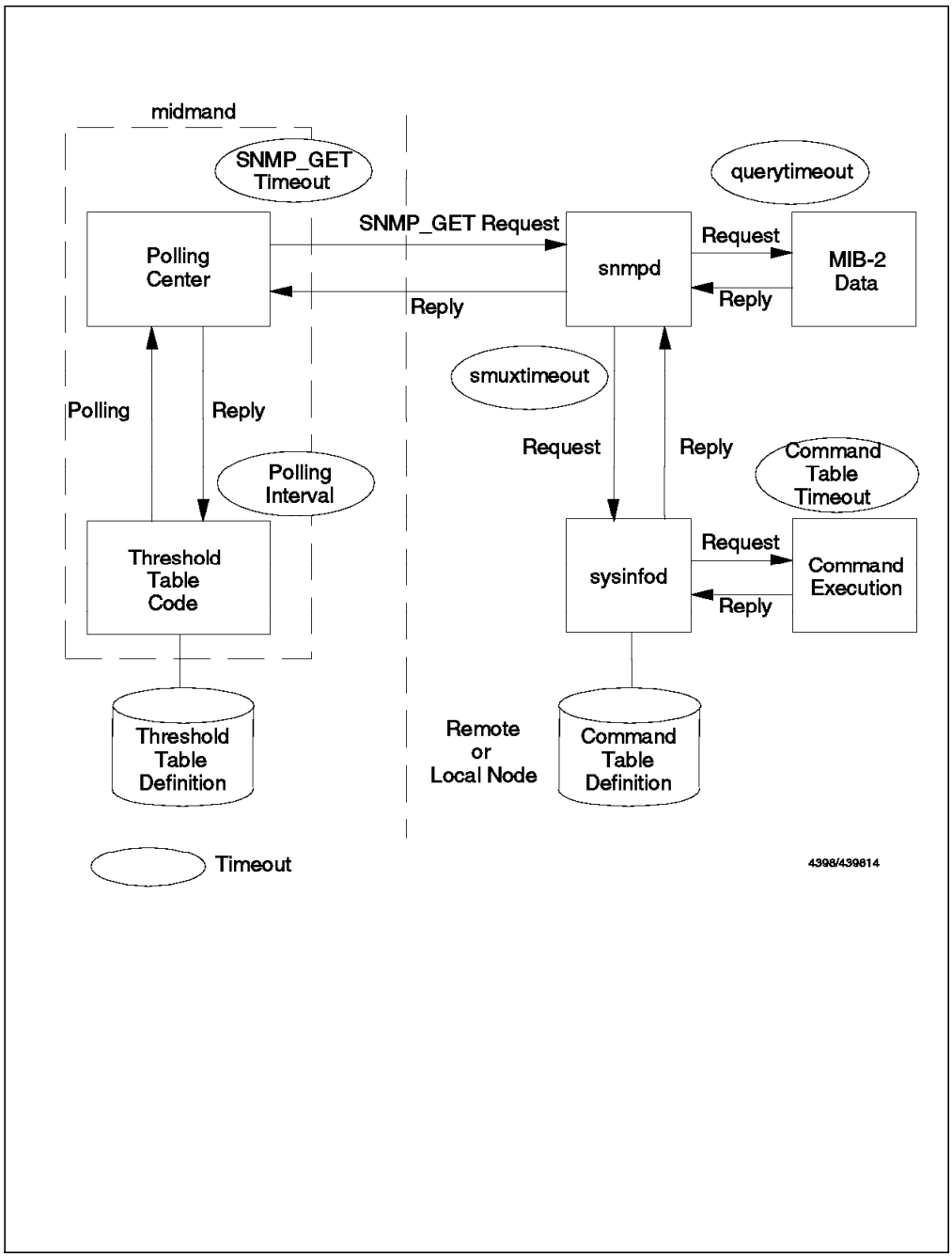


Figure 68. Flow of Querying the Command Table MIB

1. The Threshold Table code issues a poll request (via an API call) every polling interval to the *polling center*. The polling center is the SNMP support component of Systems Monitor for AIX (often called the SNMP stack).
2. The polling center issues an SNMP_GET request to the snmpd daemon of the target node.
3. The snmpd daemon of the target node decides whether it is a SMUX request or not from the /etc/snmpd.conf and /etc/snmpd.peers files. If the snmpd receives a SMUX request from the polling center, the snmpd sends the SMUX_GET request to the SMUX subagent (in this case sysinfod) using the SMUX interface.

4. The sysinfod daemon receives the SMUX_GET request from snmpd and invokes some command as defined in the command table MIB. If sysinfod can get the result of the command within the timeout interval of the Command Table, the sysinfod replies the result of the command to the snmpd daemon as the GET response. If the response of command execution has not been received in time, a Command Table timeout occurs.
5. The result of the command execution is sent across the SMUX interface from the sysinfod daemon to the snmpd daemon. If the response has not been received from sysinfod daemon in time, a smuxtimeout occurs.
6. The polling center in the MLM node receives the SNMP GET response containing the command result. If the SNMP_GET response has not been received from the polled node within the timeout period, the retry process begins.
7. When midmand receives the result of the command without a timeout the values are checked against a threshold or stored. If the response has not been received from the polled node and the time has come for the next poll, the polling too fast message is logged in the midmand.log file (see Figure 75 on page 95).

Notes

Although the Threshold Table and the polling center are described here as individual functions, in fact the polling center is a set of library routines that the Threshold Table calls. Thus we cannot see them as separate processes in the output of a `ps -ef` command.

3.4.3 Timeout Errors and the SIA Log File

There are several reasons why a timeout error may occur:

- Heavy network traffic.
- The system on which the SIA is installed is busy.
- A command defined in the Command Table needs a long elapsed time.
- The polling interval is too short.

When the timeout occurs, there may be an error message to help detect the cause of the error. For example if there is threshold table entry active on the MLM, we can query it from the configuration screen and may see the message in Figure 69 on page 92.

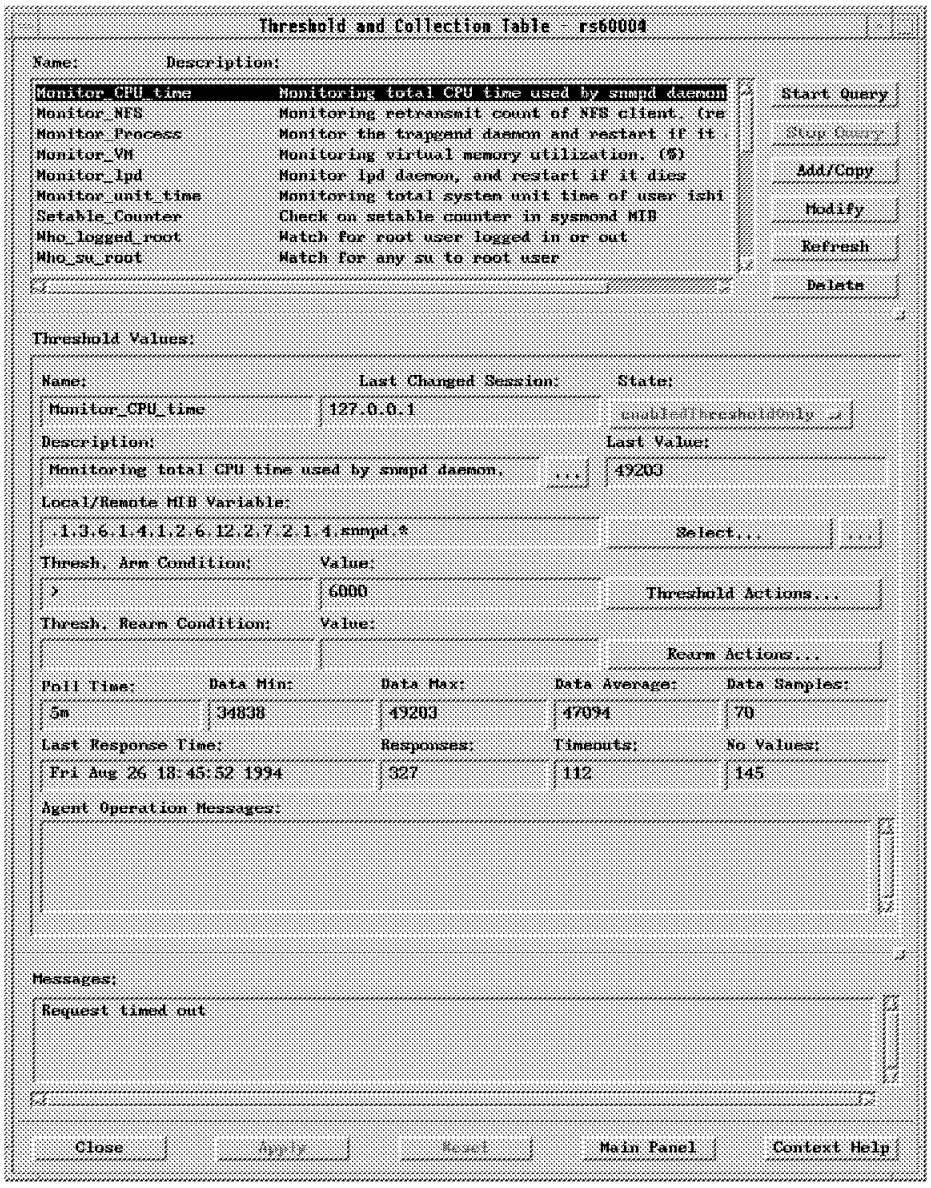


Figure 69. Timeout Error during Query MIB from the Threshold Table

The messages displayed in the Agent Operations Messages field on this screen are also logged. Messages about the result of a manual operation (for example Start Query) are displayed in the Messages field.

The Command Table configuration screen operates in a similar way; command errors and timeouts reported by the agent are displayed in the Standard Error Messages field, and messages about the result of some manual operations are displayed in the Messages field.

When timeout errors occur, they are logged in one of three log files:

- Errors detected by the snmp agent are recorded in /usr/tmp/snmpd.log on the agent system.
- Errors detected by the SIA subagent are recorded in /var/adm/sm6000/log/sysinfod.log on the agent system.

- Errors detected by the MLM subagent are recorded in /var/adm/sm6000/log/midmand.log on the agent system.

We will look at each of these log files in turn.

3.4.3.1 /usr/tmp/snmpd.log

This log file is for the snmpd daemon and has four logging levels (0-3). You change the attributes of the log file by editing /etc/snmpd.conf file. When a smuxtimeout or querytimeout occurs, they are logged in this file as shown in Figure 70.

```
7/29 11:29:33 EXCEPTIONS: noResponse after 60 seconds (SMUX 127.0.0.1+1137+3)
```

Figure 70. SMUX Timeout Log, /usr/tmp/snmpd.log. This entry means that the SMUX_GET response has not been received from the sysinfod (SMUX agent) within the smuxtimeout value of 60 seconds.

When a SMUX subagent (for example, the sysinfod daemon) is down, the snmpd daemon loses the SMUX session to it. When sysinfod is restarted it will attempt to reestablish the session. If the snmpd daemon fails, sysinfod and midmand will try to establish the SMUX session again at regular intervals, until snmpd returns.

We can see the result of sysinfod restarting in snmpd.log (Figure 71).

```
8/19 11:35:42 NOTICE: lost peer (SMUX 127.0.0.1+3298+11)
8/19 11:36:40 NOTICE: SMUX relation started with (127.0.0.1+3299+12)
8/19 11:36:40 NOTICE: SMUX packet from (127.0.0.1+3299+12)
8/19 11:36:40 NOTICE: SMUX open: 12 enterprises.2.6.12 "IBM Systems Monitor/6000" (11/ 127.0.0.1+3299+12)
8/19 11:36:40 NOTICE: SMUX packet from (127.0.0.1+3299+12)
8/19 11:36:40 NOTICE: SMUX register: readOnly 1.3.6.1.4.1.2.6.12.1.10 in = -1 out = 0 (127.0.0.1+3299+12)
8/19 11:36:40 NOTICE: SMUX packet from (127.0.0.1+3299+12)
8/19 11:36:40 NOTICE: SMUX register: readOnly 1.3.6.1.4.1.2.6.12.1.11 in = -1 out = 0 (127.0.0.1+3299+12)
8/19 11:36:40 NOTICE: SMUX packet from (127.0.0.1+3299+12)
8/19 11:36:41 NOTICE: SMUX register: readOnly 1.3.6.1.4.1.2.6.12.1.12 in = -1 out = 0 (127.0.0.1+3299+12)
8/19 11:36:41 NOTICE: SMUX packet from (127.0.0.1+3299+12)
8/19 11:36:41 NOTICE: SMUX register: readOnly 1.3.6.1.4.1.2.6.12.2 in = -1 out = 0 (127.0.0.1+3299+12)
8/19 11:36:41 NOTICE: SMUX packet from (127.0.0.1+3299+12)
8/19 11:36:41 NOTICE: SMUX register: readOnly 1.3.6.1.4.1.2.6.12.4 in = -1 out = 0 (127.0.0.1+3299+12)
8/19 11:36:41 NOTICE: SMUX packet from (127.0.0.1+3299+12)
8/19 11:36:41 NOTICE: SMUX register: readOnly 1.3.6.1.4.1.2.6.12.20 in = -1 out = 0 (127.0.0.1+3299+12)
8/19 11:36:41 NOTICE: SMUX packet from (127.0.0.1+3299+12)
8/19 11:36:41 NOTICE: SMUX register: readOnly 1.3.6.1.4.1.2.6.12.10 in = -1 out = 0 (127.0.0.1+3299+12)
8/19 11:36:43 NOTICE: SMUX trap: (0 0) (127.0.0.1+3299+12)
```

Figure 71. Log of sysinfod SMUX Session Going Down and Up

The entries in this log mean the following:

1. The sysinfod daemon went down and the SMUX session between snmpd and sysinfod was terminated at 11:36:40.
2. The sysinfod restarted and tried to connect the lost SMUX session at 11:36:40 by sending a request packet to snmpd.
3. The snmpd received registration requests from sysinfod at 11:36:40.
4. After having established the SMUX session, the sysinfod emitted the coldStart trap that we can see in the last line of Figure 71 (at 11:36:43).

This results in a coldStart event card in the NetView for AIX event display (Figure 72 on page 94).

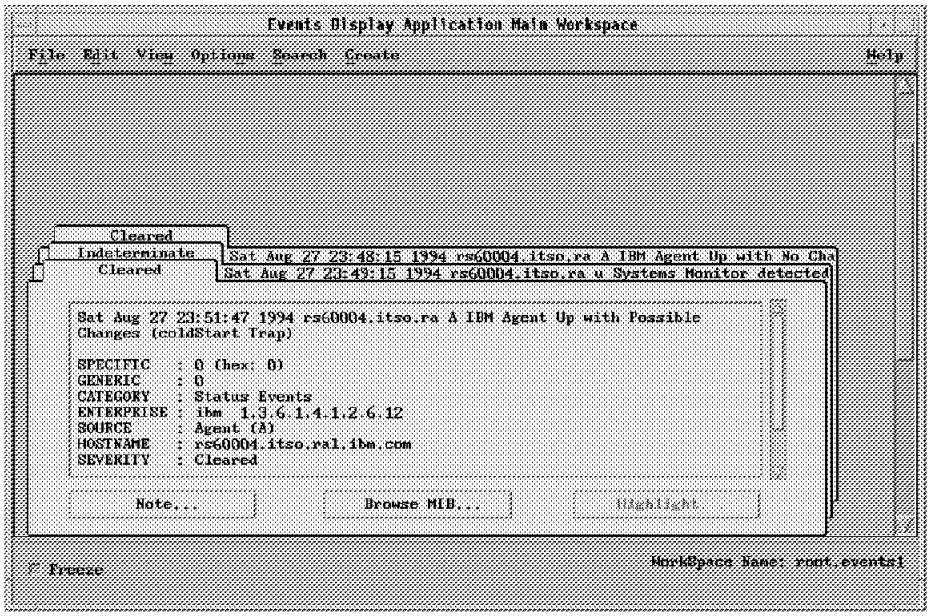


Figure 72. coldStart Trap from sysinfod

In the above case sysinfod really went down. However, the way that snmpd detected the loss of sysinfod is by waiting for the smuxtimeout value to be exceeded. This can happen, for example, when the CPU utilization of the system running the sysinfod daemon is very high (we found this happening at 95% CPU utilization). In this case, the sequence of messages in snmpd.log is almost exactly the same as above, except that the final trap that is emitted is a warm start instead of a cold start.

Notes

We got this sample of snmpd.log using the lowest logging level, 0. If you would like to get more detailed information from snmpd, you can set the logging level to 3 by modifying /etc/snmpd.conf.

3.4.3.2 /usr/adm/sm6000/log/sysinfod.log

This log file is for the sysinfod daemon. You can change the logging attributes using the System Monitor configuration (CFG) EUI. When a Command Table timeout occurs, it is logged in this log file as shown in Figure 73.

```
778019310: 08/27 16:28:30 ERROR: Command execution time out value: 3 seconds exceeded for entry name: VM_MON
778019310: 08/27 16:28:30 ERROR: Previous ERROR log messages refer to command entry name: VM_MON
```

Figure 73. Command Table Timeout Logged in /usr/adm/sm6000/log/sysinfod.log

These messages mean that the response has not been received from the command execution within three seconds. If you find this message in the log file, you should consider changing the timeout field of the Command Table entry.

In the system overload SMUX timeout situation that we described above, sysinfod also logs information about the interruption of its SMUX session to snmpd. The messages logged are shown in Figure 74 on page 95.

```
776980718: 08/15 15:58:38 process_snmp(): smux_wait(): youLoseBig (ps2pe: Error 0)
776980718: 08/15 15:58:38 SNMP disconnected - reason=4
776980718: 08/15 15:58:38 SNMP connected: systemsMonitor6000 (systemsMonitor6000)
776980718: 08/15 15:58:38 SMUX registration requested: smSiaProgramData, priority -1
776980718: 08/15 15:58:38 SMUX registration requested: smSiaResourceUsage, priority -1
776980718: 08/15 15:58:38 SMUX registration requested: smSiaProgramMessages, priority -1
776980718: 08/15 15:58:38 SMUX registration requested: smSiaSystemInformation, priority -1
776980718: 08/15 15:58:38 SMUX registration requested: smSiaCommand, priority -1
776980719: 08/15 15:58:39 SMUX registration requested: smSiaFileMonitor, priority -1
776980719: 08/15 15:58:39 SMUX registration requested: smSiaAdministration, priority -1
776980719: 08/15 15:58:39 warmstart trap emitted
776980719: 08/15 15:58:39 SMUX registration accepted: priority 0
776980719: 08/15 15:58:39 SMUX registration accepted: priority 0
776980719: 08/15 15:58:39 SMUX registration accepted: priority 0
776980719: 08/15 15:58:39 SMUX registration accepted: priority 0
776980719: 08/15 15:58:39 SMUX registration accepted: priority 0
776980719: 08/15 15:58:39 SMUX registration accepted: priority 0
776980719: 08/15 15:58:39 SMUX registration accepted: priority 0
```

Figure 74. SMUX Session Down Error Logged in /usr/adm/sm6000/log/sysinfod.log

1. The SMUX session between sysinfod and snmpd was found to be down at 15:58:38.
2. The sysinfod tried to connect the SMUX session again and the SMUX session was established at 15:58:38.
3. The sysinfod sent registration requests for all the different branches of the SIA MIB to snmpd at 15:58:38.
4. The sysinfod emitted a warmStart trap at 15:58:39.
5. The registration requests were accepted by snmpd at 15:58:39.

Notes

The reason code for SMUX session disconnection can be one of the following:

- 0 = goingDown (Normal End)
- 1 = unsupportedVersion
- 2 = packetFormat
- 3 = protocolError
- 4 = internalError
- 5 = authenticationFailure

For more detail on the operation of the SMUX protocol, refer to RFC1227.

3.4.3.3 /usr/adm/sm6000/log/midmand.log

This log file is for the midmand (MLM) daemon. You can change the logging attributes using the System Monitor configuration (CFG) EUI.

If we are using the threshold table to poll a MIB variable (for example, the SIA Command Table) and the response takes longer than the polling interval, you will see the messages shown in Figure 75 logged in the file.

```
777657853: 08/23 12:04:13 Threshold Monitor_NFS: Time out, polling too
fast, or bad community name for node rs60004.
```

Figure 75. Polling Too Fast Error Logged in /usr/adm/sm6000/log/midmand.log

If you see this message, you either need to review the complexity of the object you are polling, or you should reduce the polling frequency.

3.4.4 Resolving Command Table Timeout Problems

From the discussion above we can see that there are many ways in which timeouts may affect the operation of the Command Table. If timeouts occur frequently, you can consider the following actions:

Set Appropriate Timeout Values The default timeout values are usually adequate if the agent and MLM systems are not overloaded, and if the network has reasonable performance. However, if your configuration differs from this you should examine the timeout values. Note that with the default values, you may never see certain timeouts. For example, the smuxtimeout value of 60 seconds may be reached, but the symptom you see at the management station is of the SNMP_GET timeout (12 seconds default), because it will expire first. The SNMP_GET timeout is the one that you should look at most closely if you are working on a slow network, since its retry mechanism can cause extra traffic, which is the last thing you want if the network is already slow (see Figure 64 on page 85).

Stagger Polling Intervals If you are using the Mid-Level Manager Threshold Table to trigger Command Table functions you should try to stagger threshold polls. For example, if you are polling several MIB values at (say) five minute intervals, the MLM will be trying to issue all of the polls at the same time. This may cause congestion. It is better to use slightly varying polling intervals (four minutes, five minutes and six minutes, for example) so that the polls are more evenly spread.

Keep Commands Short When using the Command Table function, try to use commands and shell scripts which do not need a long elapsed time. Particular things to avoid are shell scripts that perform pattern matching on command output (via the `awk` command, for example).

Keep Responses Short The data transfer between Command Table (SIA) and the manager node (NetView for AIX or MLM) uses SNMP GET_RESPONSE PDUs as defined in RFC1157. These are quite complex structures, so there is overhead involved in packaging the command output and transmitting it. It is better to modify your command or shell script so that it sends short, simple replies.

Setting the SIA Nice Value If the CPU utilization of the node on which sysinfod is installed is always high, sysinfod may not be able get the processor cycles it needs to work correctly. In this situation a possible solution is to change the *nice* value of sysinfod. To reset nice you should use the `renice` command. Take care when doing this; you don't want to improve the performance of the SIA at the expense of important applications.

3.4.5 An Approach to Command Table Performance Tuning

These notes describe a practical approach to resolving timeout problems.

3.4.5.1 Step1: Checking the MLM Threshold and Collection Table

In this case we are using the MLM Threshold Table to poll a Command Table entry on an SIA machine. We can see whether timeouts are occurring from the end-user interface threshold configuration as shown in Figure 76.

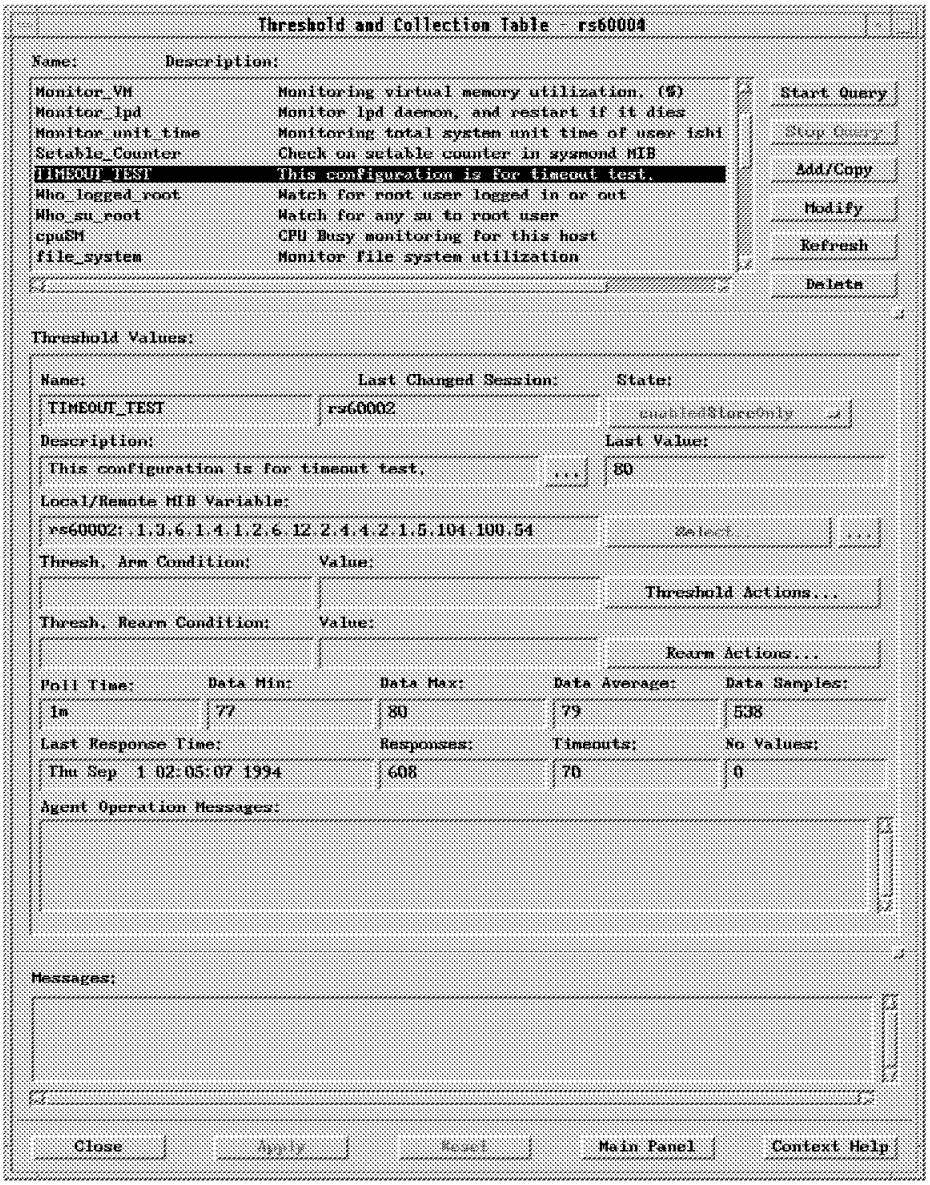


Figure 76. Total Response and Timeout Counters

In this case, the total response and timeout counters indicate 608 and 70 respectively. This is an undesirable error rate (more than 10%), so we proceed to the next step.

3.4.5.2 Step2: Checking the Log Files

As we described previously, log files are maintained on the MLM and SIA nodes in this configuration. Using the notes for each file above, we should be able to detect which timeout is the cause of our problem.

1. /usr/adm/sm6000/log/midmand.log
2. /usr/tmp/snmpd.log
3. /usr/adm/sm6000/log/sysinfod.log

3.4.5.3 Step3: Modifying a Timeout Parameter

When we detect the timeout value that is causing the problem we can proceed to modify it. In general, when you use the default values for all timeout values, the Command Table timeout and SNMP_GET timeout are the most likely to cause problems, since these values are often shorter than the required time for querying MIB data in real environments.

If the SNMP_GET timeout is too short, refer to Table 10 on page 89 to understand the retry mechanism.

To assess a suitable value for the Command Table timeout, we recommend testing the command with the `time` command. For example to test command `svmon | awk 'NR == 3 {print (($2 +$12) / ($1 + $11))*100}'` we would enter:

```
# time svmon | awk 'NR == 3 {print (($2 + $12) / ($1 + $11))*100}'
49.3467

real    0m7.74s
user    0m0.04s
sys     0m0.03s
```

Figure 77. Showing the Elapsed Time for a Command Execution

In this case, the output of the time command means that the elapsed time of the svmon command execution was 7.74 seconds.

Having obtained the elapsed time in this way we should define the timeout value as the elapsed time plus a safety margin (5-10 seconds). In this case, a suitable timeout value would be 15 seconds. We can then modify the Command Table timeout value by using the Command Table configuration screen of the Systems Monitor for AIX EUI.

Don't forget that if you alter the command execution timeout, you may also have to alter the SNMP_GET timeout value. If the SNMP GET requests are being sent from the NetView for AIX machine, you modify the /usr/OV/conf/ovsnmp.conf file using the Tools then the SNMP Configuration options from the menu bar. If the SNMP GET is being sent from an MLM node (via the Threshold Table) you have to edit the /usr/OV/conf/ovsnmp.conf file directly (Figure 63 on page 85).

3.4.6 An Example of Tuning Command Table Performance

In this section, we work through an example of how to prevent Command Table timeouts for a long-running command.

3.4.6.1 Environment

The environment of this test is shown in Figure 78 on page 99.

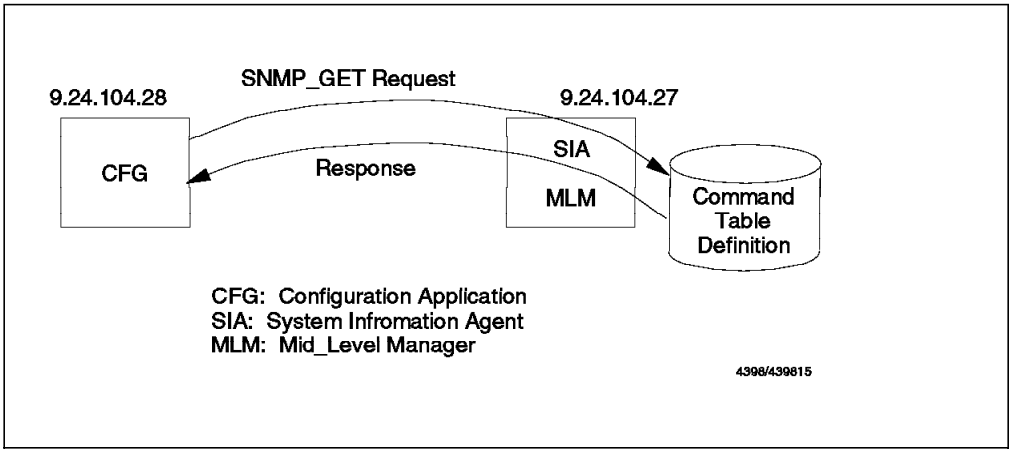


Figure 78. Test Environment for Command Table Performance Tuning

We prepared a Command Table entry for a command that needs a long elapsed time. Then we went through the tuning steps described in 3.4.5, “An Approach to Command Table Performance Tuning” on page 96, starting with default timeout values and later refining them.

3.4.6.2 Command for This Example

We used the following command for this example:

```
sleep 12; echo hello!
```

This command (clearly) needs more than 12 seconds to execute, and the default SNMP_GET timeout is 12 seconds (timeout=0.8sec, retry=3). Therefore, we expect timeout errors.

First we configured a Command Table entry for our sample command as shown in Figure 79 on page 100.

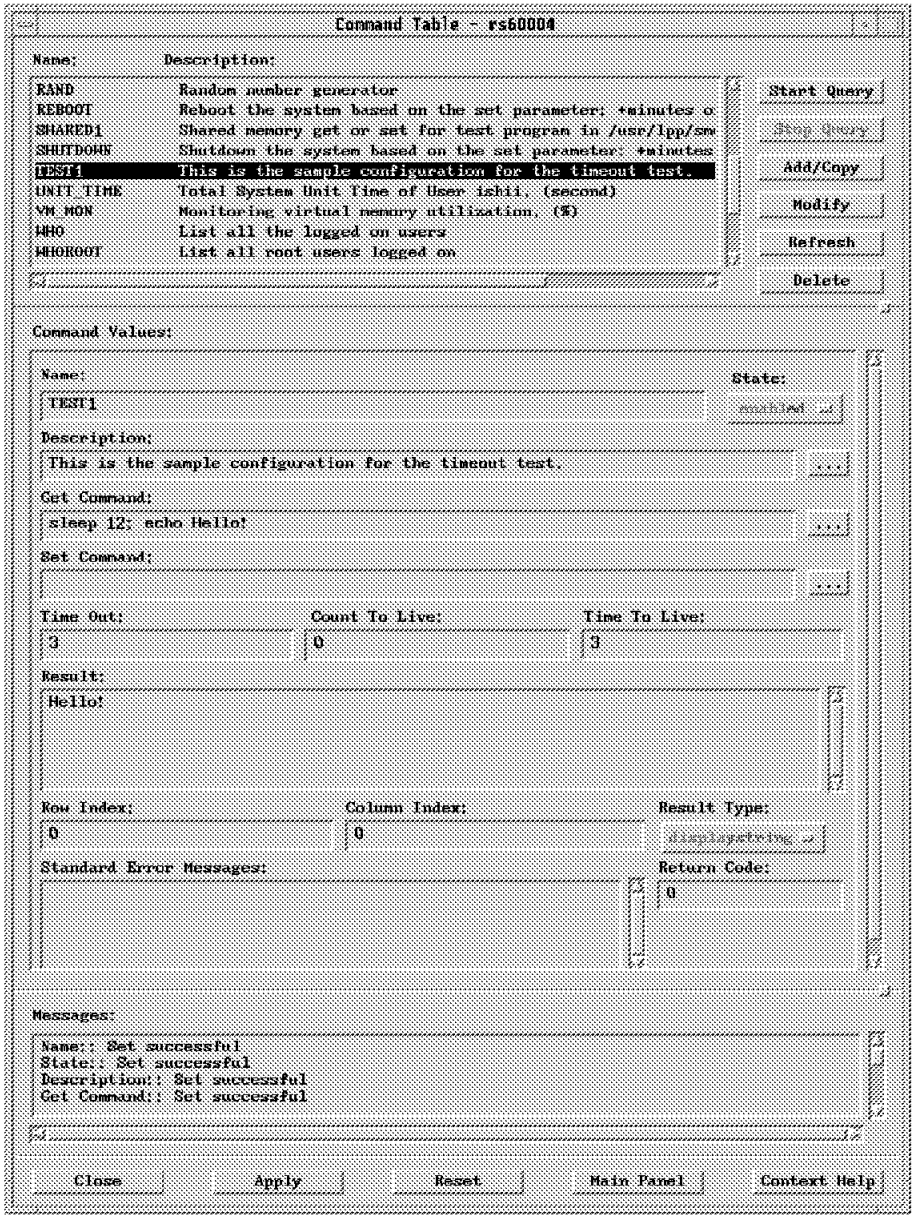


Figure 79. Command Table Configuration for This Example

3.4.6.3 Test1: Using Default Timeout Values

The default timeout value for SNMP_GET is 12 seconds and the Command Table timeout is 3 seconds. When we attempt to execute the Command Table entry from our CFG node (by selecting **Start Query** on the Command Table configuration screen), we see the result shown in Figure 80 on page 101.

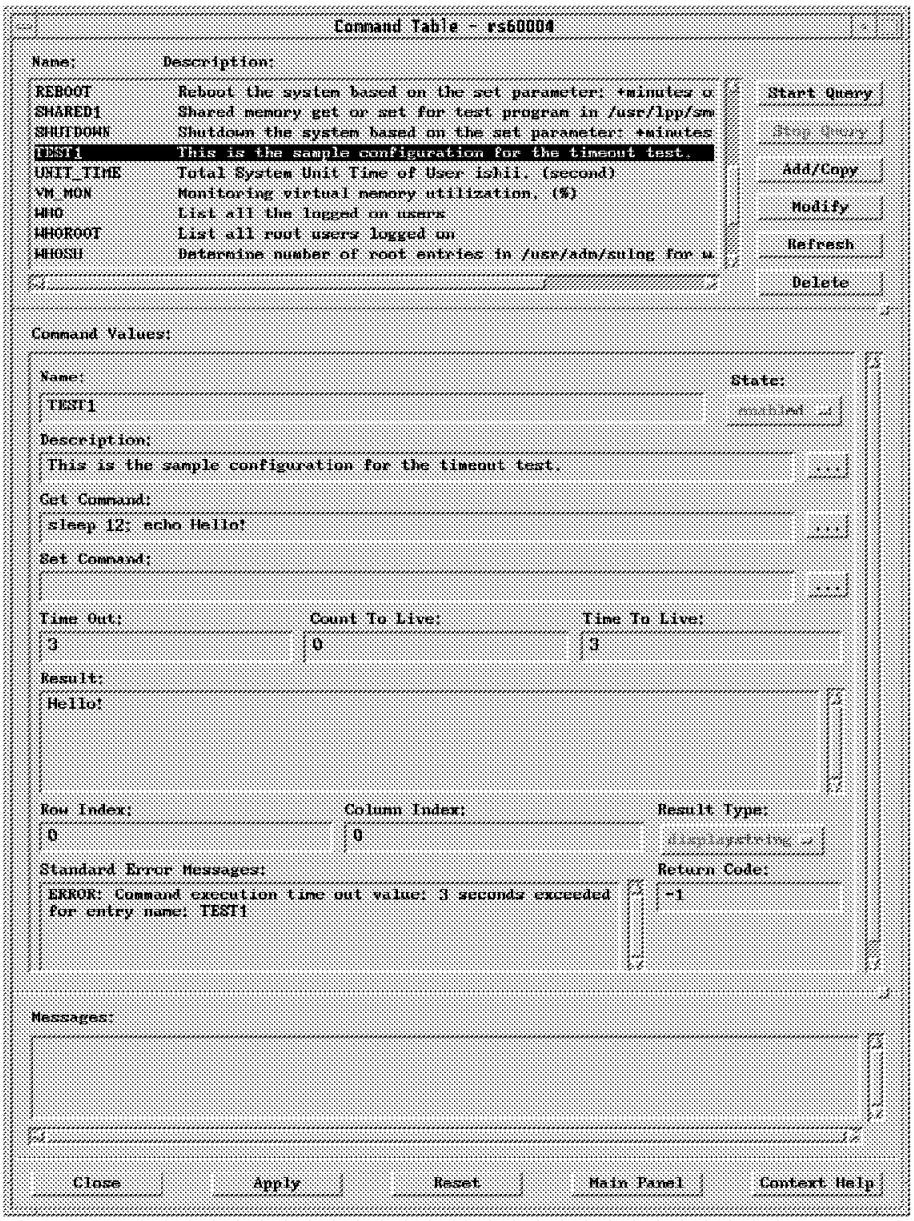


Figure 80. Command Table Timeout Messages

From this we see that the default 3 second Command Table timeout is not enough.

3.4.6.4 Test2: Modified Command Table Timeout Value

For our next step we modified the Command Table timeout from 3 seconds to 15 seconds, without altering the SNMP_GET timeout value. The result is shown in Figure 81 on page 102.

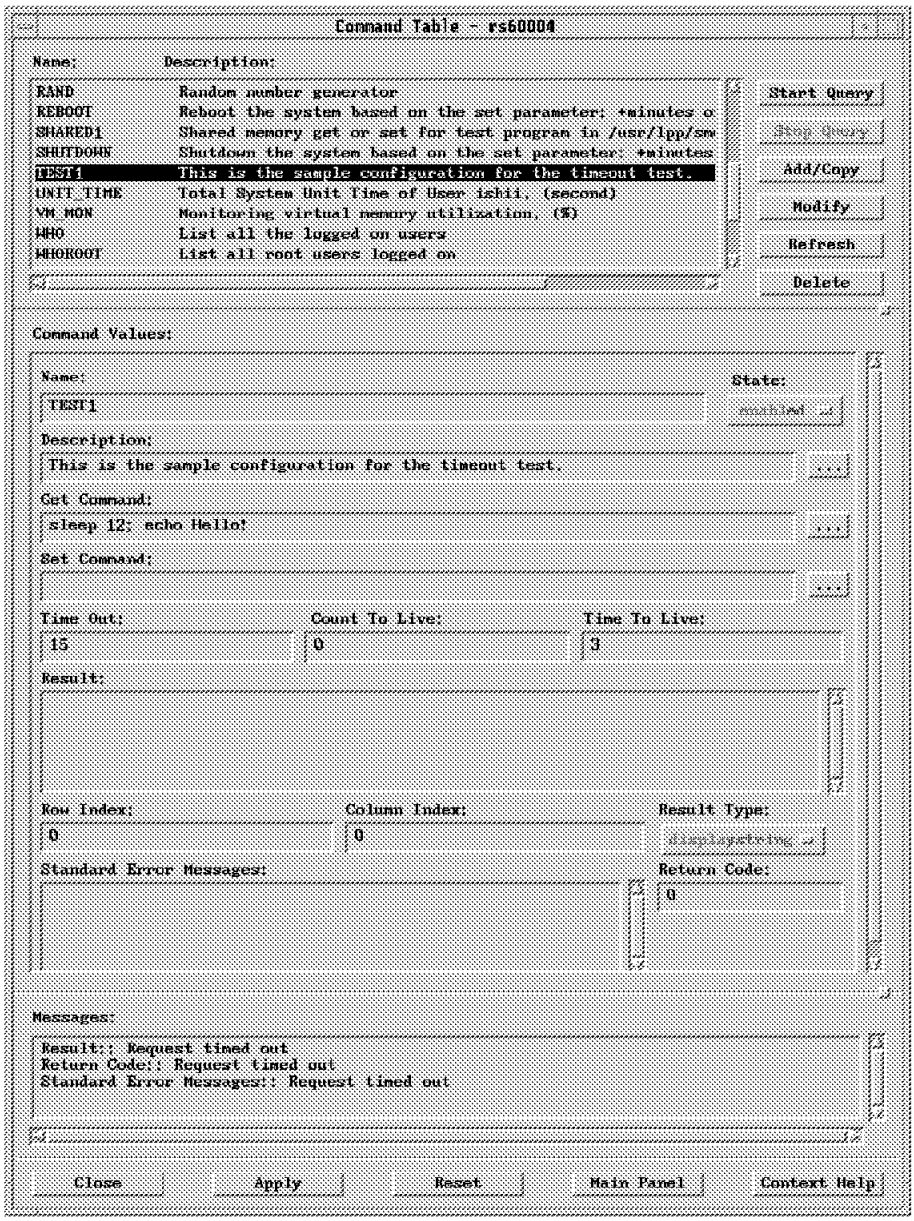


Figure 81. SNMP_GET Timeout Messages

This time the error messages are caused by the SNMP_GET timeout.

3.4.6.5 Test3: Modified SNMP_GET Timeout Value

For our next step we modified the SNMP_GET timeout from 12 seconds to 126 seconds by editing the /usr/OV/conf/ovsnmp.conf file like this:

```
127.0.0.1:ITSC:*::::::
9.24.104.27:ITSC:*:20:5:::ITSC:
9.24.104.*:ITSC:*::::::
*.*.*.*:public::8:3:300:::
```

Figure 82. Modifying the SNMP_GET Timeout

The value of 126 seconds is due to the operation of the retry function. We have set the base timeout for node 9.24.104.27 to 2 seconds, and the retry count to 5.

This means that the total timeout is 2+4+8+16+32+64 seconds (a total 126 seconds).

The result of using this modified value is shown in Figure 83.

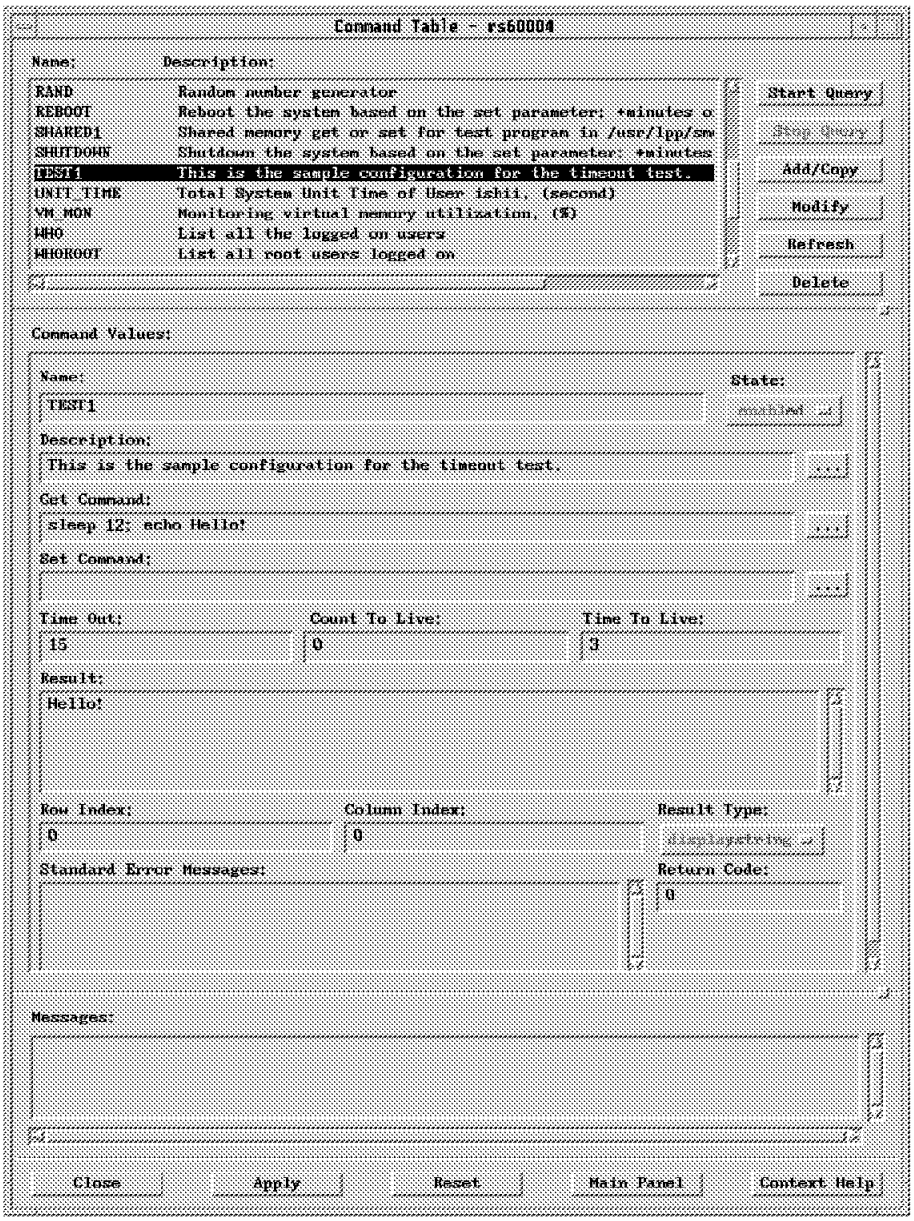


Figure 83. Successful Execution of the Sample Command

We can see that the command has now been successful. However, the changes we have made are not ideal. Because we left the base timeout low (2 seconds) and increased the retries, we will in fact have sent out three SNMP GET requests (at 0, 2 and 6 seconds). It would have been better to increase the base timeout to 15 seconds to match the Command Table timeout. Of course, in that case we should use a lower retry count too, since a retry count of 5 would lead to a total timeout of 945 seconds (over 15 minutes).

Note also, that to implement our complete scenario we may also have to replicate the modified /usr/OV/conf/ovsnmp.conf file on the MLM node, since that

is where the SNMP Get requests would be coming from when we set up the Threshold Table.

3.5 Some Other Ways to Use the Systems Information Agent

In this section, we will discuss some other possibilities for the SIA (sysinfod daemon), looking from the direction of a non-SNMP node.

3.5.1 Managing Non-SNMP Agents with the Help of Systems Monitor for AIX

When we manage the TCP/IP network with the SNMP protocol, the agent node needs at least to have the basic SNMP agent functionality. If the agent has no SNMP agent we can normally get only node up/down status information about it in NetView for AIX.

However, there are some options offered by the Systems Information Agent that allow us to extend a very limited form of management even to these non-SNMP nodes. We will now look at some of these options.

3.5.1.1 Using the snmptrap Command from an IP Node

The Systems Information Agent provides a version of the snmptrap command in directory /usr/lpp/smsia/original. It gives you a command-line interface to send an SNMP trap.

If we have a node which does not support SNMP but from which we can execute remote shell (rsh) or remote execute (rexec), we can use this to send error information to NetView for AIX or a Mid-Level Manager (see Figure 84).

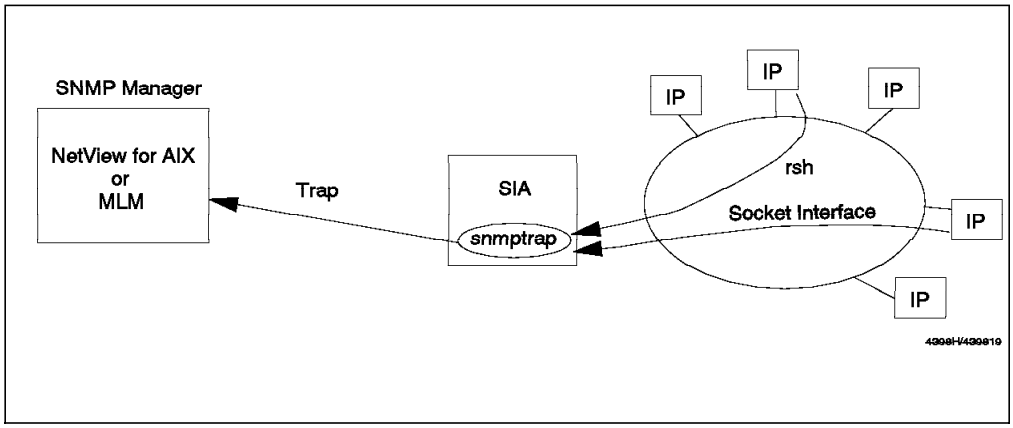


Figure 84. Managing a Non-SNMP Node Using rsh or a Socket Interface

The syntax of the snmptrap command is as follows:

```
# ./snmptrap -?
usage: snmptrap [-?] [-t] [-d] [-p portnum] dest-node community
                enterprise src-addr generic-trap specific-trap
                time-stamp [variable type value]*
valid variable types: Integer, OctetStringHex, OctetStringASCII
                      ObjectIdenitfier, Null, IpAddress, Counter
                      Gauge, TimeTicks, OpaqueHex, OpaqueASCII
```

Figure 85. The snmptrap Command

When an application error or a specific process failure occurs, the non-SNMP node can notify the error by calling snmptrap from a local program or shell script using rsh.

3.5.1.2 Example: Using the cron Daemon

In this case, our scenario is as follows:

1. Make a shell script to monitor the status of a specific process.
2. If the shell script detects the monitored process down, it invokes the snmptrap command on the SIA node using rsh.
3. The trap destination in /etc/snmpd.conf on the SIA node causes the trap to go to the MLM node or NetView for AIX.

We built a simple shell script to monitor a specific process status (Figure 86).

```
#!/bin/ksh

USAGE="usage : MONITOR Proc_Name"

PROC=$1
HOSTNAME=hostname

EXIST=`ps -ef | grep $PROC | awk '{print $8}'`

if [[ $EXIST != *$PROC* ]]
then
   rsh rs60004 /usr/lpp/smsia/original/snmptrap rs60002 ITSC 0 $HOSTNAME \
   6 2010 0 0 OctetStringASCII Proc:$PROC=Down
fi
```

Figure 86. Process Monitoring Shell Script

Next we added an entry into the crontab table so that we can monitor the process status periodically. In this example, we intend to monitor the sqlturbo (Informix RDB) process.

```
0,10,20,30,40,50 * * * * /u/ishii/SHELL/MONITOR sqlturbo 1>/dev/null 2>/dev/null
```

Figure 87. crontab Definition for Monitoring Process Status on a Non-SNMP Node

When this shell script detects the monitored process down, the following messages are logged into the /usr/OV/log/trapd.log file, if we define Netview for AIX as the trap destination node.

```
Tue Sep 06 17:31:27 1994 rs600010.itso.ral.ibm.com ? Trap found with no known format in trapd.conf(4)
Tue Sep 06 17:31:27  rs600010.itso.ral.ibm.com ?  Enterprise SM/6000 (1.3.6.1.4.1.2.6.12) community ITSC
Tue Sep 06 17:31:27  rs600010.itso.ral.ibm.com ?  generic trap:6 specific trap:2010
Tue Sep 06 17:31:27  rs600010.itso.ral.ibm.com ?  Timestamp:949991 Agentaddr:rs600010.itso.ral.ibm.com
args(1):
Tue Sep 06 17:31:27  rs600010.itso.ral.ibm.com ?  [1] .0 (OctetString): Proc:sqlturbo=Down
```

Figure 88. Log Messages from Monitoring a Non-SNMP Node

The same information would appear on a NetView for AIX event card. Although in this example we are using cron to schedule a regular check we could equally

well add logic into application code to invoke snmptrap using rsh when an unexpected situation occurs.

If the non-SNMP node did not have rsh or rexec commands available, we could instead write a simple socket interface program to execute the remote command.

3.6 Comparing SIA and the Host Resources MIB

The Systems Information Agent MIB is a private extension on the IBM segment of the MIB tree. Work has been going on in an Internet Engineering Task Force (IETF) working group to produce a more general purpose host system monitoring MIB (the host resources MIB).

The host resources MIB is defined in RFC1514 located at the following position in the MIB tree:

```
.iso.org.dod.internet.
                mgmt.
                  mib-2.
                    host.
```

Figure 89. Host Resources MIB in the MIB Tree

The host resources MIB provides many kinds of information about an installed node which is much more detailed than the MIB-2 information. We can divide it into the following categories:

- System Information
- Storage Information
- Device Information
- Process Information
- Performance Information
- Software Information

We can see from the above list that some of the classes of object are duplicated in Systems Monitor for AIX. The host resources MIB is less detailed (the number of host resources MIB objects is about 80, compared to over 600 for the SIA). The host resources MIB was designed to be platform independent which has the benefit that the same information can be retrieved from many different kinds of device (for example, OS/2, AIX, Sun or HP etc.). However, because the descriptions are generic, it is sometimes not clear exactly how they apply to a particular operating system.

The host resources MIB is used by the OS/2 agent of NetView for OS/2, and by loading the MIB source into NetView for AIX we can access the information in it through the MIB Browser. Figure 90 on page 107 shows an example of this.

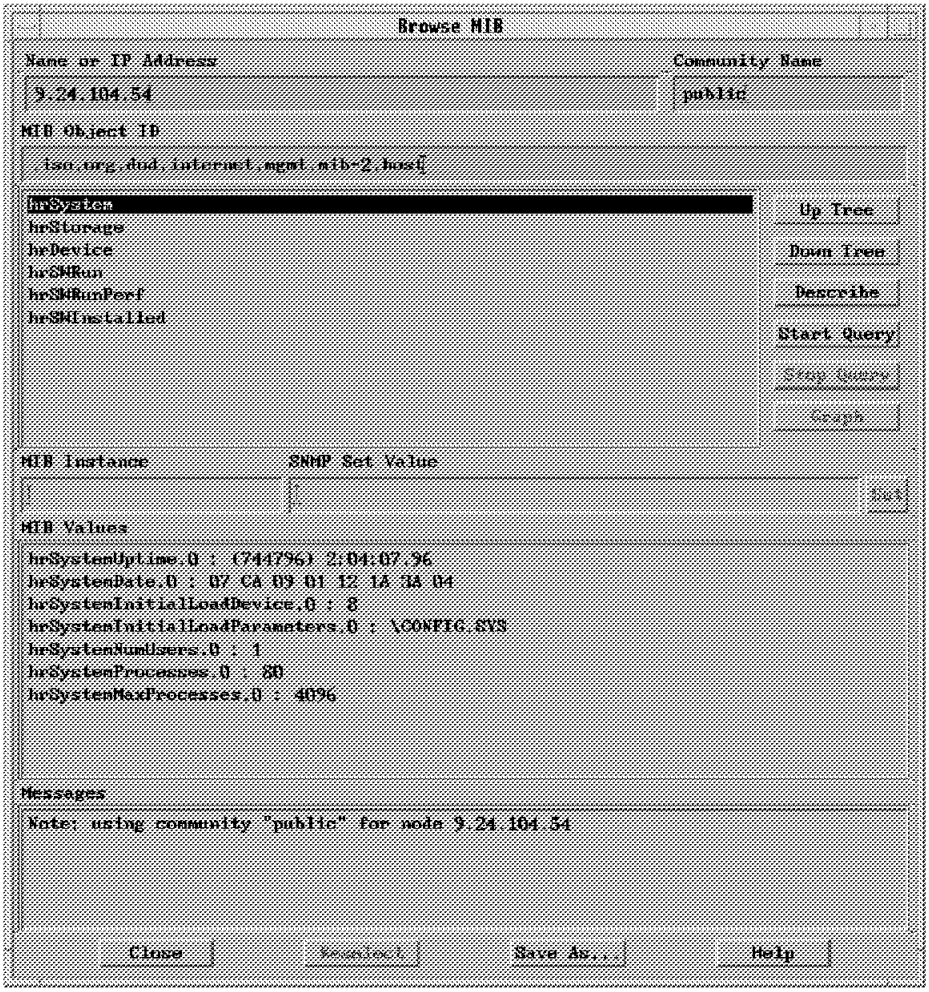


Figure 90. Host Resources MIB in the MIB Browser

The direction for Systems Monitor for AIX is to embrace the host resources MIB as it becomes more widely accepted. The form this will probably take is to set up aliases between the private SIA MIB objects and the host resources MIB objects that they equate with.

Chapter 4. Using the Mid-Level Manager for Status Polling

This chapter describes how the Mid-Level Manager can be used to perform some of the status polling and network discovery tasks that are normally done by NetView for AIX.

With NetView for AIX Version 4 the relationship between AIX Systems Monitor/6000 and NetView for AIX has been widely extended. You will now be able to configure how managers (both NetView for AIX and AIX Systems Monitor/6000) share their workload in a distributed environment. We will have a look at some of the important matters and analyze the various ways AIX Systems Monitor/6000 is involved in distributed network management. We will set up different scenarios and discuss the results.

4.1 Lab Setup

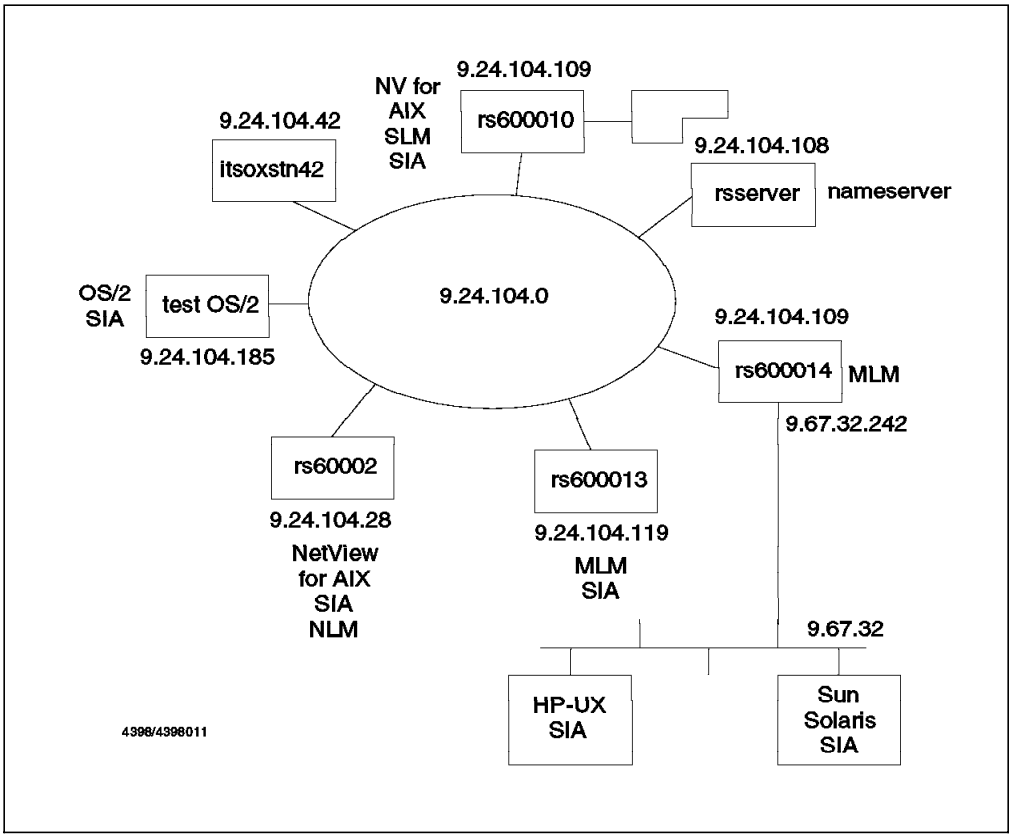


Figure 91. Lab Environment Used for the Examples

Figure 91 shows the setup for the scenarios and examples. NetView for AIX and most nodes reside in a token-ring network, which has an assigned IP address of 9.24.104.0. To set up the different scenarios, we assigned the following participants:

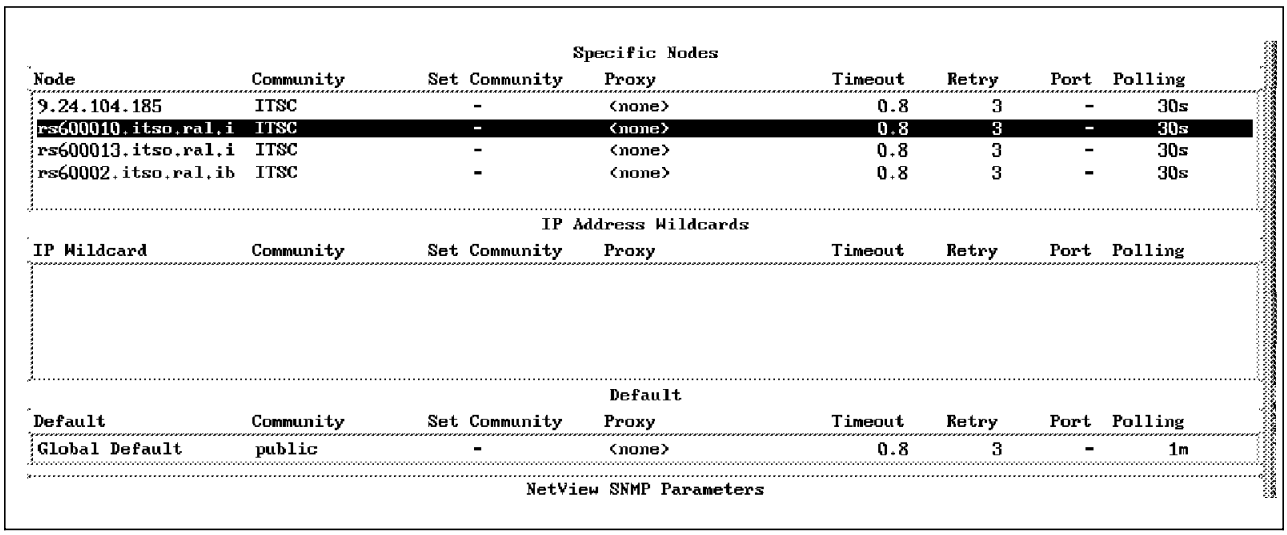
- NetView for AIX executes on node rs600010 (9.24.104.109).
- Node rs600014 acts as a router and interfaces to segment 9.67.32, which is an Ethernet segment. Node rs600014 also has a Mid-Level Manager installed.

- Nodes rs60002 and rs600013 both have Mid-Level Managers running. We will uses these two nodes to show how workload will be distributed among multiple MLMs.
- Node 9.24.104.185 is a PS/2 Model 80 running OS/2 Version 2.0 and the SIA for OS/2.

4.1.1 Lab Setup: Community Names

To ensure that NetView for AIX has write access to all MLMs and SIAs we use in our examples, we configured NetView for AIX to use a read/write community named ITSC. On the NetView for AIX side, community names *must* be set via the Options then the SNMP Configuration menus of NetView for AIX. NetView for AIX Version 4 now uses a database for SNMP configurations and maintains the /usr/OV/conf/ovsnmp.conf file only for compatibility reasons.

Figure 92 shows the community configuration we use. Note that this example contains only complete node addresses. In fact, you will normally use wild cards to assign more global communities in your network.



Specific Nodes

Node	Community	Set Community	Proxy	Timeout	Retry	Port	Polling
9.24.104.185	ITSC	-	<none>	0.8	3	-	30s
rs600010.itso.ral.i	ITSC	-	<none>	0.8	3	-	30s
rs600013.itso.ral.i	ITSC	-	<none>	0.8	3	-	30s
rs60002.itso.ral.ib	ITSC	-	<none>	0.8	3	-	30s

IP Address Wildcards

IP Wildcard	Community	Set Community	Proxy	Timeout	Retry	Port	Polling

Default

Default	Community	Set Community	Proxy	Timeout	Retry	Port	Polling
Global Default	public	-	<none>	0.8	3	-	1m

NetView SNMP Parameters

Figure 92. Community Configuration

The selected community names must exist on the target node(s) in their /etc/snmpd.conf file. To summarize, the file is used in the following way:

1. When a node *sends* an SNMP request, it uses the ovsnmp.conf file (or database, in NetView for AIX Version 4) to select the correct community.
2. When a node *receives* a request, it uses the snmpd.conf file to verify that the request is issued by a valid member of the receiving node's community.

4.1.2 Lab Setup: Changing MLM Trap Definitions

The MLM generates traps to report status changes to the top level manager NetView for AIX. Two of these traps, SM_StatusDown and SM_StatusUp are configured for *log only*. They would never show up in a workspace (control desk, events).

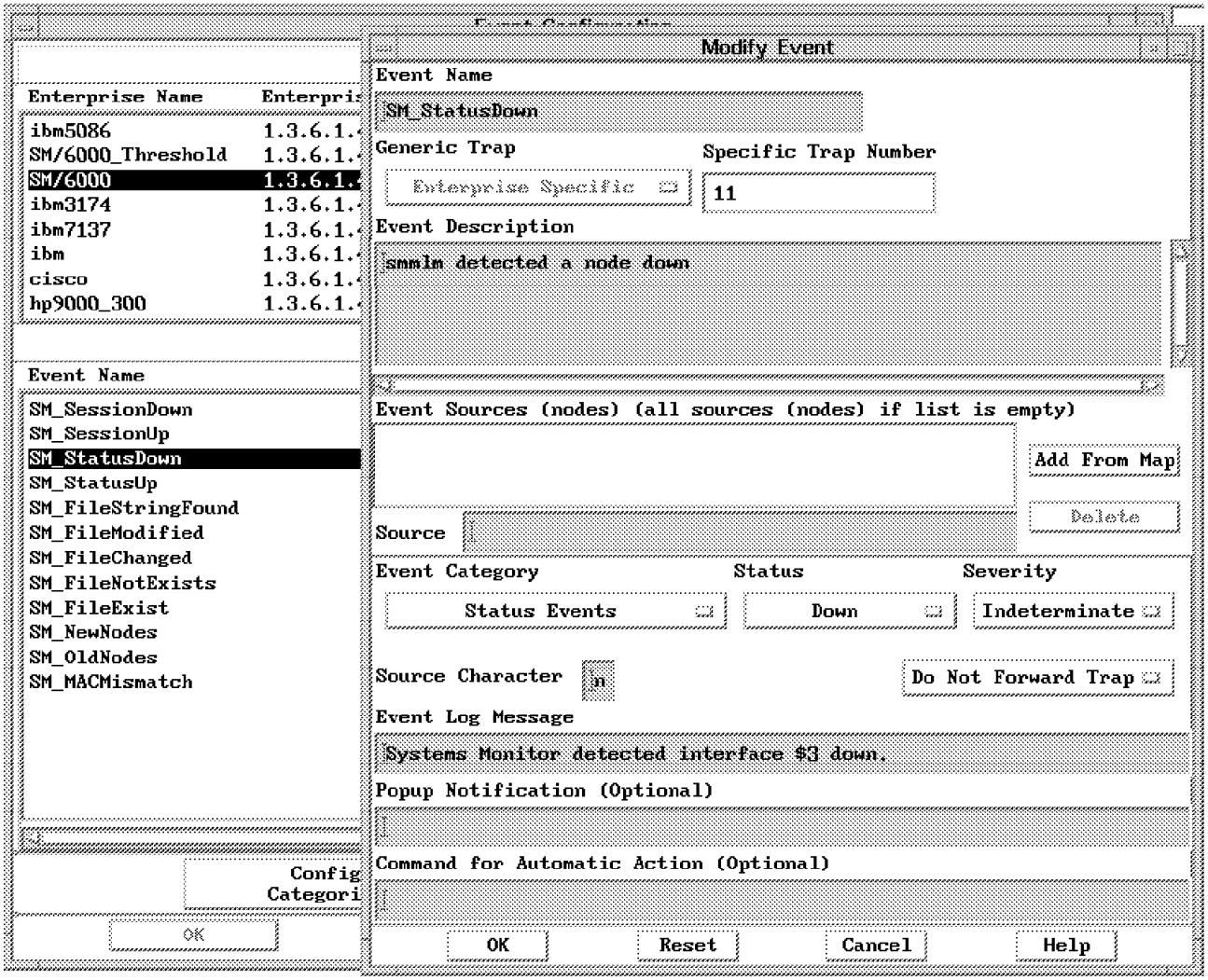


Figure 93. Changing Event Categories for MLM Traps

To demonstrate how the MLM acts, we temporarily need to change the traps to show up in a workspace, thus changing their event category. You can change the trap categories via the NetView for AIX Menu Options, then Event Configuration, then Trap Customization: SNMP. Figure 93 shows the dialogs you use to change the event category.

4.1.3 Lab Setup: Creating Dynamic Workspaces

To further check the results of the different scenarios, we use the dynamic workspace facility of NetView for AIX (Control Desk, Events, Create, Dynamic Workspace...) to filter the traps of interest.

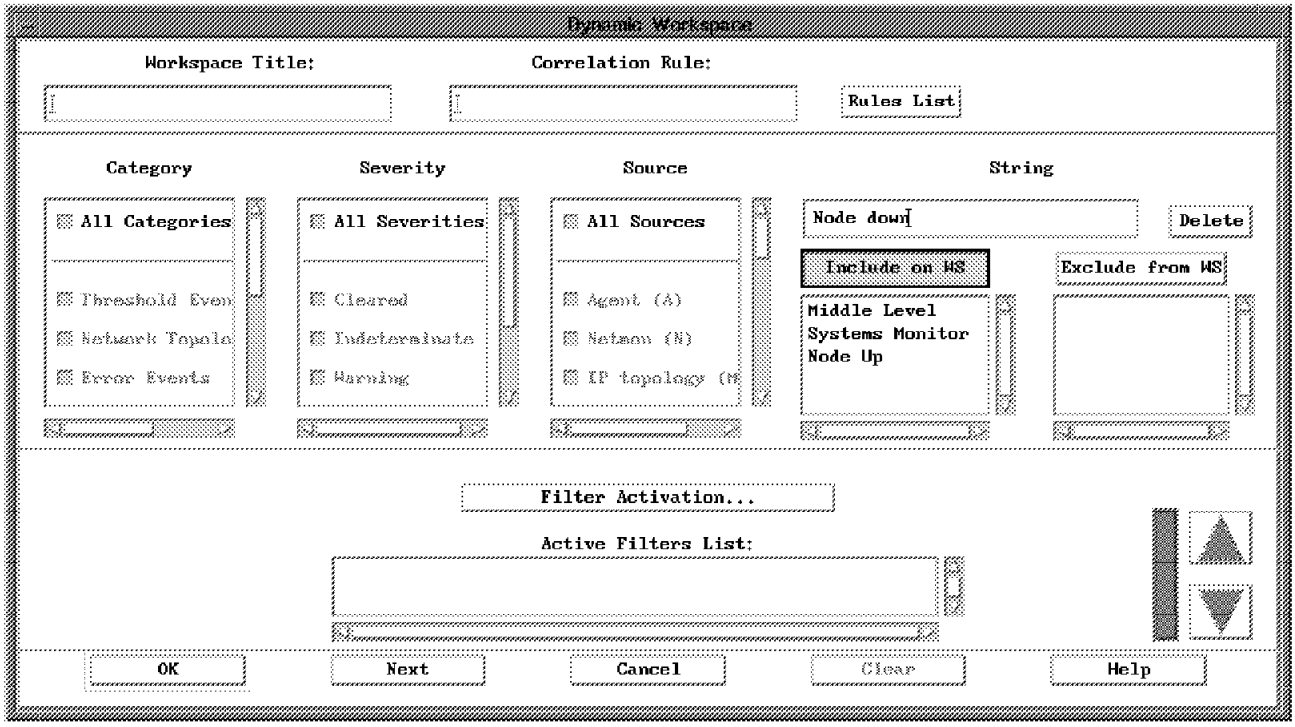


Figure 94. Setting up a Filter for MLM Traps

You can simply use the string option from the dynamic workspace dialog to filter the traps. For the scenarios, traps containing the strings Systems Monitor, Middle Level, Node Up, and Node Down will be informative. Just type the strings and click on **Include on WS** to specify the filter, then click on **OK**.

4.1.3.1 A More Elegant Way to Filter Systems Monitor Traps

In 1.3.3, “APM Console” on page 19, we introduced a special dynamic workspace called APM Console which comes with NetView for AIX Version 4. The term *console* is a combination of an events workspace and the workspace′s active filters. The APM Console filters file monitoring traps delivered to NetView for AIX by Mid-Level Manager or System Level Manager. Using this filter as a template, you can easily define another console and call it, for example, MLM console.

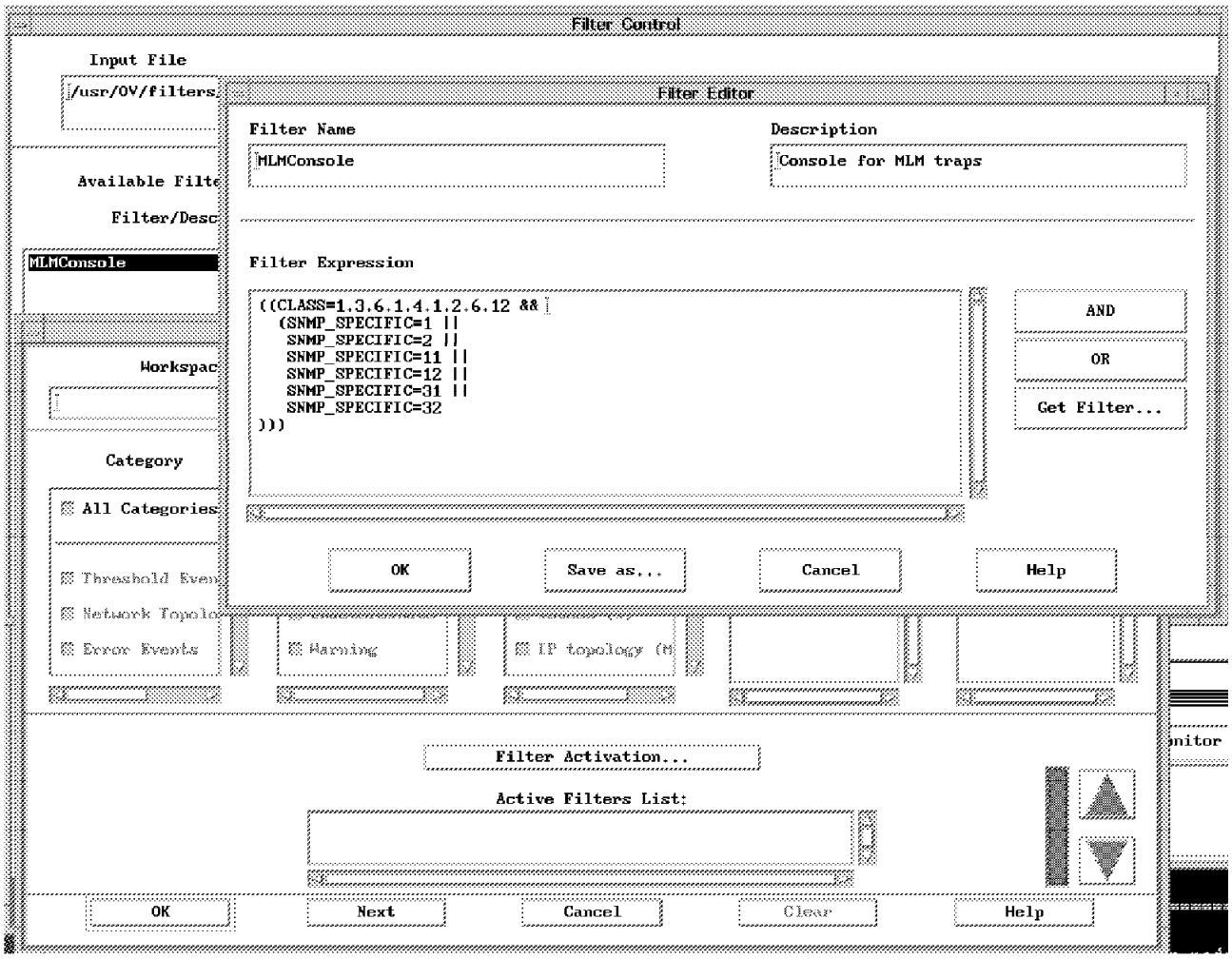


Figure 95. Setting Up a Console for MLM Traps

Figure 95 shows the complete definition. The filter may be created and saved using the NetView for AIX Filter Editor or you can use your favorite editor against the members of /usr/OV/filters to build your own MLM Console filter.

You can activate the Filter Editor via NetView for AIXs Tools, then the Filter Editor menu or from the Control Desk static workspace using the Create, then the Dynamic Workspace dialog. Once in the dialog, use the filter activation button to get access to the Filter Control Dialog where another button allows you to start the filter editor.

The filter prevents all traps which don't match the rule from being sent to the designated workspace. The rule is rather simple:

1. Look for traps having the enterprise ID of SM/6000 (1.3.6.1.4.1.2.6.12).
2. Match one of the specified specific traps:
   - SM_SessionDown (1)
   - SM_SessionUp (2)
   - SM_StatusDown (11)
   - SM_StatusUp (12)
   - SM_NewNodes (31)
   - SM_OldNodes (32)

The traps are those Mid-Level Manager sends to its trap destinations in case of status monitor changes.

To activate the console, select **Create**, and then **Dynamic Workspace** from the workspace menu. Follow the directions given in 1.3.3, “APM Console” on page 19, but select the filter name you choose instead of activating another APM Console.

4.2 Distributed Status and Discovery Polling

NetView for AIX has several concurrent polling cycles:

Configuration Polling This maintains the information that NetView for AIX keeps about a node in its databases. It typically happens infrequently (for example, daily).

Threshold Polling This is set up by the user to allow performance data to be retrieved and stored or tested against a threshold value. Typically it uses a polling period of 10-60 minutes; we will see later how the Mid-Level Manager can off-load this activity.

Status Polling This checks the *reachability* of a node. Unlike the above two polls, which are SNMP GET requests, this is a simple ICMP echo request (ping). Typically it uses a short polling interval - 1 to 5 minutes.

Discovery Polling This retrieves information from the network and analyzes it for clues to the existence of new nodes. It uses a variable polling cycle, starting at a high frequency and then dying away.

Systems Monitor Version 1 allowed you to distribute only the Threshold Polling function. With Systems Monitor for AIX Version 2 you can additionally off-load the Status and Discovery polls to the Mid-Level Manager. Systems Monitor for AIX Version 2 Release 2 further enhanced this process by allowing you to use the collection facility of NetView for AIX Version 4.

Distributed status monitoring is one of the most important features of the Mid-Level Manager, because it enables significant workload to be removed from the NetView for AIX manager, as well as reducing bandwidth utilization. This becomes particularly important when the top level manager is located on the other end of slow wide area links from the managed nodes. Distributed polling assists in reducing the load on these links.

4.3 How netmon Handles Mid-Level Managers

The netmon daemon is a daemon which discovers nodes in a network. It is part of NetView for AIX. After netmon discovers a node, it polls the node regularly to check for status, topology and configuration changes.

The netmon daemon discovers and polls only the nodes in its management region, which initially is the network (or networks) that contains the node in which netmon is running.

The netmon daemon sends ICMP Echo requests to check the status of nodes and uses SNMP to poll nodes for information.

The netmon daemon can distribute its functions to Systems Monitor Mid-Level Managers (MLMs) in either of the following two ways:

- In case there are no MLM domain collections defined or in case the collection facility daemon, nvcold, *is not* running, netmon evenly distributes the nodes in a segment to the MLMs that are in that segment.
- In case there are MLM domain collections defined and the collection facility daemon, nvcold, *is* running, netmon uses the collections to distribute status checking to the MLMs.

The netmon daemon will only use the MLMs to which it has write access. For example, if an MLM has a set community name that is not configured via NetView for AIX′s Options, then the Snmp Configuration... menu option, netmon will not be able to write to that MLM. That means netmon cannot set any of the MLM tables.

4.3.1 Example Using netmon to Distribute Workloads to MLMs

This example discusses a basic feature of the NetView for AIX Version 4 and Mid-Level Manager relationship: automatic detection of nodes executing MLM and distribution of workload from NetView for AIX to the MLM. If no collections are defined *or* the collection daemon nvcold is not running, netmon will distribute all nodes to the MLM for status checking. If netmon discovers more than one MLM in a subnetwork, it distributes the nodes equally to all MLMs. In other words, if the subnetwork contains 60 nodes, netmon will assign 30 nodes to each Mid-Level Manager.

```
rs600010:/usr/OV/lrf > ovstop nvcold
rs600010:/usr/OV/lrf > ovdelobj nvcold.lrf
ovdelobj - Static Registration Utility
Successful Completion
rs600010:/usr/OV/lrf >
```

Figure 96. Unregister and Stop nvcold

To verify this feature, we started two MLMs in the same subnetwork. First, we activated only one MLM on rs60002 and verified how NetView for AIX distributes status checking to that MLM. Then, as soon as the MLM took control, we started another MLM on rs600013. To get the the expected results, make sure nvcold is not running or no collections are defined. If you just installed NetView for AIX, the easiest way to prevent the startup of nvcold would be the removal of nvcold from the startup sequence. Figure 96 shows how to remove the registration of a daemon. Once removed from the NetView for AIX registration database, a particular daemon will not be started until it is registered again.

4.3.1.1 Inserting the First MLM

First, you should activate the remote MLM. The `smit smv2` fast path command or any other of your preferred methods will work.

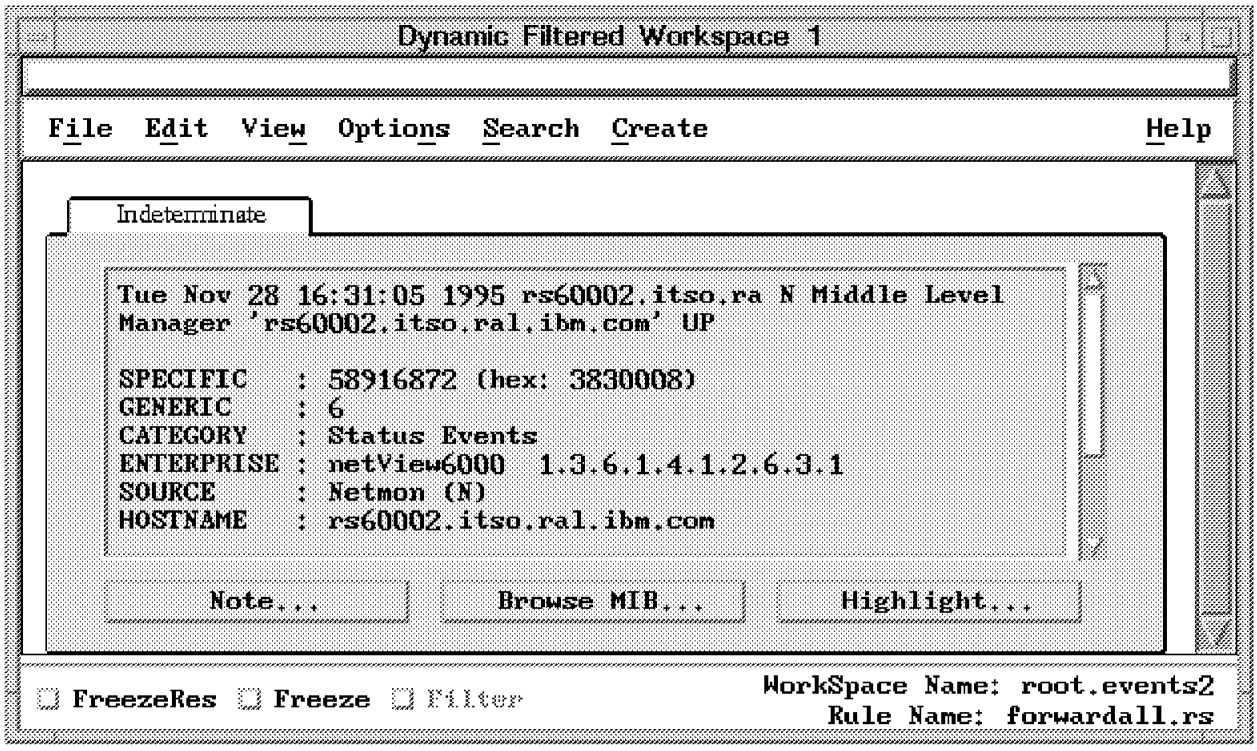


Figure 97. Middle Level Manager Up Trap

In case a dynamic workspace as described in 4.1.3, “Lab Setup: Creating Dynamic Workspaces” on page 111 has been set up, a card similar to Figure 97 will appear. It signals that NetView for AIX detected an MLM in the network. You can force the appearance of the card by stopping and starting netmon using ovstart (Figure 98 on page 116).

```
rs600010:/u/peterg > ovstop netmon
rs600010:/u/peterg > ovstart netmon
rs600010:/u/peterg >
```

Figure 98. Forcing MLM Detection

When this first MLM is discovered NetView for AIX will attempt to populate several of the MLM tables. Remember that these tables are updated using SNMP SET, so this will not work unless NetView for AIX has been correctly configured to use an SNMP community name with read/write access to the MLM MIB.

The following Mid-Level Manager tables will be updated:

Trap Destination Table NetView for AIX will add an entry in this table to ensure that all traps are sent to itself. By default, this entry will use TCP for sending traps, instead of UDP. An example of a trap destination table entry is shown in Figure 114 on page 131.

Alias Table NetView for AIX will also add entries to the Alias Table; we discuss the alias table later in 6.1.1, “Alias Table” on page 149. In brief, it allows the MLM and NetView for AIX to use a single name to refer to a list of nodes. These entries will list all the nodes that are on the same subnet as the Mid-Level Manager, and will be polled by the

Mid-Level Manager and not NetView for AIX. Each list that the alias resolves to is restricted to a maximum of 255 characters. Therefore, depending on the number of nodes in the subnet, one or more aliases will be created. The form of the alias entry name is always NVa.b.c.d.n, where a.b.c.d is the IP address of the NetView for AIX node and n is an index number.

Status Monitor Table This table contains instructions on how the MLM is to poll for status. NetView for AIX adds one or more entries in this table, to specify how often the Mid-Level Manager node should poll the other nodes in its subnet. The values that are entered for polling frequency, timeout, and maximum retry attempts are the same as are defined in NetView for AIX. That is, the defaults are:

- Polling frequency = 5 minutes or 300 seconds
- Time Out = 0.8 second

  This value is rounded up to 1 second in the status policy table.
- Maximum retry attempts = 3

By starting the Systems Monitor Configuration Application, you should be able to see how netmon configured the Mid-Level Manager.

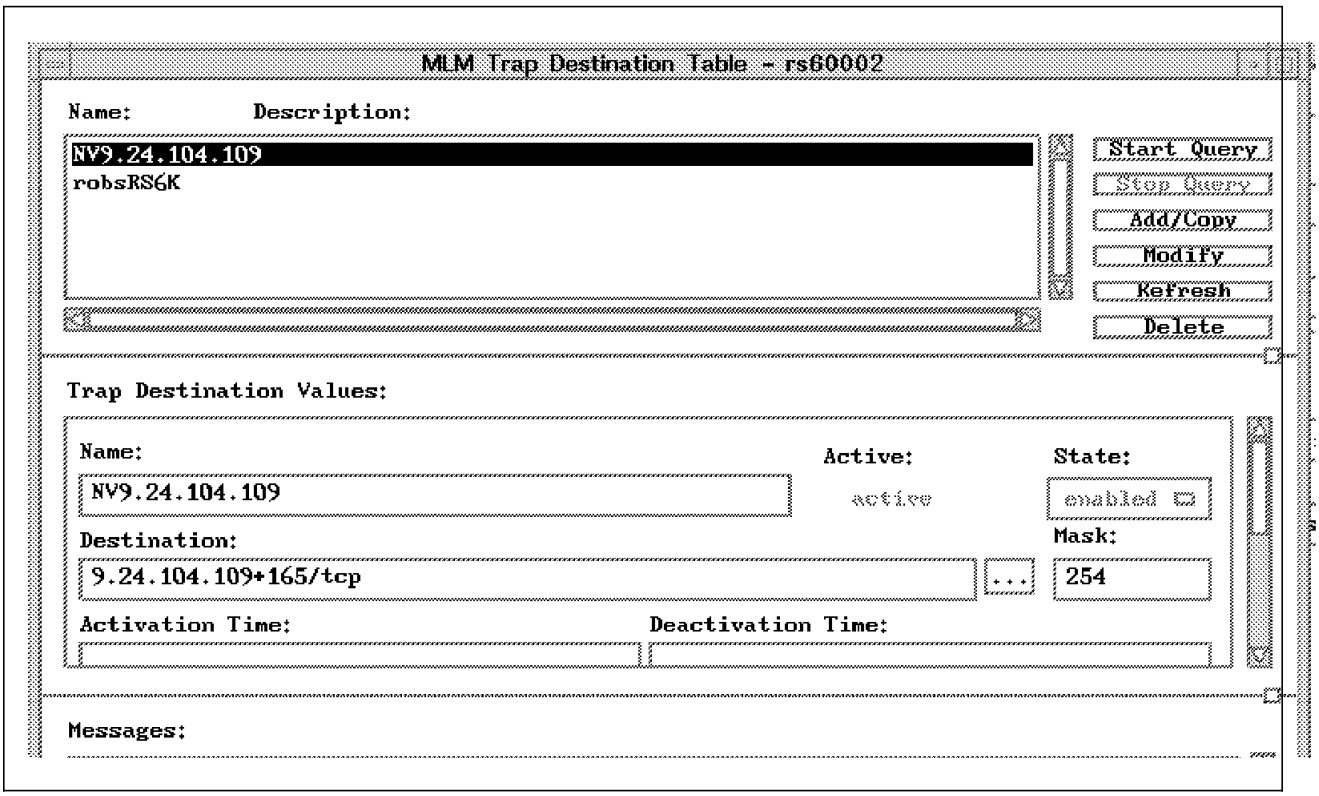


Figure 99. Trap Destination Set by NetView for AIX

The first thing you should check is the Trap Destination Table. In case you changed the TCP port to be port 165 as described in 1.3.1, “TCP Port 165 to Prevent Port Conflicts” on page 16, you should have an entry in your MLM Trap Destination Table inserted by NetView for AIX. Otherwise, this entry should reflect port 162 to be the trap destination 9.24.104.109+162/TCP. The displayed IP address should be, of course, the address of the node executing NetView for AIX.

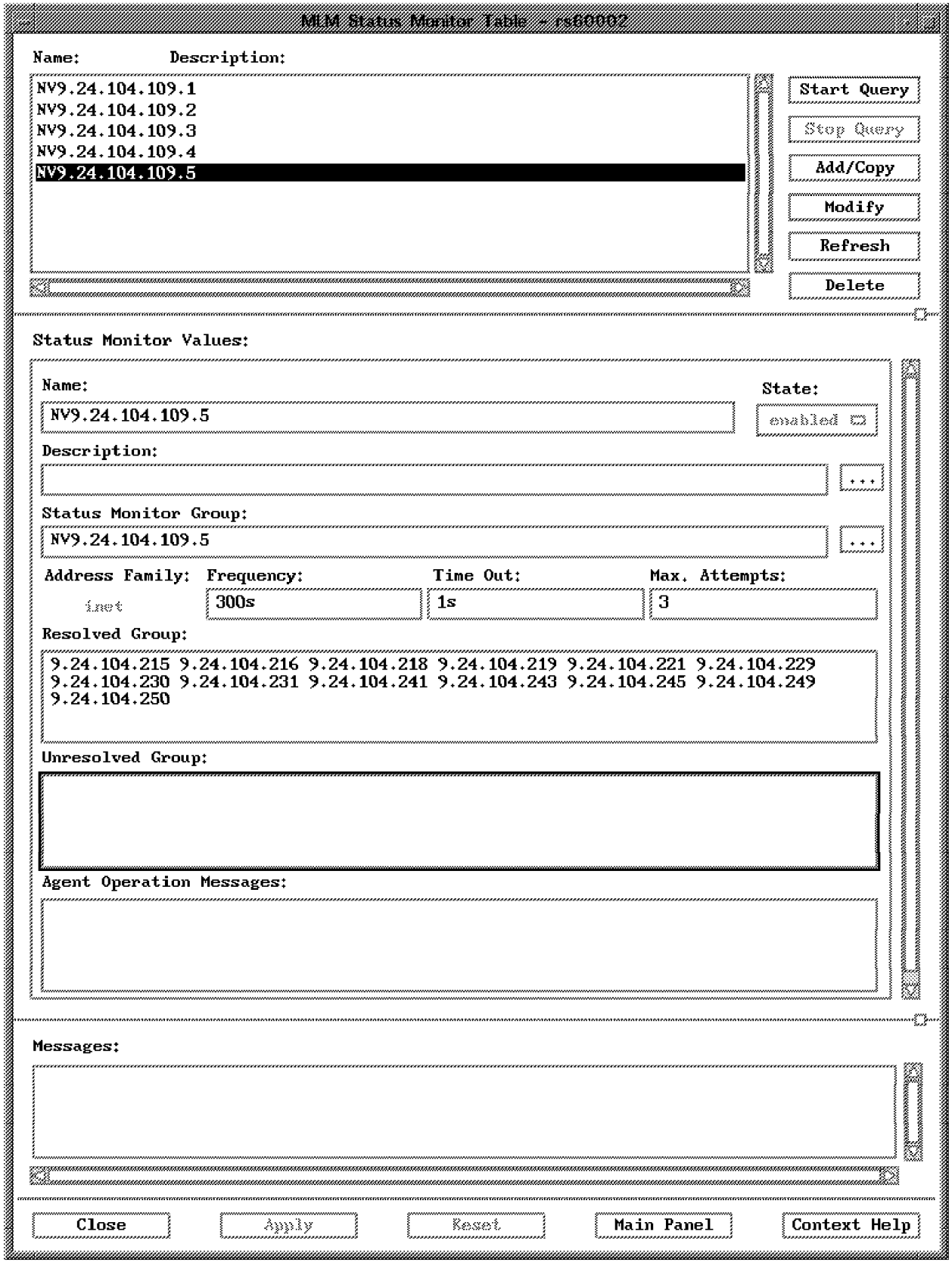


Figure 100. Nodes Distributed to MLM for Status Checking

The next MLM table you may have a look at will be the MLM Status Monitor Table. Figure 100, shows a couple of entries containing lists of nodes the MLM is supposed to do status checking on. Try to give your actual list a closer look. You will find only plain nodes; or in other words, nodes who own just a single interface. NetView for AIX will continue to poll all the nodes in the subnet which have more than one interface (more than one IP address) and therefore may act as an IP router. It seems, NetView for AIX did its job and distributed status checking to the Mid-Level Manager. To verify the correct operation, we artificially produced a node down situation.

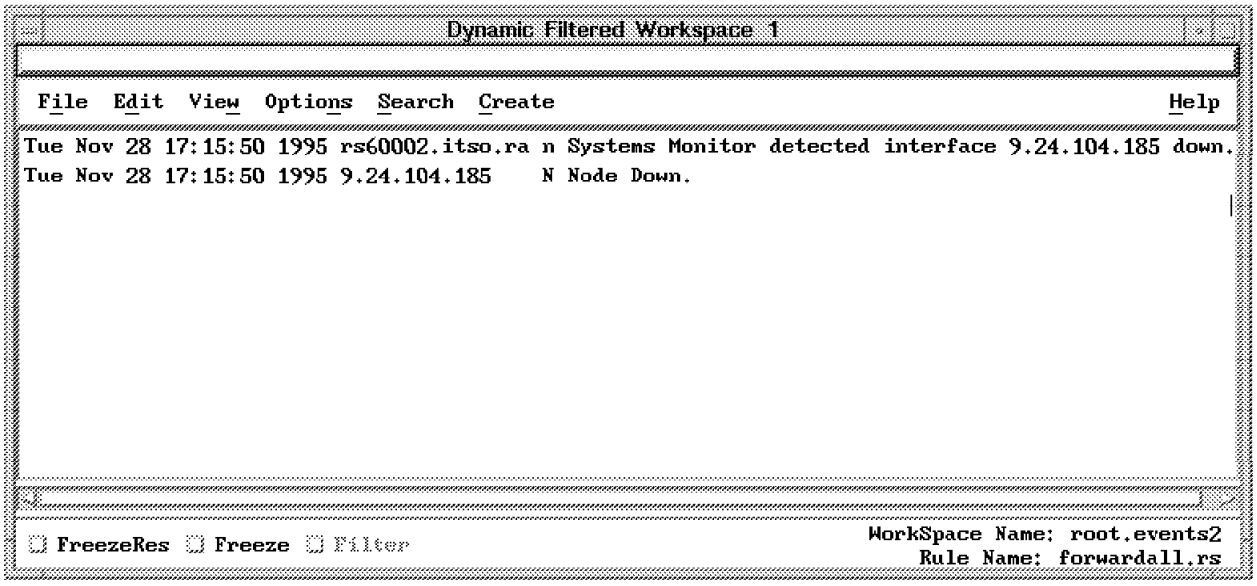


Figure 101. Result of MLM Detecting a Node Down Condition

We used the OS/2 System 9.24.104.185 and issued an `ifconfig lan0 down` command to mark the default (and only) LAN interface of the OS/2 node as down. This resulted in the following two traps to show up in our workspace, which was created as described in 4.1.3, “Lab Setup: Creating Dynamic Workspaces” on page 111:

- An SM_StatusDown (generic 6, specific 11) trap produced by the MLM and sent to NetView for AIX
- The standard node down trap generated by netmon

For a complete list of traps generated by the Systems Monitor agent, refer to Appendix B, “Systems Monitor Traps” on page 307. Because the node down trap is always generated by NetView for AIX, the SM_StatusDown trap has a default category of log only. That prevents the trap from being sent to the workspaces. In addition, the node down trap causes a correct color change of symbols through the maps.

You can now reconfigure the SM_StatusDown trap to its normal configuration, unless you want the two traps to show up each time.

4.3.1.2 Inserting a Second MLM

The previous part of our example hopefully gave you an idea of how NetView for AIX and Mid-Level Manager work together. But what happens when more than one MLM is detected in the same subnetwork? Under normal circumstances, you would set up just one Mid-Level Manager to manage its subnetwork. This offloads some work from the central manager and limits the network traffic NetView for AIX produces to poll all its discovered nodes giving more bandwidth to real network traffic.

Management tasks can be quite complex and network segments may contain a high number of nodes. This can produce a significant workload to the Mid-Level Manager. NetView for AIX Version 4 has been extended to take care of such a situation; if NetView for AIX detects more than one Mid-Level Manager in the same subnetwork, it offloads responsibilities to *both* Mid-Level Managers.

Using the same lab setup as before, we simply activated another Mid-Level Manager running on rs600013.

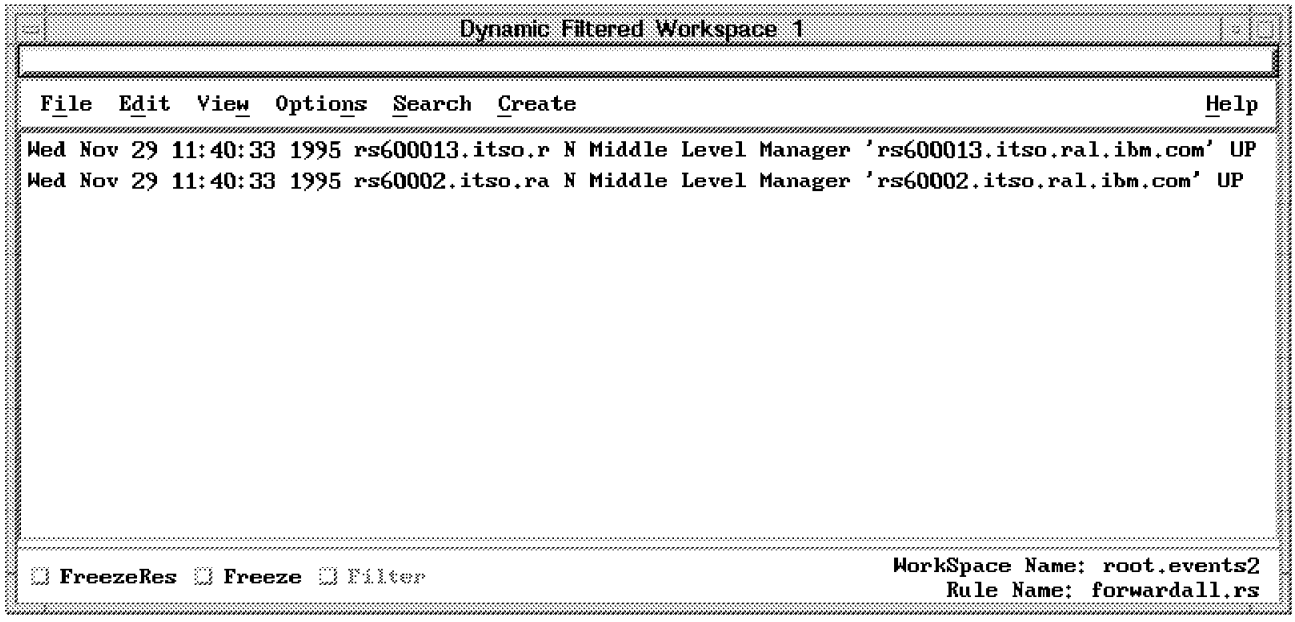


Figure 102. Two Mid-Level Managers Detected

As Figure 102 shows, NetView for AIX now detected two Mid-Level Managers. As with the first Mid-Level Manager, NetView for AIX will make a trap destination entry to the Mid-Level Managers Trap Destination Table to make sure traps generated by this MLM will be delivered to NetView for AIX.

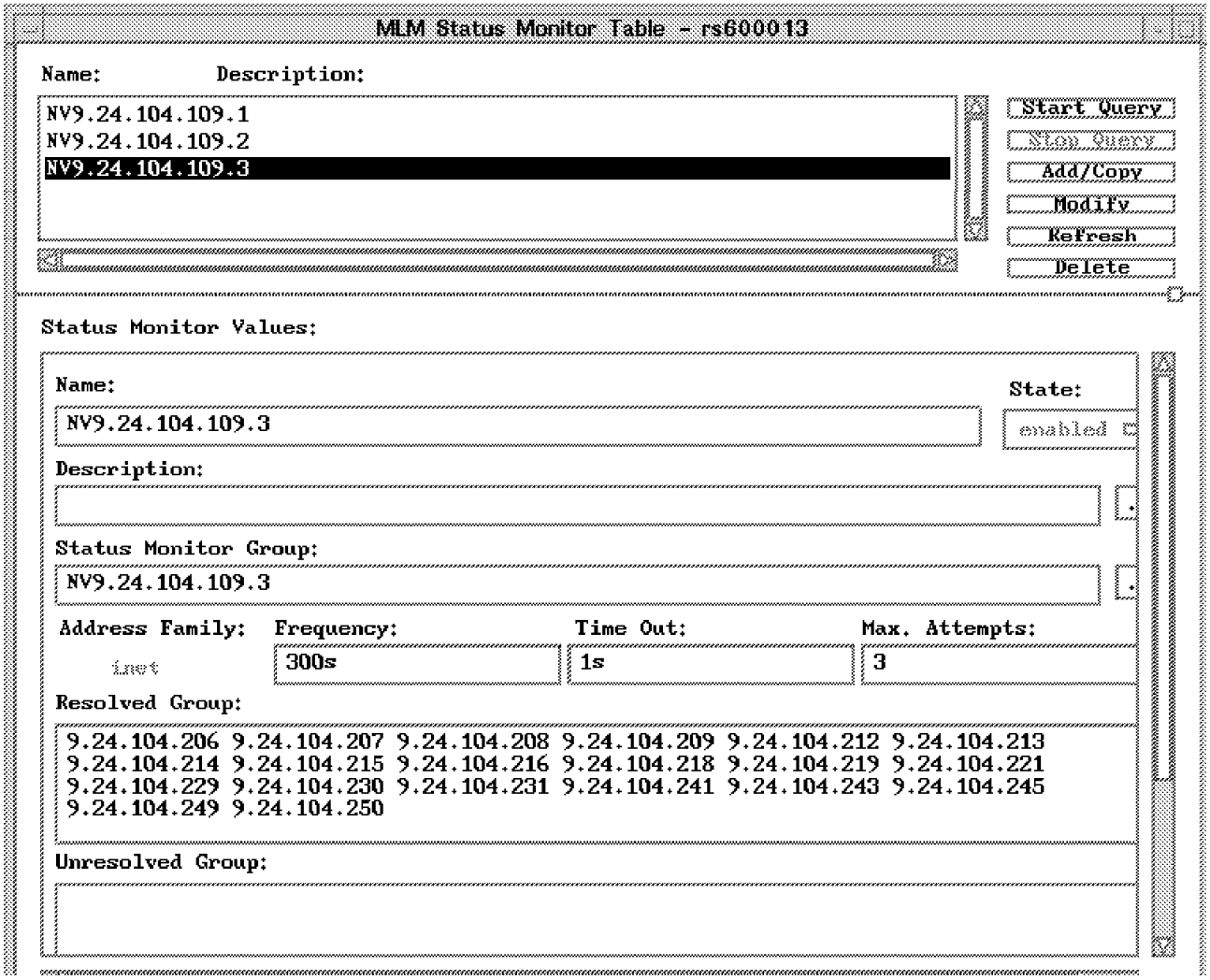


Figure 103. Distributed Nodes on the Second Mid-Level Manager

If you log in to the new Mid-Level Manager using the Configuration Application and have a look at the Status Monitor Table, you will see something similar to Figure 103. A number of nodes are already given to the second MLM for status checking. Because the new inserted manager does some of the work now, the MLM inserted, first lost a number of nodes it had initially assigned. If you compare Figure 104 on page 122 to Figure 100 on page 118, you will notice the changes.

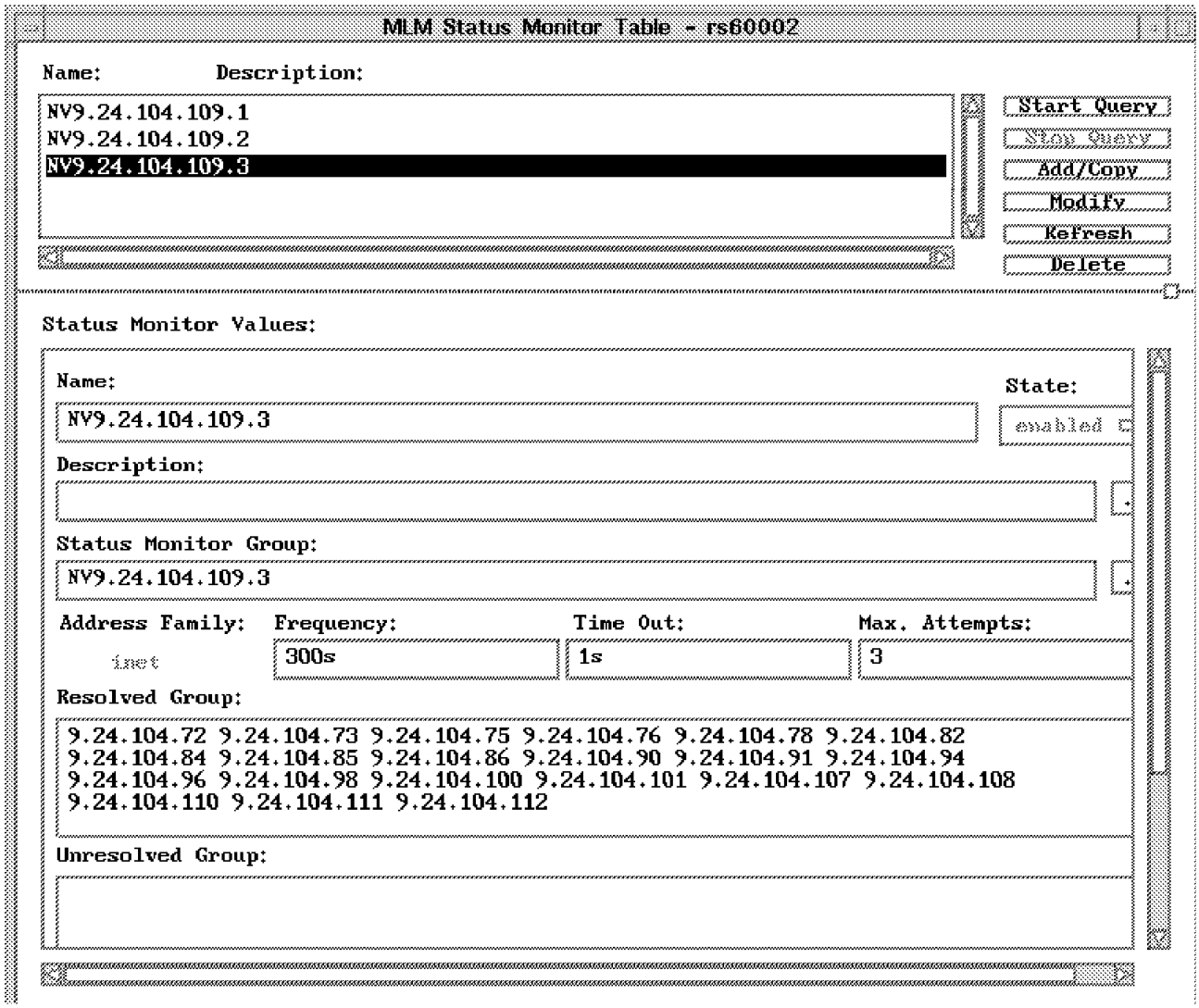


Figure 104. The Updated Status Monitor Table of rs60002

4.3.1.3 Removing a Mid-Level Manager from the Subnet

What happens if one of the designed MLMs disappears from the network? If netmon follows its own rules, it should redistribute the nodes given to the failing Mid-Level Manager to the remaining MLMs in the subnet.

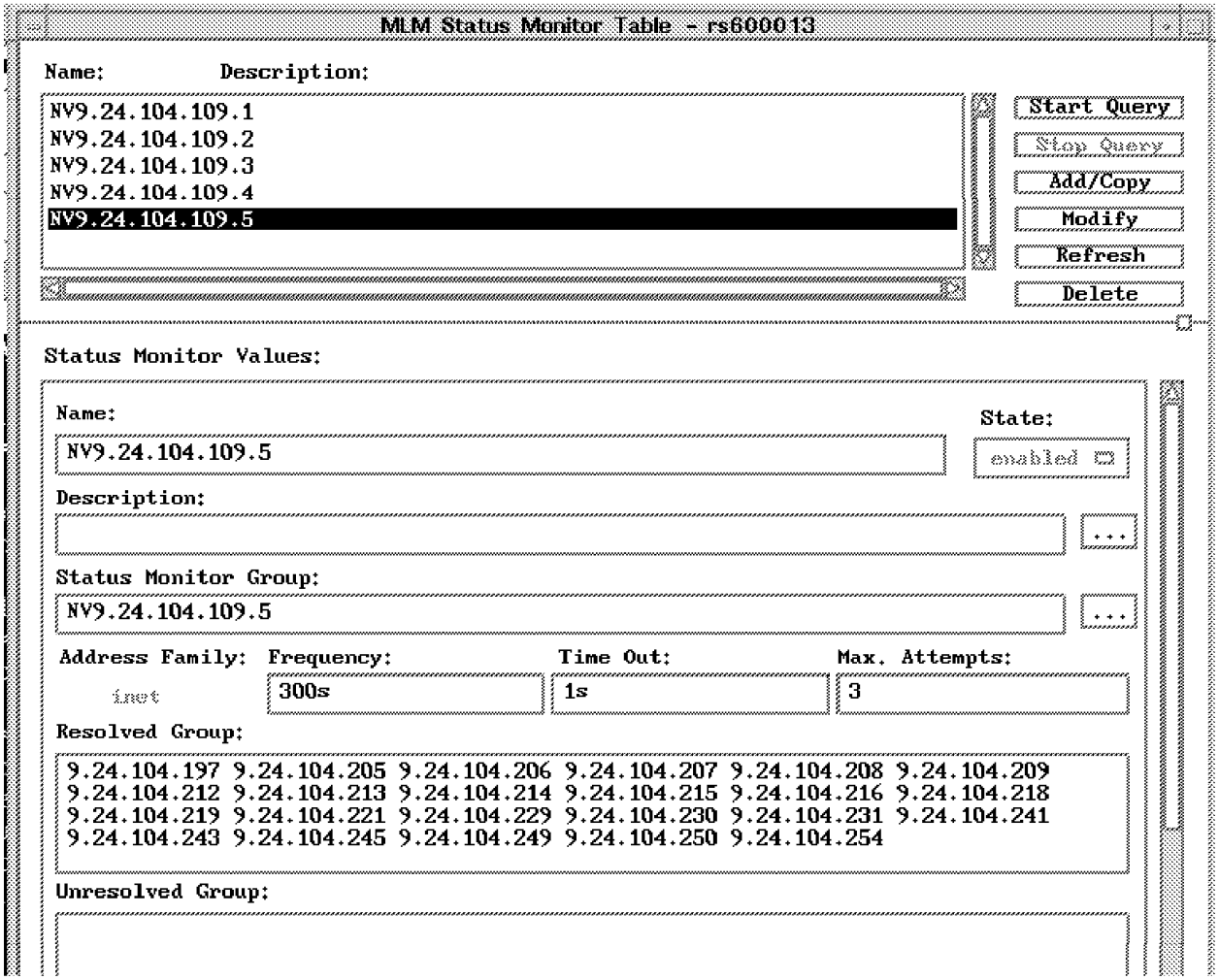


Figure 105. Status Checking Given to rs600013 After rs60002 Failed

To see if netmon redistributes status checking in case of a failing MLM, we stopped MLM on rs60002. After a while, we checked the Status Monitor Table of rs600013. Figure 105 verifies that netmon distributed the responsibilities of rs60002 to the remaining Mid-Level Manager running on rs600013.

Note

What happened, regarding the examples we discussed in this chapter, when we stopped the Mid-Level Manager on one of our two nodes in the subnetwork and instead, brought up a NetView for AIX manager? Well, as far as the distribution of status checking to other managers is concerned, nothing happens. NetView for AIX distributes status checking to Mid-Level Managers by populating their Alias and Status Monitor Tables. NetView for AIX, although also a manager, does not work table-oriented as does AIX Systems Monitor/6000. The netmon daemon simply does not know how to delegate duties to another NetView for AIX.

Currently the only way to transfer management tasks between NetView for AIX managers is the manager backup facility of NetView for AIX. You can find an in depth discussion of manager backup in *Examples of Selected Configuration and Customization Matters with NetView for AIX and Its Family*, GG24-2521.

4.4 In Case of Problems

Now, you have set up your environment, activated the MLM(s) and are waiting for netmon to perform its distribution, but nothing happens.

To solve this problem, always check your community assignments first. The netmon daemon can only distribute to Mid-Level Managers if it has write access. If you ever see authentication traps from the nodes you manage, you should recheck your community setup.

Next, check if the trap reception ports on both side of the connections match.

```
rs600010:/u/peterg > netstat -a |grep nvtr
tcp    0    0  rs600010.itso.ra.nvtra rs60002.itso.ral.1276  ESTABLISHED
tcp    0    0  rs600010.itso.ra.nvtra rs600013.itso.ra.1623  ESTABLISHED
tcp    0    0  *.nvtrapd-             *.*                    LISTEN
rs600010:/u/peterg >
```

Figure 106. Mid-Level Managers Connected to the Trap Reception Port

In a working environment, you should always see your Mid-Level Managers connected to the appropriate TCP port on your NetView for AIX node. At least you should see a LISTEN on the nvtrapd port, which was issued upon startup of correct customized NetView for AIX. You may examine the status of your connections with the `netstat -a` command and filter the output with grep. The result should be similar to Figure 106, which shows the correct connection between NetView for AIX and the two Mid-Level Managers we used in our examples.

If the results are not as expected, you can examine the log file on the Mid-Level Manager side to see if the MLM had problems delivering the traps.

```
rs60002:/var/adm/smv2/log > tail midmand.log
817759542: 11/30/95 14:25:42 Received trap 1(0) from 9.24.104.28
817759543: 11/30/95 14:25:43 No filter rule was matched:
                             default action = FORWARDED
817759543: 11/30/95 14:25:43 ERROR: Tcp trap destination connection
                             to 9.24.104.109 at port no. 162 was closed
rs60002:/var/adm/smv2/log >
```

Figure 107. midmand.log Entries Resulting from a TCP Port Mismatch

Figure 107 shows the log entries which typically result from an incorrect port set up. We caused an snmpd warm start by releasing a `refresh -s snmpd` command on rs60002. This results in a trap being received by the Mid-Level Manager who was unable to deliver the trap via TCP to its top level manager.

```
rs60002:/var/adm/smv2/log > tail midmand.log
817760441: 11/30/95 14:40:41 Received trap 1(0) from 9.24.104.28
817760441: 11/30/95 14:40:41 No filter rule was matched:
                             default action = FORWARDED
rs60002:/var/adm/smv2/log >
```

Figure 108. midmand.log Entries Resulting from Correct Delivery

If the ports on both sides match, a typical trap processing log on the Mid-Level Manager side should always look like the entries shown in Figure 108.

4.4.1 Changing the Polling Frequency of a Node

The status poll is the only means that NetView for AIX has to detect whether a node is up or not. Therefore, the accuracy of the network map and the speed with which status changes are reflected is dependent on the polling interval. For the majority of nodes the default interval of 5 minutes may be satisfactory, but for important nodes we will want to poll more frequently. You can alter the polling interval on a per-node or per-network basis, using Options, then SNMP Configuration from the NetView for AIX menu bar.

How does this work when status polling has been devolved to a Mid-Level Manager? You make the configuration change in exactly the same way as before, using the NetView for AIX SNMP Configuration option. This causes updates to the MLM alias and status monitor tables.

It is also possible to change the polling intervals of individual nodes by adding entries to the status policy table manually. This is *not* recommended. It is better to use the single user interface of NetView for AIX to control polling, so that you have an accurate view of what is going on at each point.

4.4.2 What Happens when Mid-Level Manager Discovers a Node is Down?

When MLM status polling discovers that a node or interface has gone down it will send a status trap to NetView for AIX, indicating that the status of the interface has changed. This is a generic trap from the SM/6000 enterprise ID (.1.3.6.1.4.1.2.6.12 in dotted decimal), with a specific trap number of 11. The trap is normally defined to be log only, but it causes NetView for AIX to generate an Interface Down and Node Down event. The result of this is that the user sees no difference between an interface down event detected by the MLM and one detected by NetView for AIX.

Figure 109 on page 126 shows the event card display for a node failure detected by Mid-Level Manager, and Figure 110 on page 126 shows the corresponding entries in the trap log.

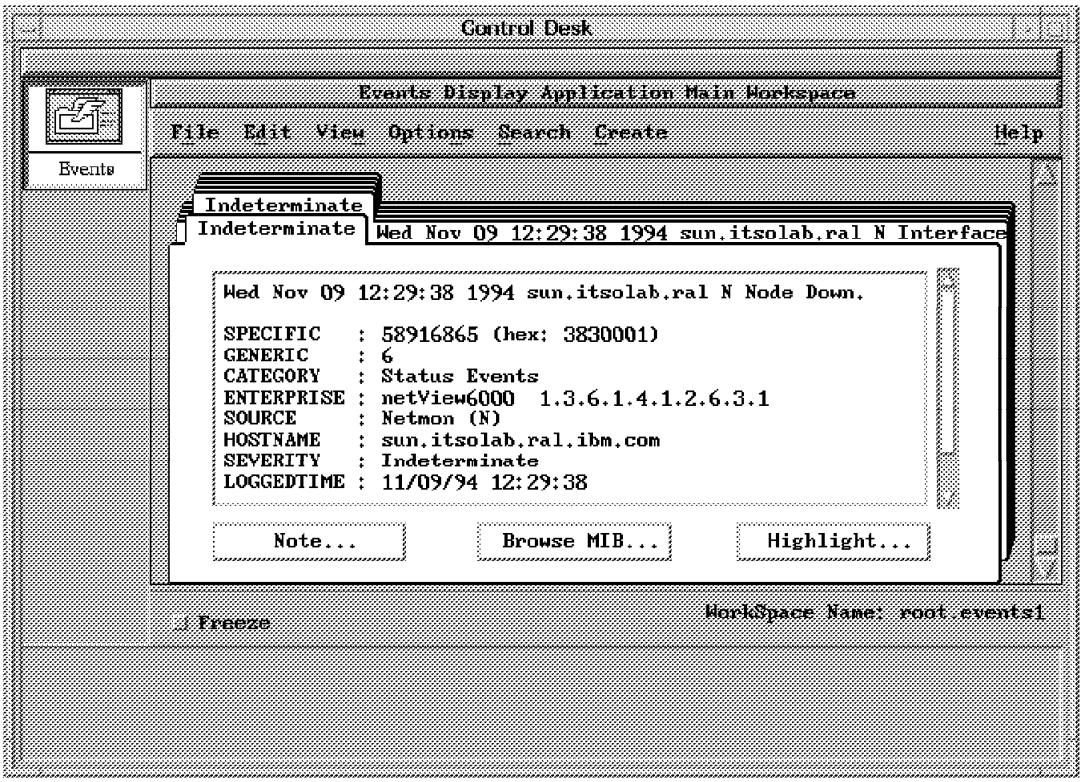


Figure 109. MLM-Detected Node Down

```
784402178  7  Wed Nov 09 12:29:38 1994 midman                    n Systems Moni
tor detected interface 9.67.32.11 down.
784402178  3  Wed Nov 09 12:29:38 1994 sun.itsolab.ral.ibm.com   N Interface 9.67.32.11 down.
784402178  3  Wed Nov 09 12:29:38 1994 sun.itsolab.ral.ibm.com   N Node Down.
```

Figure 110. Interface Down Events in trapd.log

The first trap does not appear on the event card display because it is log-only.

When Mid-Level Manager discovers that a node or interface has come up, it will send an Interface Up status trap to NetView for AIX, which is identical to the Interface Down trap with a specific trap number of 12 instead of 11. This will result in NetView for AIX generating an Interface Up and Node Up event.

4.4.3 What Happens if the Mid-Level Manager Dies?

If the Mid-Level Manager daemon dies for any reason, then NetView for AIX will take control of the status polling. We will illustrate this with a brief scenario.

The MLM on node midman is polling all nodes on subnet 9.67.32, at 30-second intervals and NetView for AIX on rs60002 is polling midman at the same interval. We stop the MLM daemon using SMIT, and then restart it about a minute later. This is reflected by a pair of events (Middle Level Manager Down/Up) in the NetView for AIX event cards (see Figure 111 on page 127).

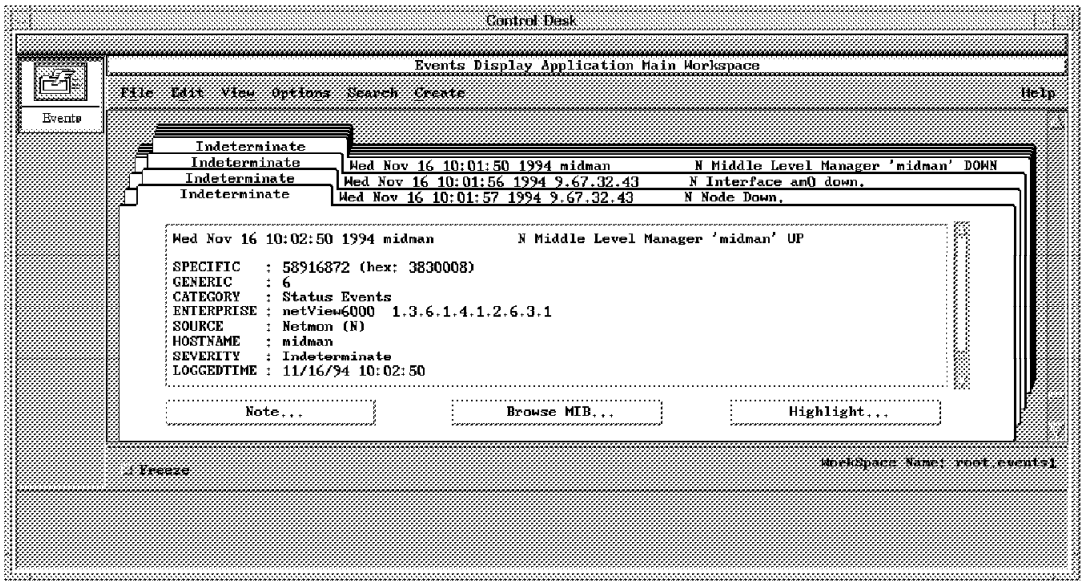


Figure 111. Events Generated by Mid-Level Manager Stop/Start

During the period between these two events, rs600010 took responsibility for status polling in the 9.67.32 subnetwork. We can see this by looking at a trace of network activity on the NetView for AIX node. Appendix F, “IP Trace for MLM Failure Scenario” on page 319 shows an annotated version of such a trace.

In this case we did not completely shut down the midman machine but only stopped the agent, midmand. NetView for AIX detected this because the polling it does of an MLM node is *not* the simple ICMP echo that is used for normal status polling. Instead, it does an SNMP GET request for a field in the Mid-Level Manager MIB.

In general, the operator will not see any change whether the Mid-Level Manager is doing status polling or NetView for AIX is itself doing it. However, one thing you may notice from the event cards shown above is that NetView for AIX showed a node in the 9.67.32 network going down during the time that it was responsible for status polling. Was it just chance that the node went down at that time? The answer is no, because the node in question (address 9.67.32.43) is an Xstation. For normal activity, it does not need to have IP routes beyond the local network where its boot host lives. Thus, midman can see this node, because it is on the same network, but rs60002 cannot. Hence, when the MLM fails, the Xstation appears to have failed too, but only because it is unreachable from the NetView for AIX host.

4.4.4 Discovering a New Node

In addition to doing status polling on behalf of NetView for AIX, the Mid-Level Manager can also perform new node discovery. It gives you three different options for configuring the following:

Passive Sensing This has the least network and system overhead. It means that the source node of any packet received by the MLM will be polled, if it is not already in the polling list.

Active Sensing This consumes more system resource than passive sensing, but is also more effective. Unlike passive sensing, this will look for addresses contained within the body of received packets, and then investigate them.

Broadcast Search This sends an ICMP echo (ping) to the broadcast address, to try to elicit some traffic from devices which are not actively using the network. You can specify the interval at which to send the broadcast.

For a normal LAN environment, the overhead of using active sensing, with an occasional (say: every 30 minutes) broadcast search is very small, and this is the recommended configuration. You can control how much work the active- and passive-sensing routines do by setting values for the sampling interval and the maximum number of packets they are to analyze during that interval.

We configured node discovery on midman by selecting the **Node Discovery** option from the Systems Monitor configuration panel. The resulting panel and the changes that we made to activate discovery are shown in Figure 112 on page 129 and Figure 113 on page 130. As always with Systems Monitor for AIX, the configuration is done using SNMP SETs, so you need to use a community name with read/write access.

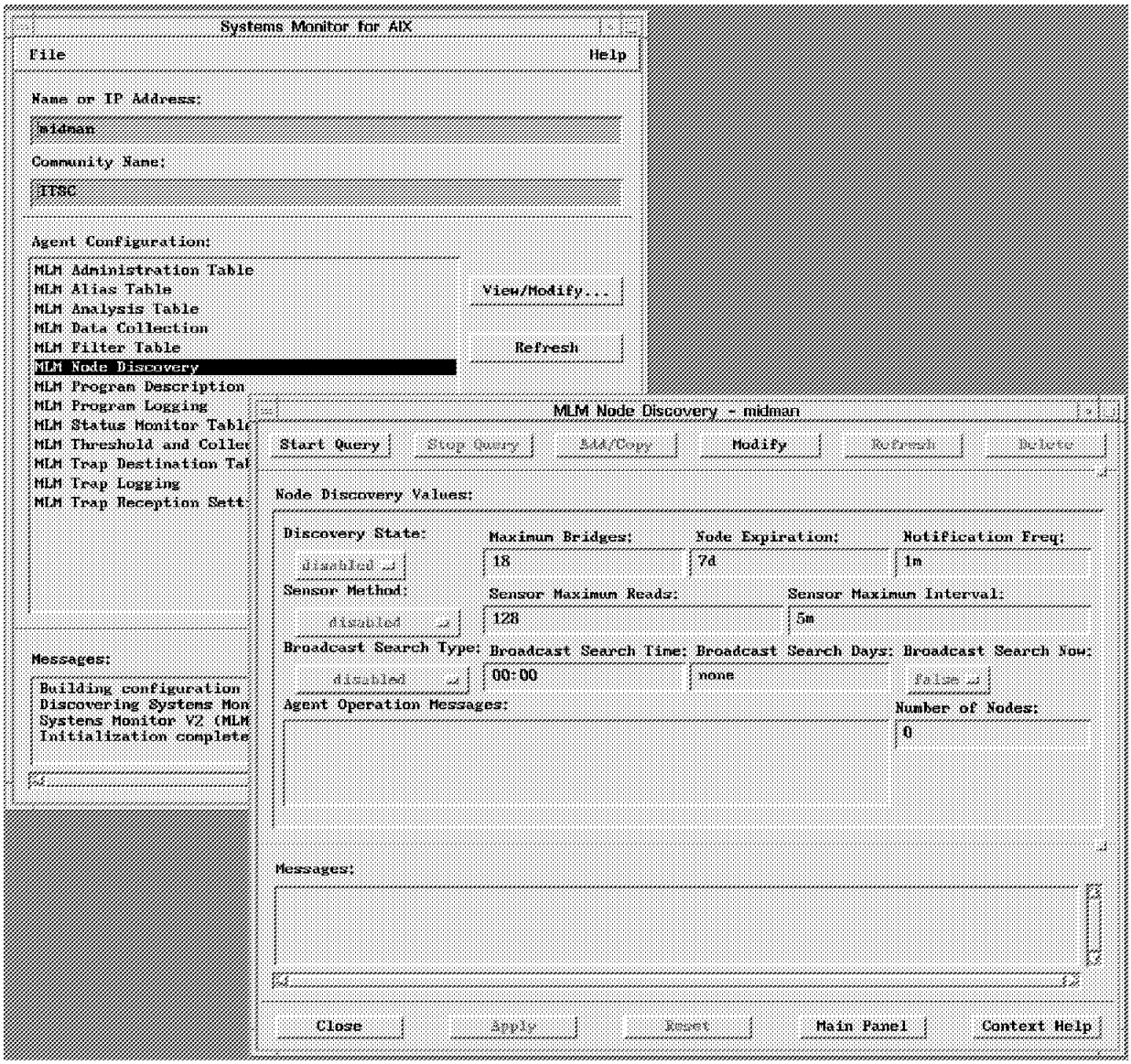


Figure 112. Accessing the Node Discovery Table

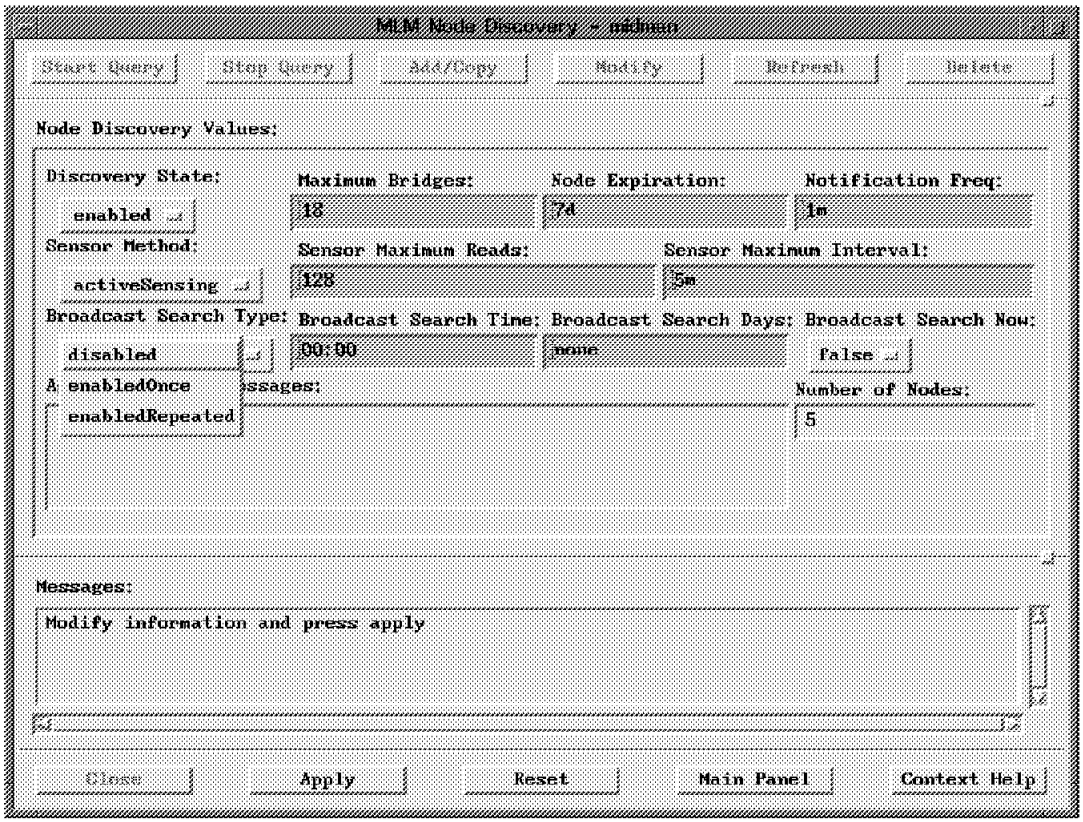


Figure 113. Enabling Node Discovery. We have enabled discovery with active sensing. In this panel you can also see the broadcast search options. The smallest broadcast poll interval you can set is 00:01 (one minute).

A few minutes after we had activated discovery, we saw Node Added events appearing in NetView for AIX. These were normal netmon-generated events, but they were triggered by traps from the MLM. You do not see these traps from the MLM in the event display because they are defined to be log-only.

4.4.5 Mid-Level Manager Status Polling with Manager Fallback

NetView for AIX Version 3 introduced a new Manager Fallback capability. This allows two or more machines running NetView for AIX to be assigned as primary or fallback managers for specific parts of the network. To allow manager fallback to operate, each manager has first to discover the whole network (the parts for which it is primary manager *and* the parts for which it is the backup). Then the manager fallback configuration is applied, either manually or by invoking a seedfile.

What happens if part of the network contains a Mid-Level Manager in this situation? When the managers discover the MLM, they will each define themselves in the trap destination file and populate Status Monitor Table with their own alias lists. Figure 114 on page 131 shows an example of a Trap Destination Table with entries for both a primary and backup manager.

You might expect that having two Status Monitor Table entries referring to the same nodes would cause duplicate polling. However, the MLM handles the two lists intelligently so that it only polls each node in the network once, as long as the polling intervals match. Any status changes are sent as traps to *both*

managers, but they only have an effect on the display of the active one, since on the backup manager the network will be unmanaged.

When manager fallback takes place, the backup manager will request current node status from the MLM.

Figure 114. Traps Destination Table with Two Managers Defined

Chapter 5. Breaking the Limits: Collections

So far, we discussed the basic things NetView for AIX Version 4 (more exactly netmon) provides to help managing Mid-Level Managers in our network. Although automatic off-loading of status monitoring to MLMs helps a lot, there are the following limitations:

- The netmon daemon tries to distribute nodes equally to all MLMs in a network, but you may want to delegate monitoring of a special group of nodes (for example, all hubs) to a particular Mid-Level Manager.
- The netmon daemon distributes only single nodes to remote Mid-Level Managers. It does not delegate control of routers to the MLMs. Under certain conditions, you might need to control even such special members of your network by a Mid-Level Manager.
- The netmon daemon does not distribute nodes to a Mid-Level Manager, who is not in the MLMs subnetwork. Even if you fill a Status Monitor Table with these nodes manually, netmon would continue its own status checking.

To overcome these limitations and allow a more flexible way of distributing management tasks to remote managers, NetView for AIX Version 4 introduces collections.

We will discuss the impact of collections on AIX Systems Monitor/6000, and show various examples of how you can use AIX Systems Monitor/6000 with collections. For information about the collection facility per se we suggest you review the related chapters in *Examples Using NetView for AIX Version 4,* SG24-4515 and the product documentation.

5.1 Object Collection Facility Summary

The Object Collection Facility is a new tool introduced in NetView for AIX Version 4. It allows objects already existing in NetView for AIX's object database to be grouped in separate submaps. Collections use definition rules to find objects in the database to be included in a collection.

A collection rule uses the following different definition types:

- Node list
- Subnet
- Object attribute
- Another collection rule

All these types can be combined using simple Boolean operators (logical AND and OR).

Once a collection is defined, objects matching the collection rule will be included in the collection. An object, that no longer fits under a collection rule will be removed from the collection.

NetView for AIX automatically updates status changes of objects into submaps created by collections. This allows network administrators to group objects into collection submaps and use these collections for monitoring purposes.

You can build collections using the Collection Editor. Specific collections will be automatically built by NetView for AIX. Also, there are APIs available, which

allow to you generate, manipulate and maintain collections from other applications.

5.1.1 The Collection Editor

The Collection Editor is an executable program which is also integrated into NetView for AIX. You can start the Collection Editor from the AIX command line or by selecting **Tools** and then **Collection Editor** from the NetView for AIX menu.

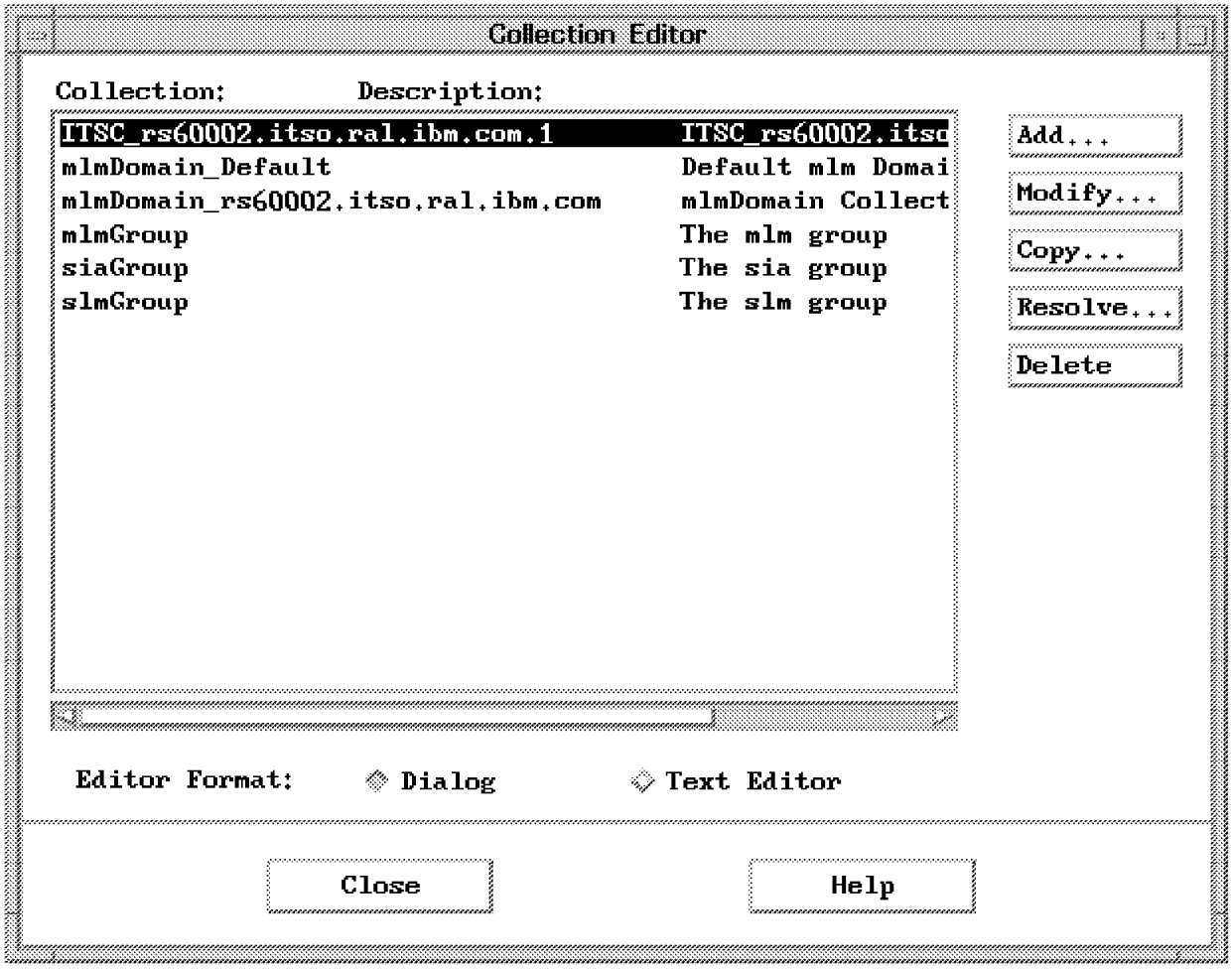


Figure 115. The Collection Editor Main Dialog

This will result in the display of an entry dialog listing all defined collections along with push buttons to Delete, Modify and Copy existing collections. Two buttons in the dialog are of special interest:

- The Resolve button allows you to get a list of all nodes a given collection rule will resolve.
- The Add button allows you to define new collections from scratch.

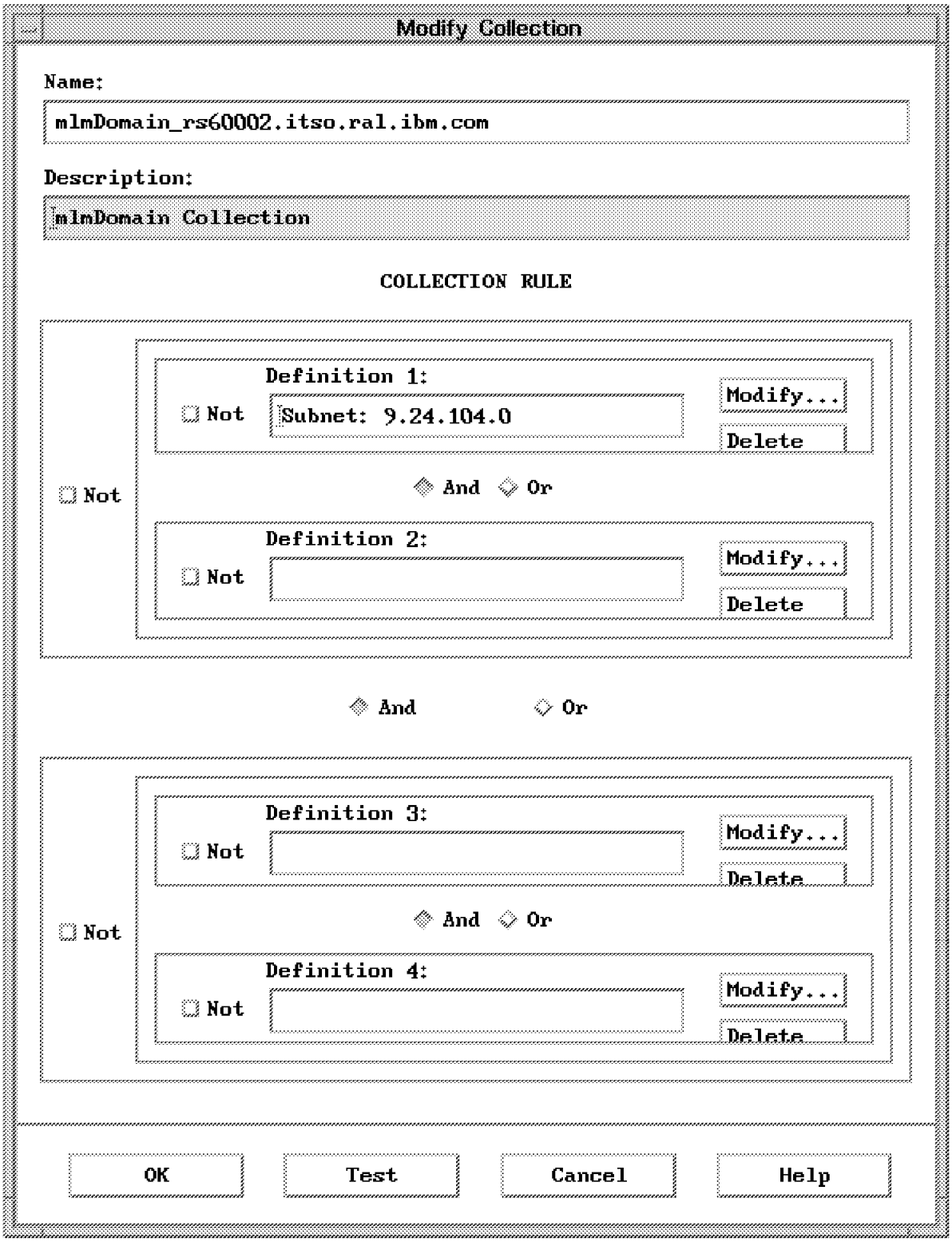


Figure 116. Collection in Dialog Format

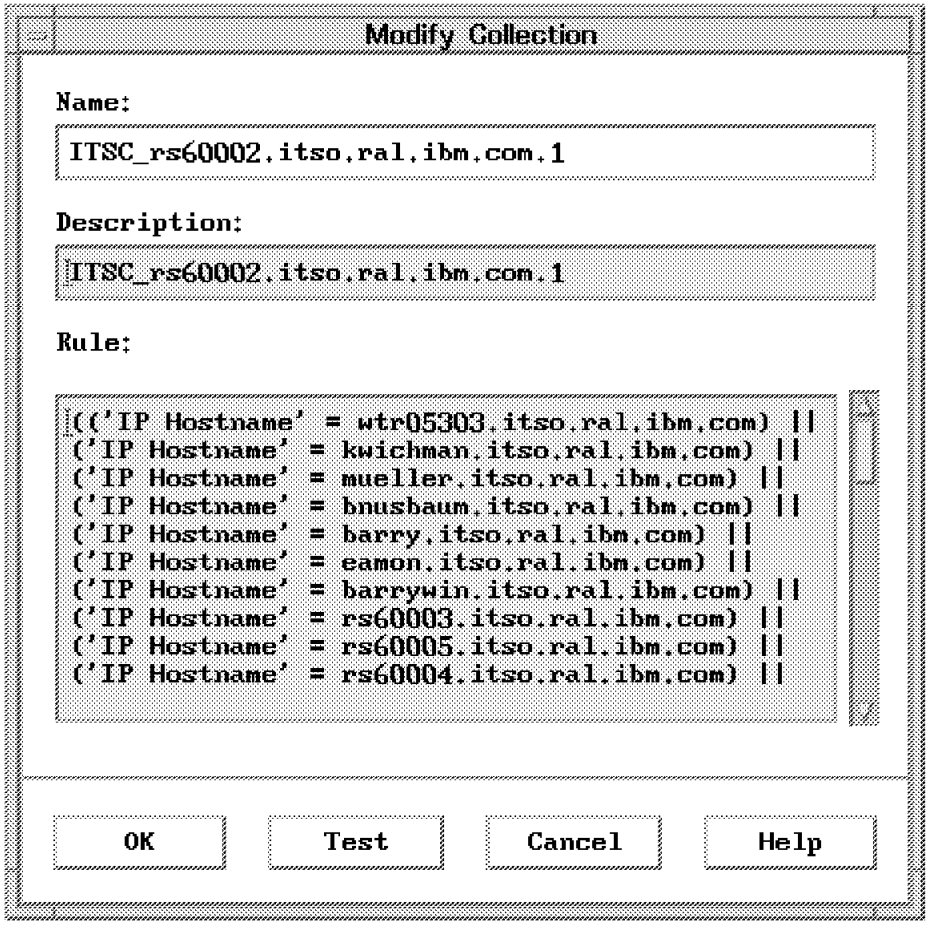


Figure 117. Collection in Text Editor Format

The Collection Editor offers two formats when you add or modify a collection rule. The Dialog format, as shown in Figure 116 on page 135, is the format you use for the most collections you define. The second format, shown in Figure 117, is a free style format and requires knowledge about the syntax of collection rule definitions. If a collection rule becomes too complex to be displayed in the Dialog format, the Collection Editor automatically selects the Text Editor format.

In both formats, you build collections by using collection types and combining multiple collection types with Boolean operators. The Collection Editor accepts the following collection types:

Node List This is a list of all the nodes you want to be a member of the collection. The syntax is `IN 'node1 node2 ... nodeN'`. If you use the Dialog form, the Collection Editor will give you a warning if the node cannot be found in the object database. However, you might include a node not found in your definition anyway.

Subnet You can use this type to include complete network segments into the collection. The syntax is `IN_SUBNET <subnet_mask>`. You must form the subnet parameter by specifying complete dotted decimal IP addresses and set the host part of the address to zero.

Attribute This type specifies a matching attribute. If you use the Dialog form, the Collection Editor provides you with a list of available attributes. Once you select an attribute, a list of applicable modifiers is displayed. The syntax is '<attribute_name>' = '<value>' where <attribute_name> is a valid attribute and <value> is a valid modifier for that specific attribute.

Collection Rule An already defined collection rule can be part of another rule. Including complete collection rules allows you to define more complex collections. The syntax is IN_COLLECTION <collection> where <collection> specifies the name of an already defined collection.

To combine multiple collection types, the Collection Editor uses one unary Boolean and two binary Boolean operators:

|| Logical OR

&& Logical AND

! The NOT operator

You can use the Boolean operators to connect collection types, and you must use parentheses to group sub-expressions.

5.2 In Case of Accidents

Building collections requires the Collection Editor to build more or less complex expressions. You always need to think about what to include in an expression, place parentheses at their right place and use the correct Boolean operators. Using the Collection Editor dialog places the parentheses for you, but it is still easy to choose the wrong Boolean operator.

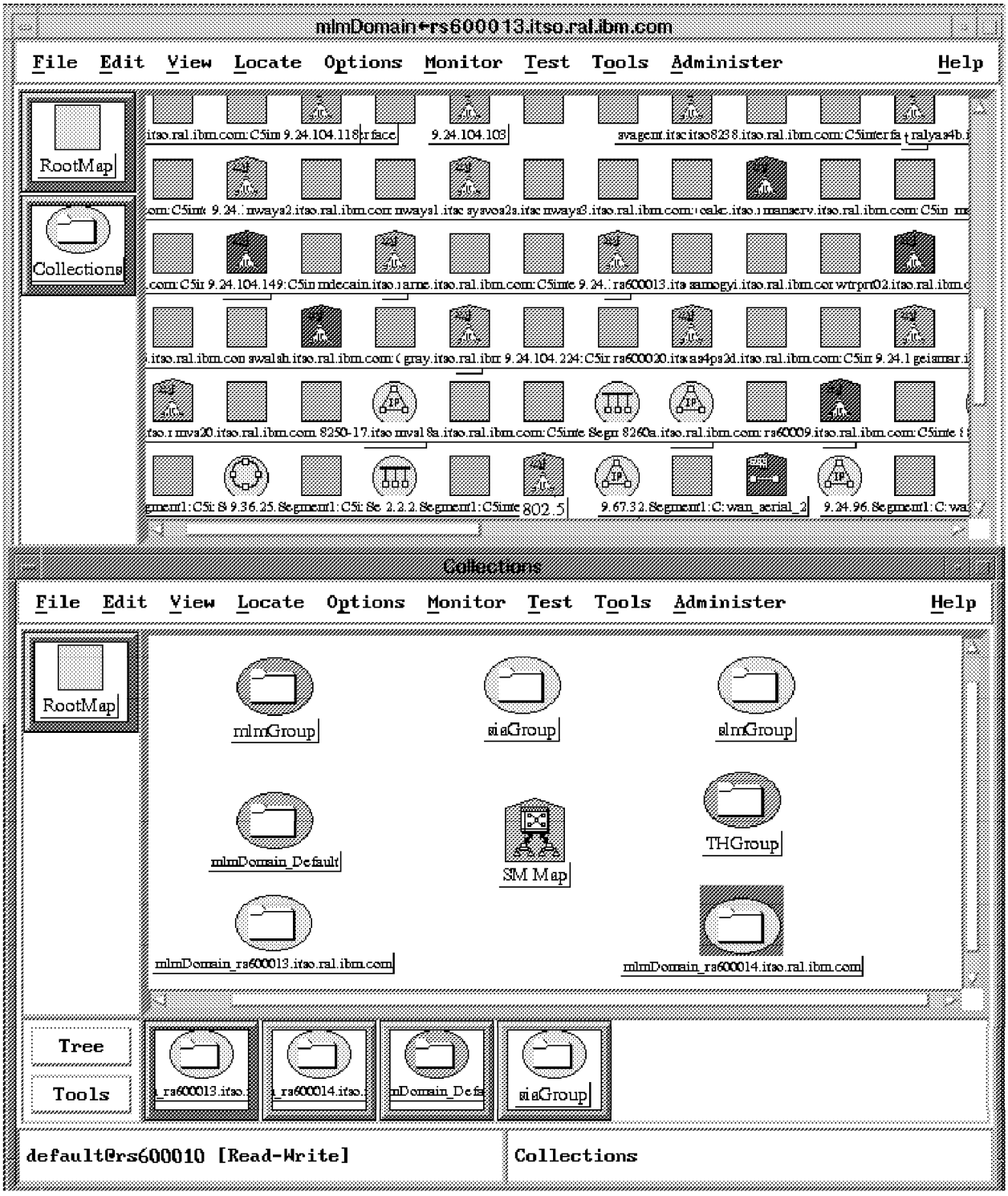


Figure 118. Result of an Incorrect Collection Definition

Wrong parentheses or an OR (||) operator placed in an expression where an AND (&&) would be correct could lead to a collection containing almost everything. A fine example is shown in Figure 118. We produced this collection accidentally when we were developing the examples in this redbook. The collection we wanted to build was not a sophisticated one, just having all hubs and routers excluded from the MLM who manages our subnet.

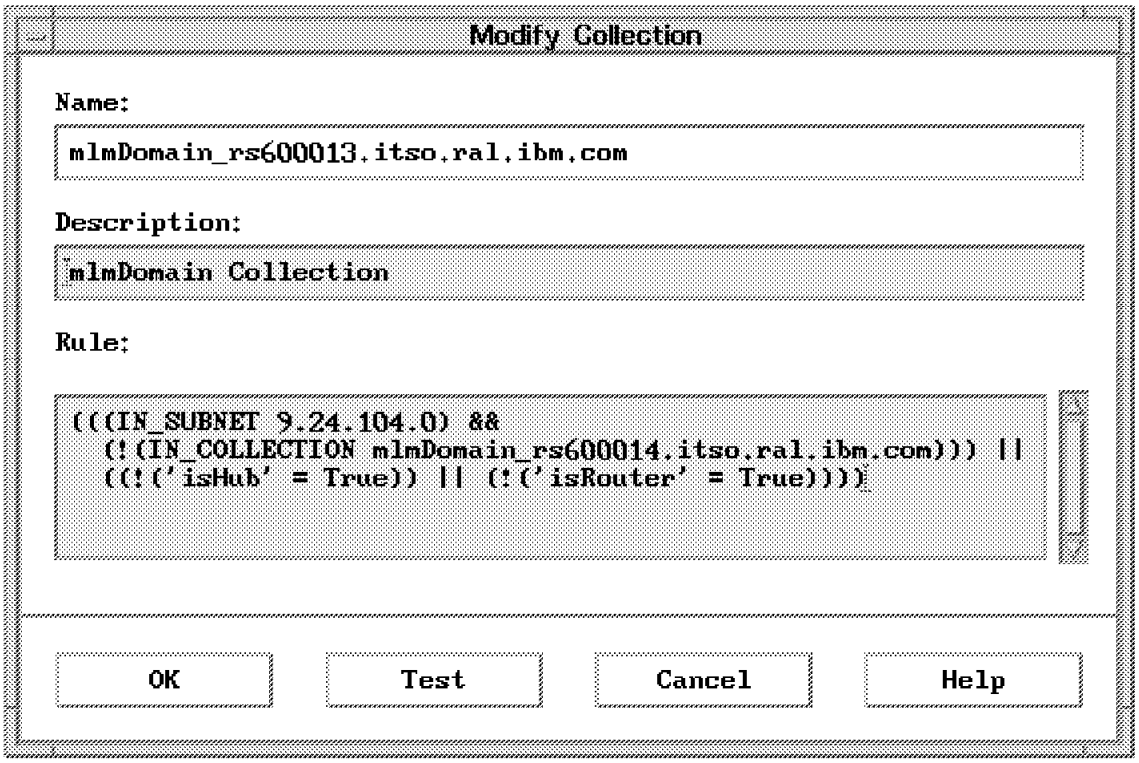


Figure 119. Incorrect Collection Definition

Figure 119 shows the collection definition. The collection definition itself was expressed using the dialog. The mistake we have made was connecting the two attribute type expressions shown in the last line with the OR operator.

This single operator finally produced the collection as shown in Figure 118 on page 138. Once a collection definition is submitted to the Collection Facility, the object database is searched and every symbol is compared against the definition. Matching objects are included in the collection. In our case, matching objects were all the objects in the database which didn't have the isRouter or the isHub attribute set. That means, even card symbols and unmanaged objects have been included into the collection.

Unfortunately, searching the object database and building the collection is a fast process. Eventually you will see the first evidence of a wrong expression when a card symbol is inserted into submaps where they really don't belong.

So, what can we do to correct such a collection and prevent the building process of unwanted collections in the future? The solution is simple and easy. Correct the expression using the Collection Editor, use the Test button and examine the results before submitting a definition.

Be patient once you submitted the corrected definition. While building a collection is fast, changing a collection, particularly a big one, takes some time. Depending on the size of the database, the repair process may take minutes to hours. In our case, the collection contained about 950 objects. It took around more than 1 hour to correct the error.

5.3 Using MLM Domain Collections to Distribute Status Checking

The netmon daemon looks for collections that are named with the format prefixmlmhostname, where:

prefix Is the string you specify with the netmon -C option or provide it when you use SMIT to update daemon options, thus changing the local registration file of that daemon. The prefix is prepended to the MLM hostname. If the -C option is not specified, netmon looks for collections using the default prefix, mlmDomain_.

mlmhostname Is the hostname of the node executing the Mid-Level Manager.

Assume that you configured netmon with an MLM Domain Collection prefix of ITSC_. If your network contains an MLM with the hostname mlm1.city.company.com, the netmon daemon will look for a collection named ITSC_mlm1.city.company.com. The netmon daemon assumes that the collection contains a list of nodes that the MLM is to manage.

Unlike its standard behavior, netmon will give all of the nodes′ interfaces to the MLM for status checking.

To keep synchronized, netmon tries to make a connection to the collection facility daemon, nvcold, during initial configuration, at configuration check time, and during the demand poll of the MLM node.

The MLM domain collections are not created automatically. If you are using the Agent Policy Manager (APM) for thresholding or file monitoring, you can use the same MLM domain collections that APM automatically creates. However, these collections use the default prefix, mlmDomain_. For more information about APM, see Chapter 8, “Introduction to Agent Policy Manager” on page 261. Alternatively, you can create MLM domain collections using the netmon -a 50 option (see 5.5, “Example Using the Netmon Daemon to Create Collections” on page 143).

5.3.1 Rules Regarding How the Netmon Daemon Uses MLMs

At a glance, the following are the rules that netmon uses for status checking:

- If the collection facility daemon, nvcold, is running *and* MLM domain collections are defined for any of the MLMs that netmon is using, netmon will use the collections to distribute status checking to the MLMs. If netmon discovers that an MLM is not operational, netmon will take over control of status checking of those nodes defined in the collections.
- If the collection facility daemon, nvcold, is not running *or* if there are no MLM domain collections defined, netmon will use the default method for distributing status checking to MLMs. That is, the netmon daemon will automatically distribute the nodes in a segment equally to the MLMs.

You *cannot* use a combination of these methods to distribute status checking. The netmon daemon either uses collections to distribute status checking or it automatically distributes status checking.

A Simple Scenario

Consider the following conditions:

- There are three MLMs in the network, rs60002, rs60003 and rs600013.
- The netmon daemon is configured to use all MLMs it discovers for status checking.
- The collection facility daemon, nvcold, is running.

If there are no collections defined, the netmon daemon will equally distribute the nodes in the segment among the three MLMs.

If you create a collection named mlmDomain_rs60002, netmon distributes status checking for the nodes defined in the collection to the rs60002 MLM. The netmon daemon now assumes that it will only use collections for distributing status checking to all of its MLMs. Because there are no collections defined for rs60003 and rs600013, netmon will discard all Status Policy Table entries for those two Mid-Level Managers and will not update them until you define collections for these MLMs also.

If netmon discovers that rs6000 is not operational, netmon will take over control of the status checking for rs60002. As soon as rs60002 is operational again, netmon will distribute status checking based on the mlmDomain_rs60002 collection.

5.4 Preparing NetView for AIX to Work with Collections

Up to this point, we were using just the netmon daemon and its capabilities to distribute some work to remote Mid-Level Managers. With collections, almost everything changes. In addition to netmon, two more daemons, nvcold and C5d, will interact with remote managers. In 4.3.1, “Example Using netmon to Distribute Workloads to MLMs” on page 115, we prevented nvcold from being started at each startup of NetView for AIX. Now it is time to reactivate the daemon.

```
rs600010:/u/peterg > cd /usr/OV/lrf
rs600010:/usr/OV/lrf > ovaddobj nvcold.lrf
ovaddobj - Static Registration Utility
Successful Completion
rs600010:/usr/OV/lrf > ovstart nvcold
rs600010:/usr/OV/lrf >
```

Figure 120. Registering nvcold

As Figure 120 shows, NetView for AIX′s static registration utility along with the according registration file nvcold.lrf is used to register the daemon. From now on, it will be started upon each ovstart you issue to start up NetView for AIX.

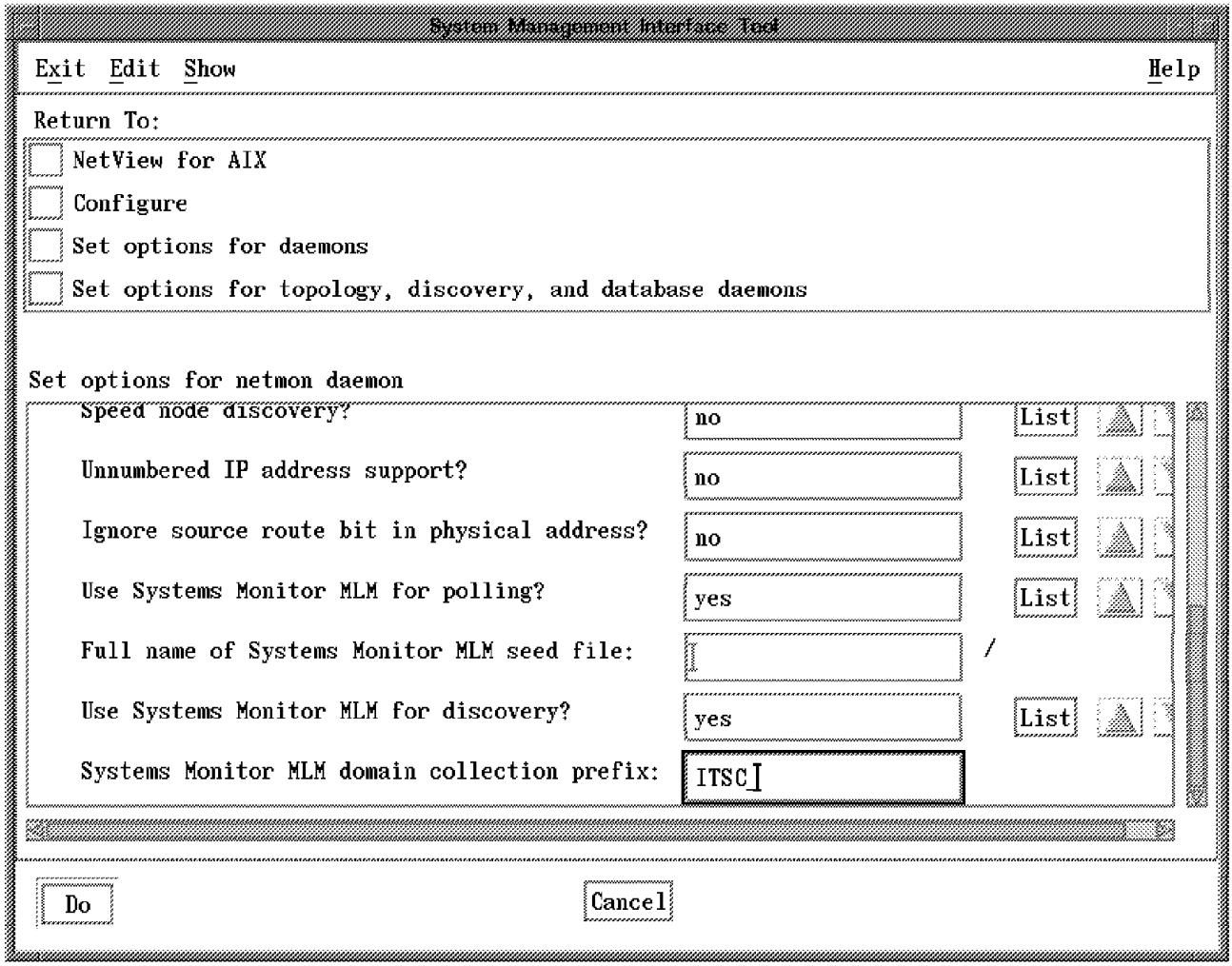


Figure 121. Specifying an MLM Domain Collection Prefix

To identify defined collections, we use an MLM collection prefix of ITSC_. You can specify this prefix via SMIT. Execution of the SMIT dialog causes netmon to be stopped. Then the local registration profile of netmon, /usr/OV/lrf/netmon.lrf, gets updated with the MLM Domain collection prefix. Before finishing, SMIT will restart the daemon.

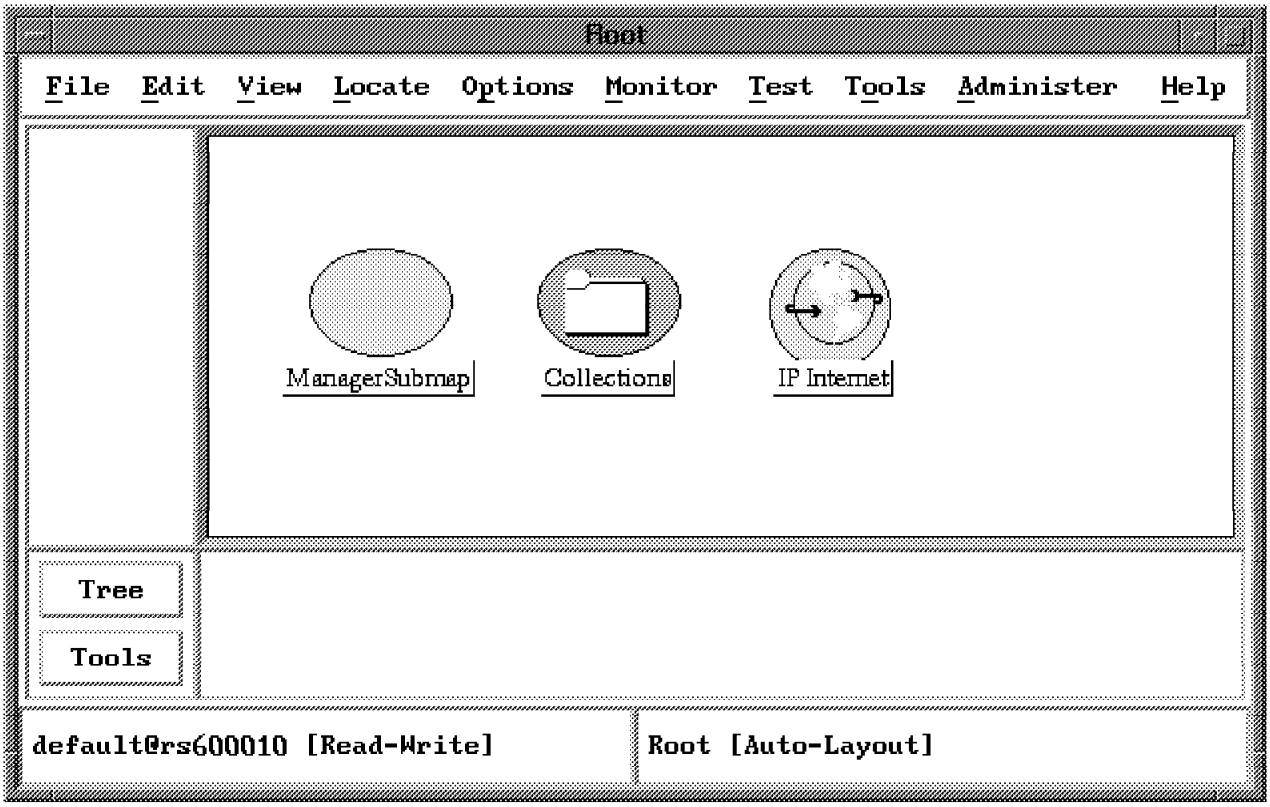


Figure 122. Collection Root Map Icon

After activating the nvcold daemon, you can start NetView for AIX or restart it if it was already started.

Note

Additionally, we executed a new map generation to prepare NetView for AIX for the collection scenarios. In a live customer environment, perhaps with maps containing custom objects such as location symbols and their submaps, issuing a new map generation would be disastrous.

After NetView for AIX is up again, the rootmap contains another Root Symbol named Collections as shown in Figure 122. At its initial start, nvcold will insert the icon into the root map and attach an empty submap to it. We will fill this submap in the following chapters.

5.5 Example Using the Netmon Daemon to Create Collections

The following example assumes that you did not use the collection facility yet, that you have set up NetView for AIX as described in 5.4, “Preparing NetView for AIX to Work with Collections” on page 141, and that nvcold is running.

```
rs600010:/u/peterg > ovstatus nvcold
 object manager name: nvcold
 behavior:            OVs_WELL_BEHAVED
 state:               RUNNING
 PID:                 19112
 last message:        Initialization complete.
 exit status:         -
rs600010:/u/peterg >
```

Figure 123. Checking nvcold Status

You may check the status of nvcold using the ovstatus command. The result should look like the console output in Figure 123.

Starting with NetView for AIX Version 4, netmon introduces a new command option.

```
netmon -a 50
rs600010:/u/peterg > netmon -a 50
```

Issuing the netmon command causes netmon to build collections depending on the current distribution scheme. The netmon daemon must be up and running before issuing the command.

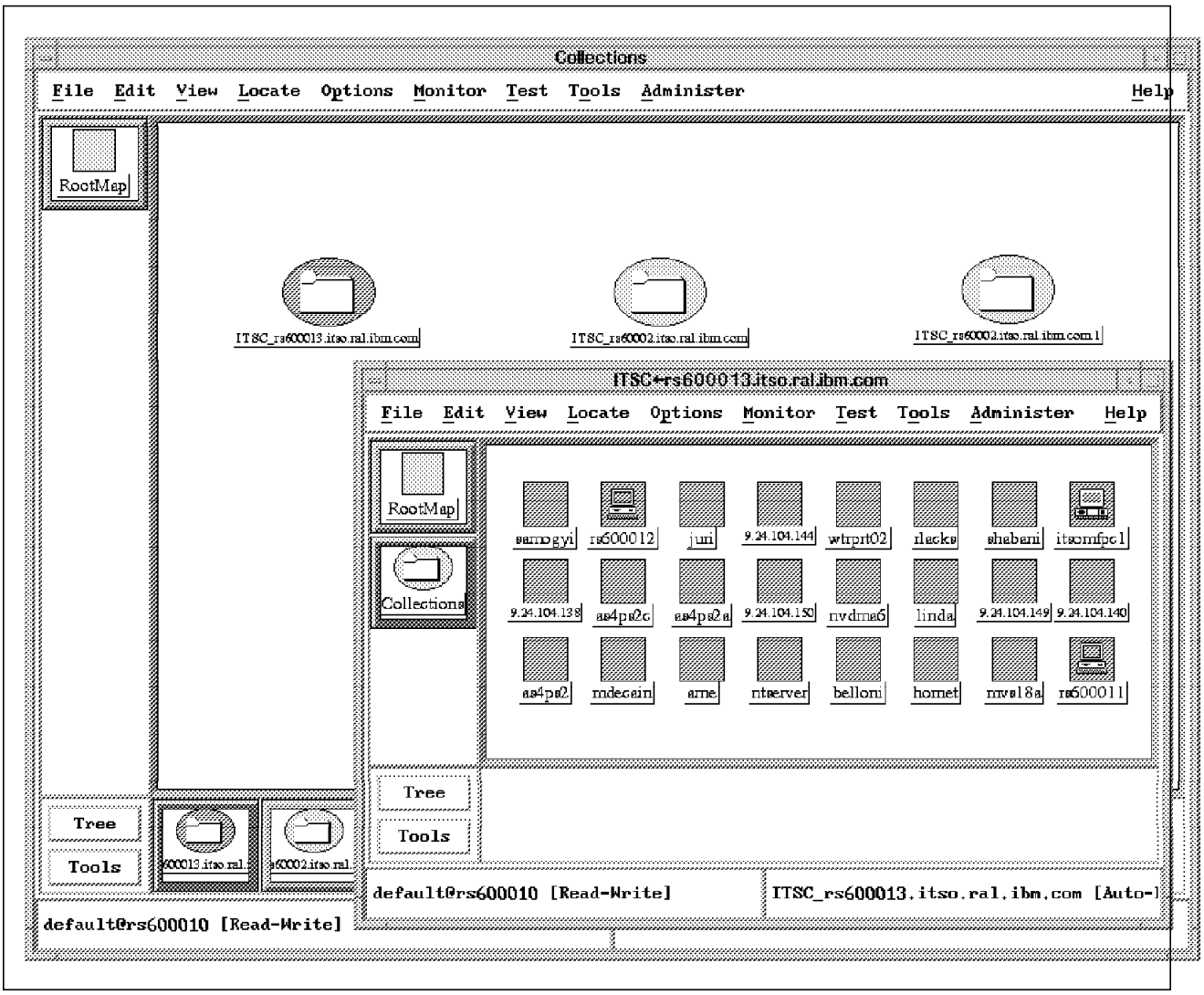


Figure 124. Collections Produced by the Netmon Daemon

By now, you should see entries in the Collections submap. The netmon daemon should have generated collections based upon its current distribution status. All the created collections will be pre-appended by the defined MLM Collection prefix as specified in 5.4, “Preparing NetView for AIX to Work with Collections” on page 141. In our example, we specified ITSC_.

Double-clicking on one of the Collection symbols now should open a submap containing all the nodes the Mid-Level Manager manages. You then may use the Collection Editor to have a look at how netmon defined the collections or even modified its entries.

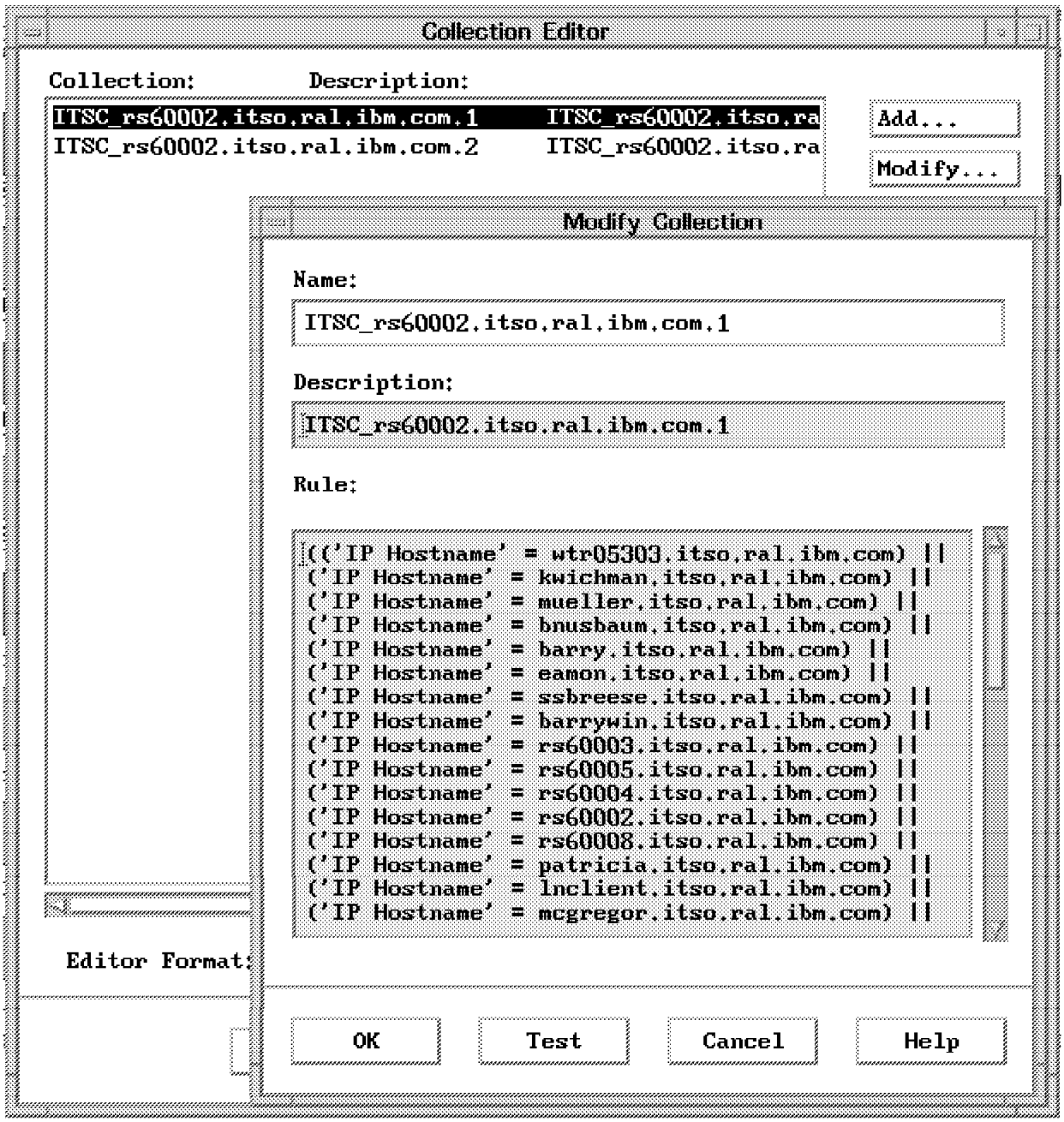


Figure 125. Part of the Collection Rule Built by the Netmon Daemon

The netmon daemon uses a rather simple approach building the collections as you can easily see from Figure 125. It uses only the IP Hostname attribute to list all the nodes it has given to the Mid-Level Manager for status checking. Perhaps this is not very flexible and you might use the generated collections just to start your own set of collections.

Chapter 6. Other Functions of the Mid-Level Manager and System-Level Manager

In Chapter 4, “Using the Mid-Level Manager for Status Polling” and Chapter 5, “Breaking the Limits: Collections” we looked at the status polling and node discovery features of Mid-Level Manager. In this section we will discuss its other capabilities for distributing management function. Most of these capabilities are also provided by the System Level Manager but the SLM can only poll the node it is installed on.

6.1 MIB Processing Tables

The MLM processing tables can be used together to effectively configure the Mid-Level Manager to collect, threshold and analyze data, as well as collecting and filtering traps. Several of the tables can work in isolation, but are often most effective when they are combined with entries in other tables to perform specific tasks. The relationship between the tables is shown in Figure 126 on page 148.

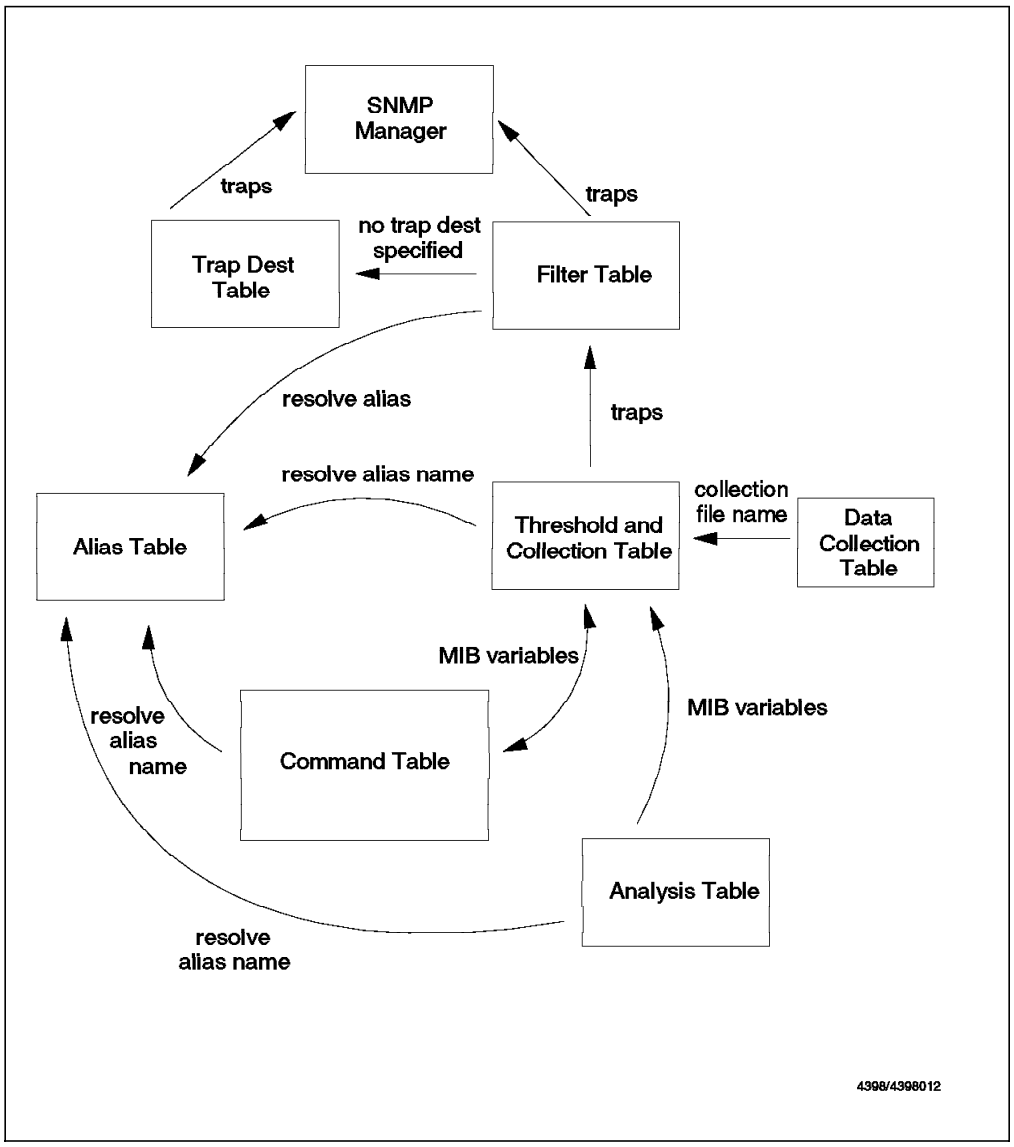


Figure 126. Mid-Level Manager MIB Processing Tables

It can be seen from this diagram that all the tables are inter-related and use variables from each other. For example:

- Whenever the Analysis, Command, Threshold, Filter and Trap Destination Tables need to use node addresses, they may refer to the Alias Table to resolve alias names into address lists.
- The Command Table and Analysis Table can input the results of their calculations to the Threshold Table, which then decides, based on its own configuration, whether to store the data and whether or not a trap should be sent to the SNMP manager if a threshold is exceeded.
- If it decides to store the data, the Data Collection Table provides input on where the data collection files are located, and the behavior of those files.
- If the Threshold Table is configured to send a trap to the SNMP manager, the Mid-Level Manager looks in the Filter Table to see if the trap matches any of the configured filters. If it does, and the filter is set to send the trap, or it does not match a filter, the trap is sent to the SNMP manager.

- The Trap Destination Table is then used to determine which SNMP manager the trap should be sent to, unless the matched filter specifies a trap destination, in which case this table need not be referred to.

The situation for the System Level Manager is rather simpler. It can only poll the node it is installed on, so the Alias Table is not supported. The other tables operate in exactly the same way as on the Mid-Level Manager.

Each of the tables will now be described in more detail. Examples of how they can be configured can be found in Chapter 7, “Systems Monitor Examples” on page 161.

6.1.1 Alias Table

This table is used to specify an alias name that will be used to describe a group of nodes. These alias names may then be used in other Mid-Level Manager tables. For example, the Threshold Table, the Status Policy Table and the Analysis Table, can use aliases defined in the Alias Table. As we have already noted, the Alias Table is not supported by the System Level Manager. A sample of one of the aliases that was configured for our network is shown in Figure 127 on page 150.

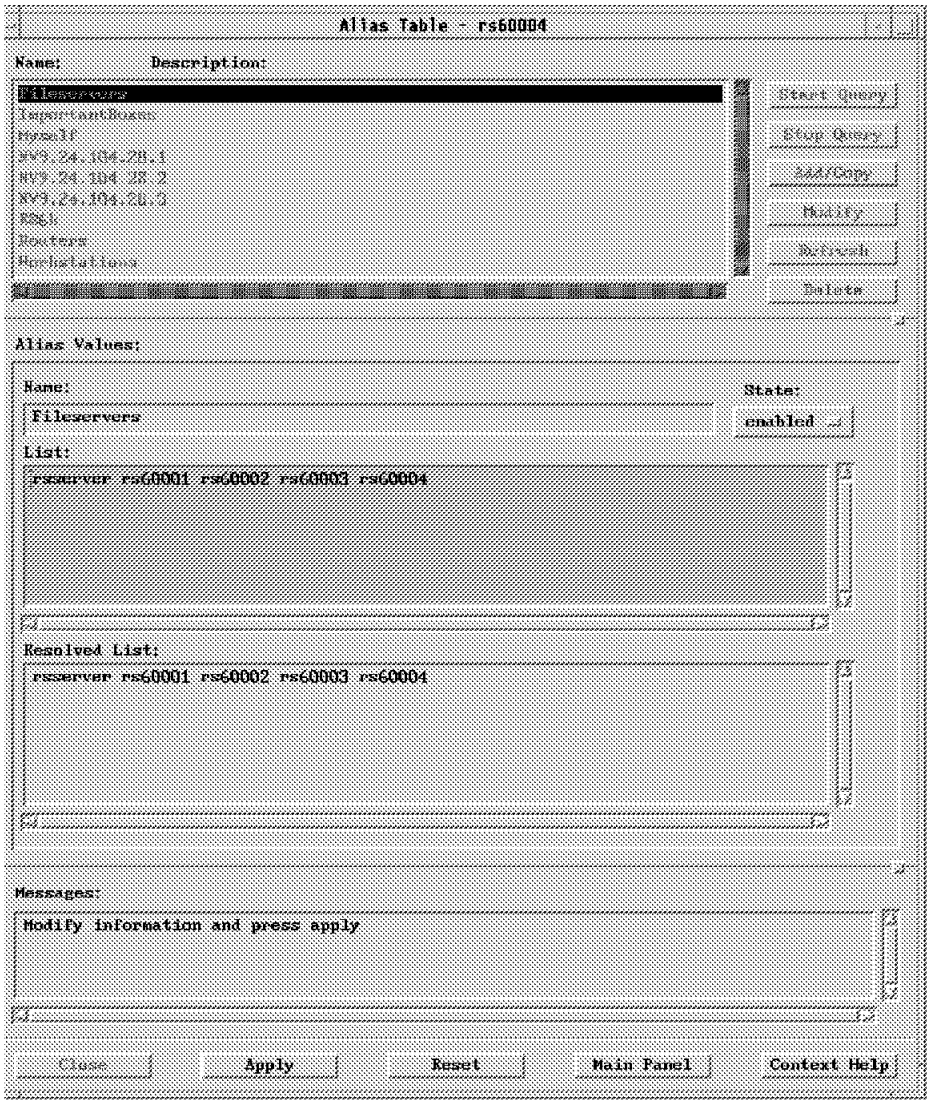


Figure 127. Alias Table Entry, Fileservers

The Name field corresponds to the name of the alias (and is also the instance ID for the table) and the List field contains the list of nodes or IP addresses that are associated with that alias. The names are verified using the TCP/IP gethostbyname API call and the verified names or addresses are copied into a second field called the Resolved List.

Examples

Examples of how this table can be used in conjunction with other Systems Monitor for AIX tables, can be seen in 7.1.2, “Monitoring Status of the lpd Daemon” on page 165 and 7.3.1, “Monitoring Paging Space” on page 220.

6.1.2 Analysis Table

This table is used to perform complex arithmetic expressions on MIB variables. In this way an SNMP manager (or MLM) can retrieve an arithmetic combination of several MIB instances, with one SNMP GET. The result of the Analysis Table can then be used as input to other tables, for example the Threshold Table.

The result of the analysis can be an integer, a counter or a gauge. Figure 128 shows the configuration screen for the Analysis Table with these options displayed.

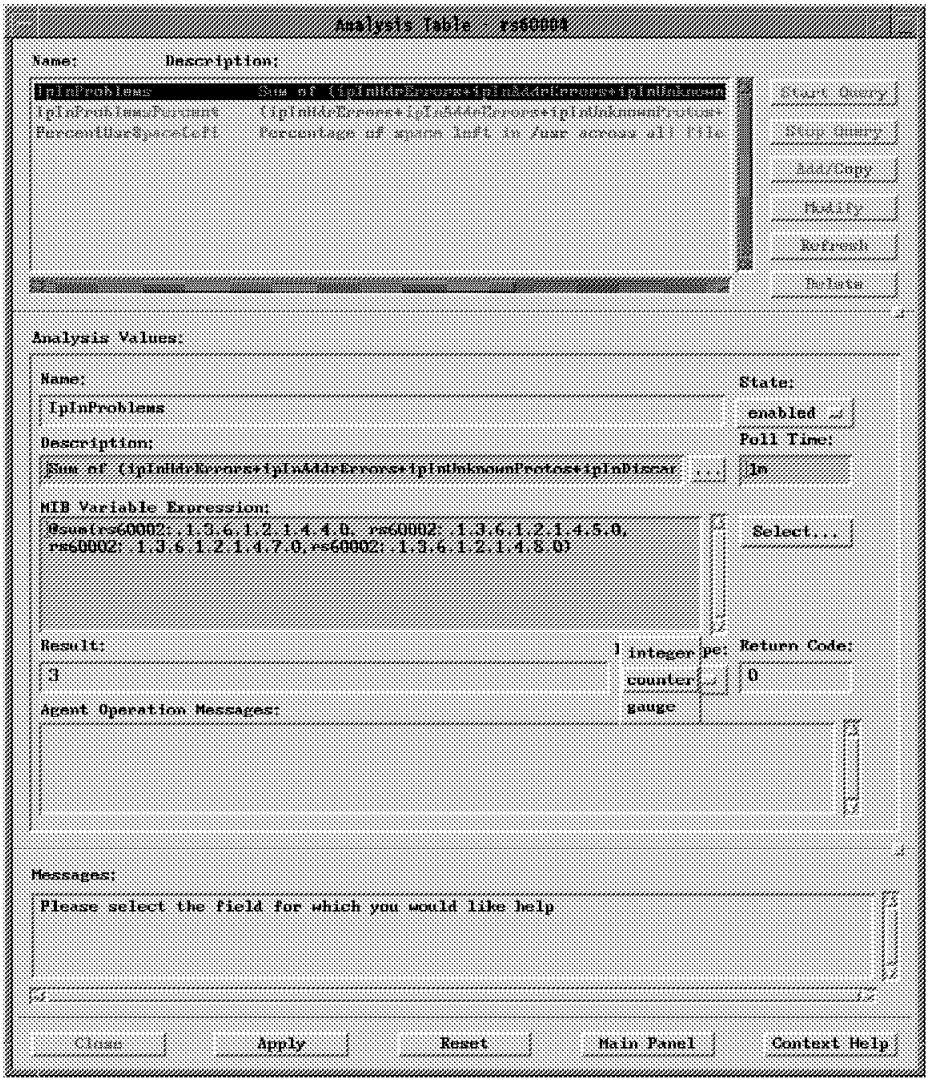


Figure 128. Analysis Table

The Data Type button has been pressed in this figure to show the three options. It is important that this is set correctly, since it is this variable that the other tables will use to monitor the output of the expression.

The poll time specifies how often the Mid-Level Manager will perform an SNMP get on the node (in this case rs60002) for the result of the expression. If the Start Query button is selected then the analysis will be performed immediately, and the result will be displayed in the result field.

The analysis can be performed on an individual host, or on a group of nodes by using an alias.

Examples

An example of using this table can be seen in 7.3.7, “Monitoring the Percentage of IP Datagrams in Error” on page 246, where the Analysis and Threshold Table are used together.

A detailed listing of the various expression operators that are available and how expressions are formed can be seen in *Systems Monitor for AIX Users Guide*, SC31-7150 and also by using the context help facility of Systems Monitor for AIX for the field titled MIB Variable Expressions.

6.1.3 Threshold and Collection Table

This table is one of the most important components of the Mid-Level Manager. It has the following two functions:

1. It can be used to collect and store MIB data from any node in the IP network.
2. It can be used to compare the collected data to a specified threshold and generate an alert, and/or run a command, when the threshold is exceeded.

The Threshold Table with a sample entry can be seen in Figure 129 on page 153, with a list of the available options that can be used to specify whether data is stored, thresholded, both or neither.

Figure 129. Threshold Table

The Threshold Table can be used to monitor MIB instances of any kind:

- Objects defined in the Systems Monitor for AIX MIB
- Objects that have been user-created by means of the Command Table or the analysis table
- Objects in MIB-II
- Objects in standard or private MIB extensions

For example, if a LAN hub had a private MIB containing (say) port utilization data the MLM Threshold Table could be used to poll it on a regular basis.

The MIB object that is to be monitored should be entered in the Local/Remote MIB Variable field and the instance should be appended to the end of this object ID.

The MIB instance could be the following:

- .* to signify that *all* instances of that MIB are to be collected.
- .instancename.

The instancename is the name of the particular instance to be monitored. It can be either in dotted decimal or in the case of Systems Monitor MIB tables it could be the instance name in a text form. For example an instance name of /usr could be used if the target of the threshold entry is in the SIA File Systems Table (see 2.3.2, “Understanding Systems Monitor MIB Instances” on page 33 for an explanation of this).

Examples

Examples of these functions can be seen in Chapter 7, “Systems Monitor Examples” on page 161 and in particular 7.1.2, “Monitoring Status of the lpd Daemon” on page 165, 7.2.1, “Who su Configuration” on page 207, 7.3.1, “Monitoring Paging Space” on page 220, 7.3.2, “Monitoring Processor Utilization” on page 225, 7.3.3, “Monitoring File System Utilization” on page 227 and 7.4.1, “Collecting MIB Data with Mid-Level Manager” on page 253.

6.1.4 Data Collection Table

MIB data that is stored by the Threshold Table is saved in the /usr/adm/sm6000/collect directory, in files called midmand.col, midmand.col1, midmand.col2 etc. The number, size and behavior of the files can be configured by modifying the Data Collection Table. The Default Table configuration can be seen in Figure 130.

Figure 130. Data Collection Table

6.1.5 Filter Table

The Filter Table can be used to determine which traps are to be sent to the NetView for AIX manager, when, and to which manager. Traps can be blocked completely, throttled or sent. A sample entry with a list of these options can be seen in Figure 131.

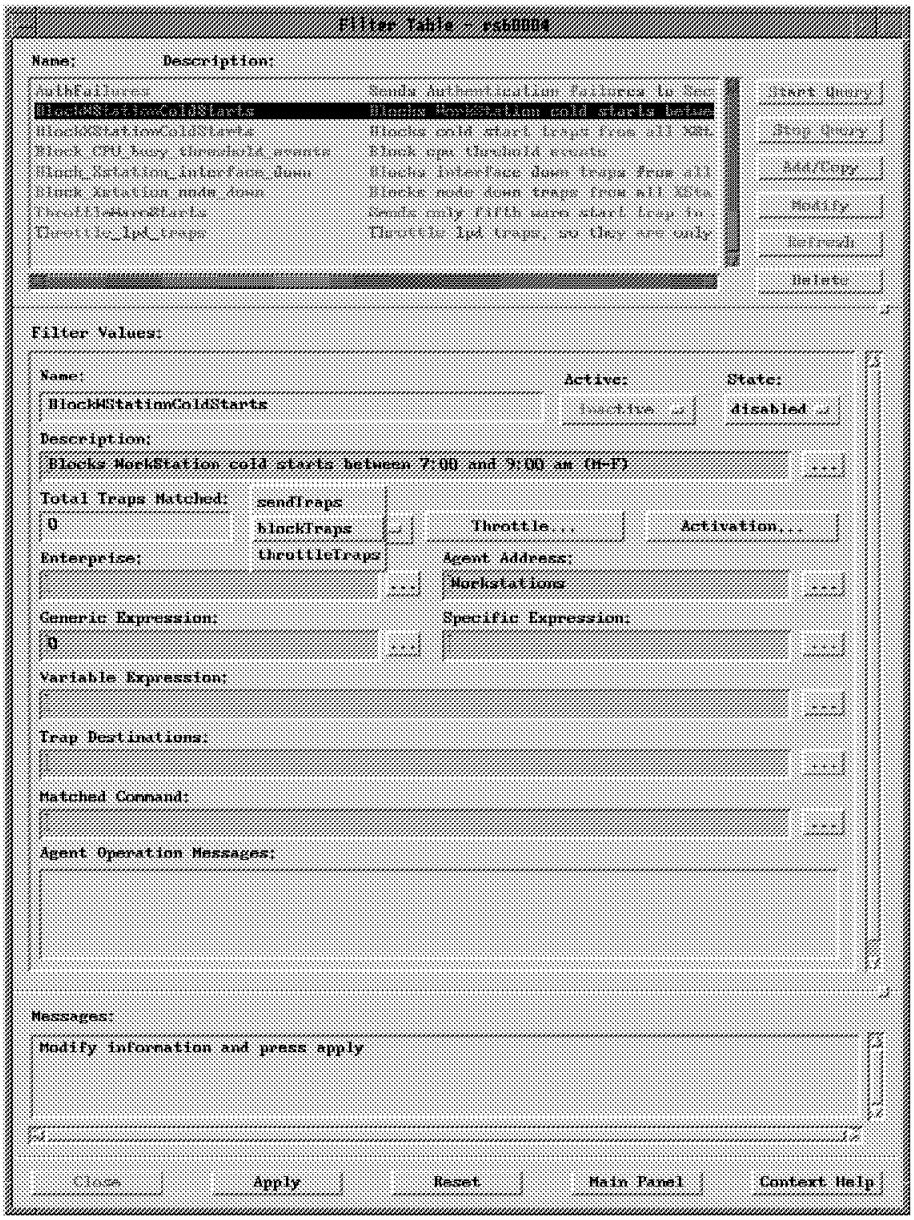


Figure 131. Filter Table

When a trap comes in, the Mid-Level Manager will look in the Filter Table to see if it matches any of the set filters. If it does match, the effect will depend on the setting of the Filter Actions field:

- If it is blockTraps the trap will not be sent.
- If it is sendTraps the Mid-Level Manager will send the trap to the node or IP address specified in the Trap Destinations field, unless this field is empty. If Trap Destinations *is empty* the trap will be sent to nodes listed in the Trap Destination Table (6.1.7, " Trap Destination Table" on page 158).

- If it is throttleTraps the number of traps that are sent depends on the throttle configuration, which can configured to send the first N traps, or only start sending the traps after N have been received.

Figure 132 shows the throttle configuration table, and an example of how to throttle traps can be seen in 7.1.2.6, “Filtering Traps” on page 177.

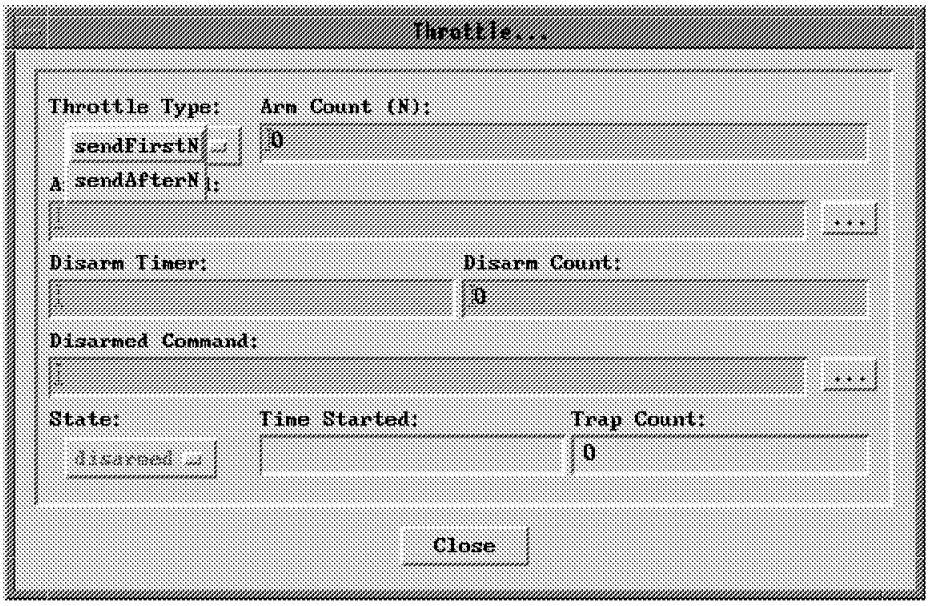


Figure 132. Filter Throttle Configuration Table

Filters can be activated at particular times of the day too, and Figure 133 shows the configuration screen, which in this case is set to apply the filter from 7:00 am to 9:00 am every weekday.

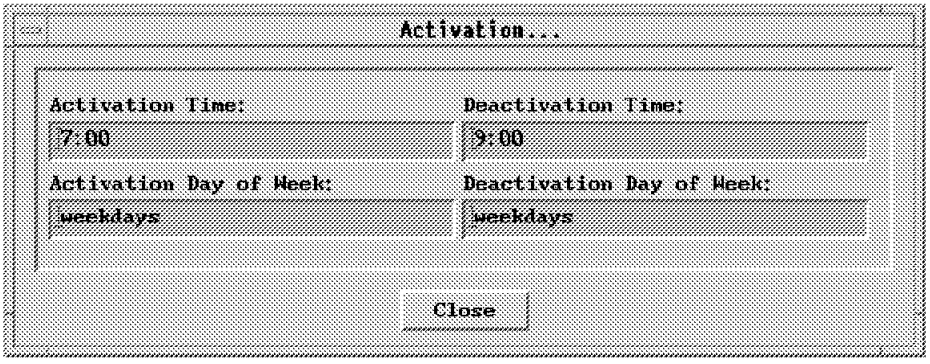


Figure 133. Filter Activation Configuration Table

Examples

See Chapter 7, “Systems Monitor Examples” on page 161 and in particular 7.1.2.6, “Filtering Traps” on page 177 for more information and examples of how to use the Filter Table.

The filter can be configured to match the following trap fields:

- Enterprise ID of the trap (the Enterprise field in Figure 131 on page 155).

- Generic trap number (the Generic Expression field in Figure 131 on page 155).

  This will always be 6, for Systems Monitor for AIX configured traps.

- Specific trap number (the Specific Expression field in Figure 131 on page 155).

- Agent address.

  This could be a hostname, IP address or alias name and is the name of the agent that generated the trap.

  Be Careful

  Some traps will be generated by the Mid-Level Manager and some by the Systems Information Agent. Make sure that you are filtering on the correct agent address, for example, interface status traps are generated by the Mid-Level Manager and not the agent whose interface was found to be down, whereas agent cold start traps (generic trap number 0), are generated by the node on which the agent was started.

- Grouping environment variables

  Variable Expression field in Figure 131 on page 155. This field can be used to group together variables (a list of which can be displayed using the Context help).

If the Mid-Level Manager receives a trap that does not match any of the filters in the Filter Table, then it looks at the Trap Reception Table to decide what to do with the trap (that is, whether to block or send the trap).

6.1.6 Trap Reception Table

This table can be used to specify the default action that the Mid-Level Manager is to take with traps that it receives that do not match any of the entries in the filter table. The possible actions are as follows:

- Send all traps
- Block all traps

In addition, this table can be used to specify which trap reception ports, and protocols the Mid-Level Manager is to receive traps from. The default is to receive traps on port 162, using both TCP and UDP protocols.

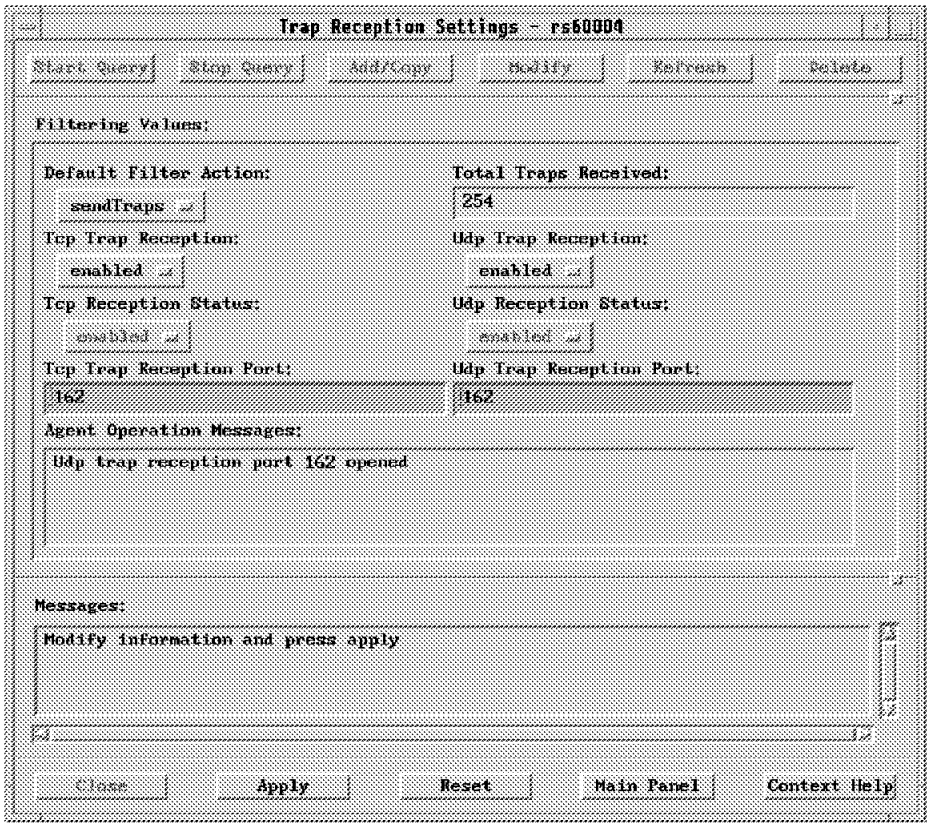


Figure 134. Trap Reception Table

The MLM receives traps from SIA and SNMP agents in the subnetwork on the UDP port, thus it is always best to leave this port enabled. Traps from other Mid-Level Manager nodes may be received on the TCP port, and it is important to enable this port in a network where there are hierarchical Mid-Level Manager nodes installed.

6.1.7 Trap Destination Table

This table lists the SNMP manager nodes that are to receive traps from the Mid-Level Manager. It is also possible to specify when traps are to be sent to the manager, on a time of day and day of the week basis, via the activation and deactivation fields. For example, the table entry shown in Figure 135 on page 159 will result in traps being sent to a NetView for AIX manager running on node rs600010, between the hours of 6pm and 8am, every day of the week.

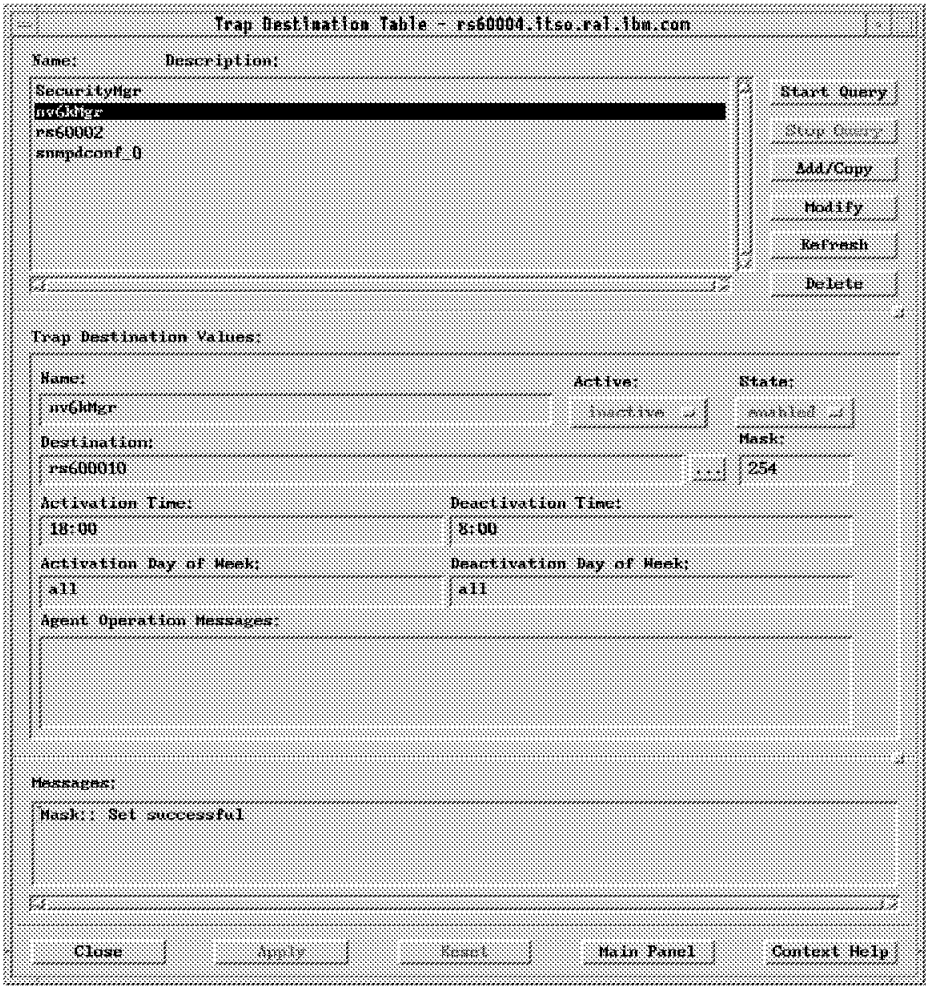


Figure 135. Trap Destination Table Entry

The Mask field is used to specify which traps are to be sent to the destination node, a mask of 254 denoting that all traps should be sent. It is also possible to specify that the Mid-Level Manager send traps to NetView for AIX using the TCP port. An example of a Trap Destination Table entry using TCP is shown in Figure 114 on page 131.

6.2 Where to Place Mid-Level Managers

The Mid-Level Manager can poll, threshold and collect data from nodes on the same subnetwork as itself, and therefore in order to off-load the NetView for AIX system to the maximum extent, a Mid-Level Manager node should be installed on every subnet in the network.

There is obviously a cost associated with doing this, and there is a balance that must be struck between the cost of installing Mid-Level Manager and the benefits and bandwidth savings that will result.

Chapter 7. Systems Monitor Examples

In this chapter we include many samples of Systems Monitor configurations that we developed during the project. You should find examples of all of the main functions of the SIA and MLM. In each case where we have used an MLM function we have also described what differences, if any, would be implied by using an SLM instead. In general, any scenario that uses the MLM to monitor one or more AIX nodes can also be implemented using the SLM. The difference is that the SLM has to be installed on every monitored node.

You may wonder how to decide whether to install the SLM on each node, or whether to use an MLM. There is a financial answer to this question: the MLM is about five times the price of the SLM, so the break-even point occurs at about six or seven managed nodes. However, there are other questions you should ask, which may affect your decision, for example:

- Do you want to use the status polling functions that only the MLM provides? If so, you will be purchasing an MLM anyway and so you do not need to also purchase SLMs.
- Is your network comprised of widely-distributed single nodes? If so, it is better to use SLMs, thereby confining polling within each monitored machine, instead of having the MLM polling across the network.
- How much automation do you want to do? The SLM can execute automatic actions directly on the monitored node, whereas the MLM will have to use remote command execution.
- Is security an issue for the monitored node? Using the MLM implies that the monitored node must allow external access for SNMP GET requests and, potentially, remote commands. In the case of a sensitive host (a firewall, for example) this may not be acceptable, in which case the SLM is a more secure alternative.

The examples that we show here use the Systems Monitor for AIX agent functions directly, to illustrate the kind of things that they can do. However, you should bear in mind that using the SIA file monitor and the MLM and SLM Threshold Table is made much easier by the new Agent Policy Manager function of Systems Monitor for AIX Version 2 Release 2. Refer to Chapter 8, “Introduction to Agent Policy Manager” on page 261 for a description of this.

Although it unlikely that you will have a requirement for a scenario that is *exactly* the same as the ones we describe, you should find these examples useful in deciding how to perform the configuration steps.

The chapter is organized in four sections:

Many of the techniques used in each section can be applied to a number of different management requirements. We include a brief summary of the examples within each section at the start of the section, to assist you if you are looking for an example of a specific Systems Monitor function.

7.1 General Systems Monitoring

The examples in this section are aimed at monitoring the health of processes and functions in a distributed AIX environment.

You will find the following examples here:

- 7.1.1, “Monitoring Printer Status” shows how to add a remote command to NetView for AIX, using the Command Table function of the SIA.
- 7.1.2, “Monitoring Status of the lpd Daemon” on page 165 extends the printing theme by showing how to automatically monitor the status of a daemon: lpd.
- 7.1.3, “Monitoring Status of Print Queues” on page 180 extends our printing automation by providing a warning of failing print queues.
- 7.1.4, “Monitoring Number of Jobs in the Queue” on page 186 completes our printer monitoring by automatically monitoring for excessive print job queues.
- 7.1.5, “Alternative Trap Destination” on page 194 shows how to send the events generated by the previous three examples to a specific ″printer management″ station.
- 7.1.6, “Monitoring /etc/hosts for Data Changes” on page 195 uses the File Monitor Table to monitor system configuration changes.
- 7.1.7, “Monitoring /etc/resolv.conf to Verify It Exists” on page 198 uses the File Monitor Table to check for correct system configuration.
- 7.1.8, “SNA Session Status Monitoring” on page 201 shows how to monitor the health of a subsystem and automate recovery.

7.1.1 Monitoring Printer Status

Printing is a fundamental part of working day life and can be the cause of stress and grief to users when printers fail or print queues become blocked up.

Monitoring the status of printers and print queues is therefore an important part of a systems administrator’s role. This example and the four that follow show an approach to monitoring and automating the print daemon (lpd), the status of print queues and the number of jobs in a queue using Systems Monitor for AIX.

First, we want to show the general printing status on a machine running the SIA. The command that we want to issue is `lpstat`. Figure 136 on page 163 shows the Command Table configuration for this.

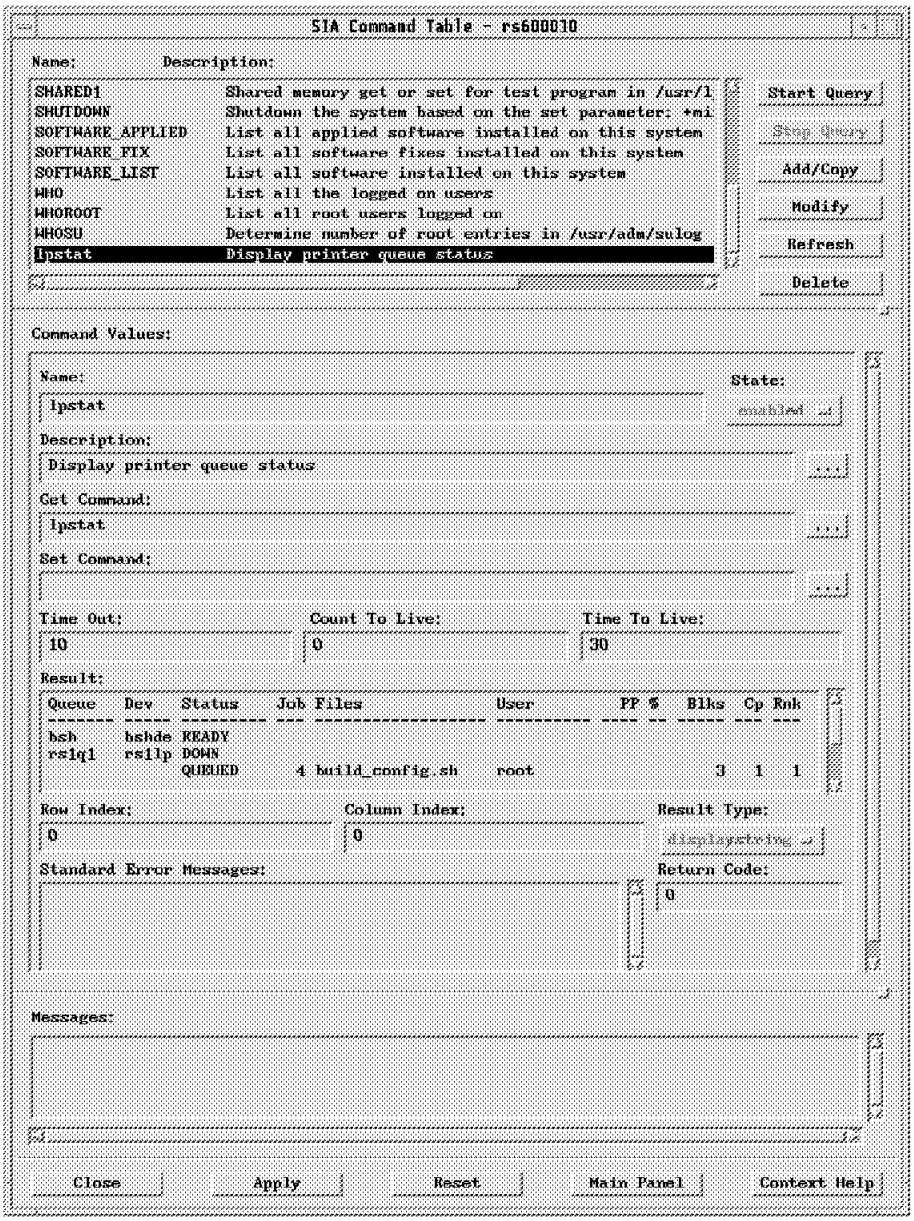


Figure 136. Defining lpstat in the Command Table

We can now execute lpstat on the remote SIA machine whenever we want, by bringing up the Systems Monitor EUI and pressing Start Query on the Command Table entry. It would be much simpler, however, to not have to start the EUI, but to be able to do this directly from NetView for AIX. We *could* use the MIB browser, but we would have to first convert the Command Table entry name into a MIB instance. In addition, the format of the response in the MIB browser would be difficult to read for long responses such as returned by the `lpstat` command.

We achieved the aim of a simple display in three steps:

1. We developed a shell script to convert Command Table entry names to instance IDs and then use the `snmpget` command to retrieve the command results.

2. We used the xnmappmon utility of NetView for AIX to format the result in a Motif window.
3. We created a registration file to add the function to the NetView for AIX menus.

The shell script, smcmd, is listed in Figure 280 on page 313. Most of its logic is involved with converting the Command Table entry name into its ASCII dotted-decimal equivalent.

The registration file, lpstat.reg, is shown in Figure 137. This file adds a new selection to the Monitor menu to invoke smcmd within a xnmappmon window. The SelectionRule specification ensures that the entry is only active if a node running the Systems Information Agent or Systems Monitor V1 has been selected. To install the menu entry, you simply have to place the registration file in the /usr/OV/registration/C directory, or its subdirectories.

```
/*
 * Registration file to add lpstat command to NV for AIX menus
*/

Application "Print Status"
{
     MenuBar "Monitor"
     {
      "Display Print Queue Status       "   f.action "lpstat";
     }

     Action "lpstat"
           {
            SelectionRule (isSIA)||(isSYSMON);
            MaxSelected 1;
            MinSelected 1;
            Command "xnmappmon -geom 700x450 \
                    -commandTitle \"Print Queue Status\" \
                    -cmd /u/raleigh/scripts/smcmd -h $OVwSelection1 -n lpstat
            }

}
```

Figure 137. NetView Registration File lpstat.reg

We activated the new menu entry by restarting the NetView for AIX EUI and then selecting the **Monitor** and then the **Display Print Queue Status** option with an SIA node selected. The resulting display is shown in Figure 138 on page 165.

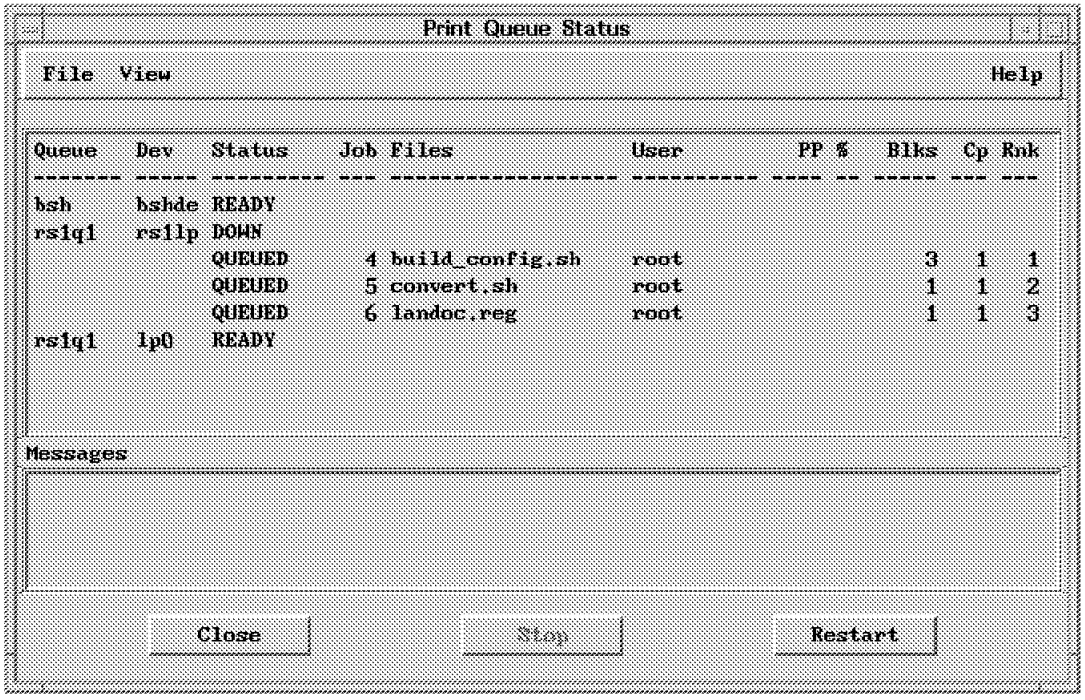


Figure 138. Result of Selecting the Menu Entry for lpstat

7.1.2 Monitoring Status of the lpd Daemon

In the previous example we have given the operator a convenient way to look at the status of the printing subsystem from the NetView for AIX network map. However, it would be preferable if the system could do the monitoring in the background and only inform the operator when something was wrong. We will examine how we can use the facilities of Systems Monitor to do this automatic monitoring for the components of the printing process.

First of all we will look at the lpd daemon and configure the Mid-Level Manager to monitor it on a regular basis. We will also configure the MLM to automatically restart it if it fails, while keeping NetView for AIX operators informed of what is happening. To achieve this we have to do the following:

1. Configure the Alias Table to add an alias for all the nodes on which we want to monitor the lpd daemon.
2. Configure the Threshold Table on the Mid-Level Manager, rs60004, to monitor the lpd daemon and send traps to NetView for AIX.
3. Configure these events on NetView for AIX.

7.1.2.1 Alias Table Entry

This table is used to specify an alias name that will be used to describe a group of nodes. These alias names will then be used in other Mid-Level Manager tables. For example, both the Threshold Table and the Status Policy Table can use them.

The alias used in this example is called RS6k and it can be seen in Figure 139 on page 166.

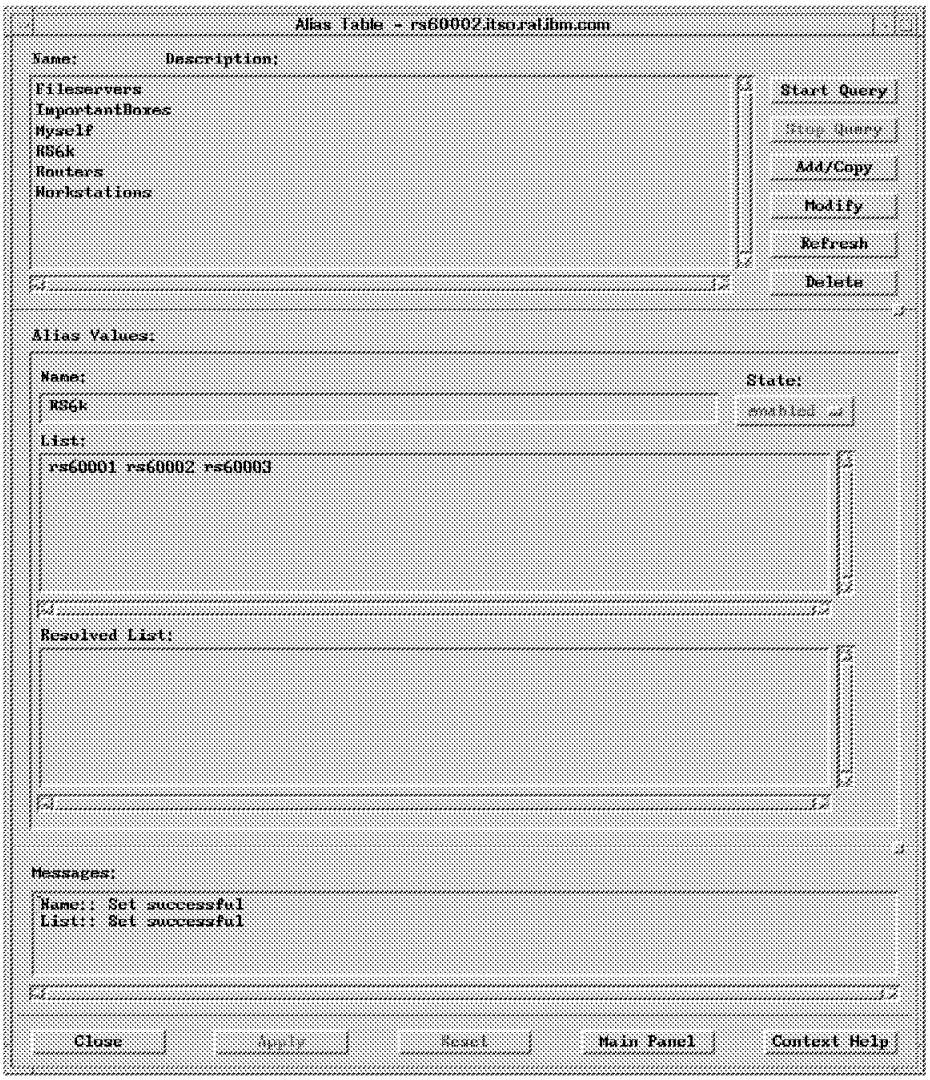


Figure 139. Entry RS6k in the Alias Table

The Name field corresponds to the name of the alias and the List field contains the list of nodes or IP addresses that are associated with that alias. These names or addresses are resolved by TCP/IP naming services to form a list called the Resolved List.

Considerations if Using an SLM in Place of the MLM: The System Level Manager cannot poll nodes other than the one on which it is running, so it does not provide an alias table to map a name to a node list. If you wanted to perform the polling for this example using the SLM instead of the MLM, you would have to have the SLM agent installed on each monitored node.

7.1.2.2 Threshold Table Entry

The entry shown in Figure 140 on page 167 was added to the threshold table to monitor the status of the lpd daemon for the alias group RS6k, which includes the nodes rs60002 and rs60004.

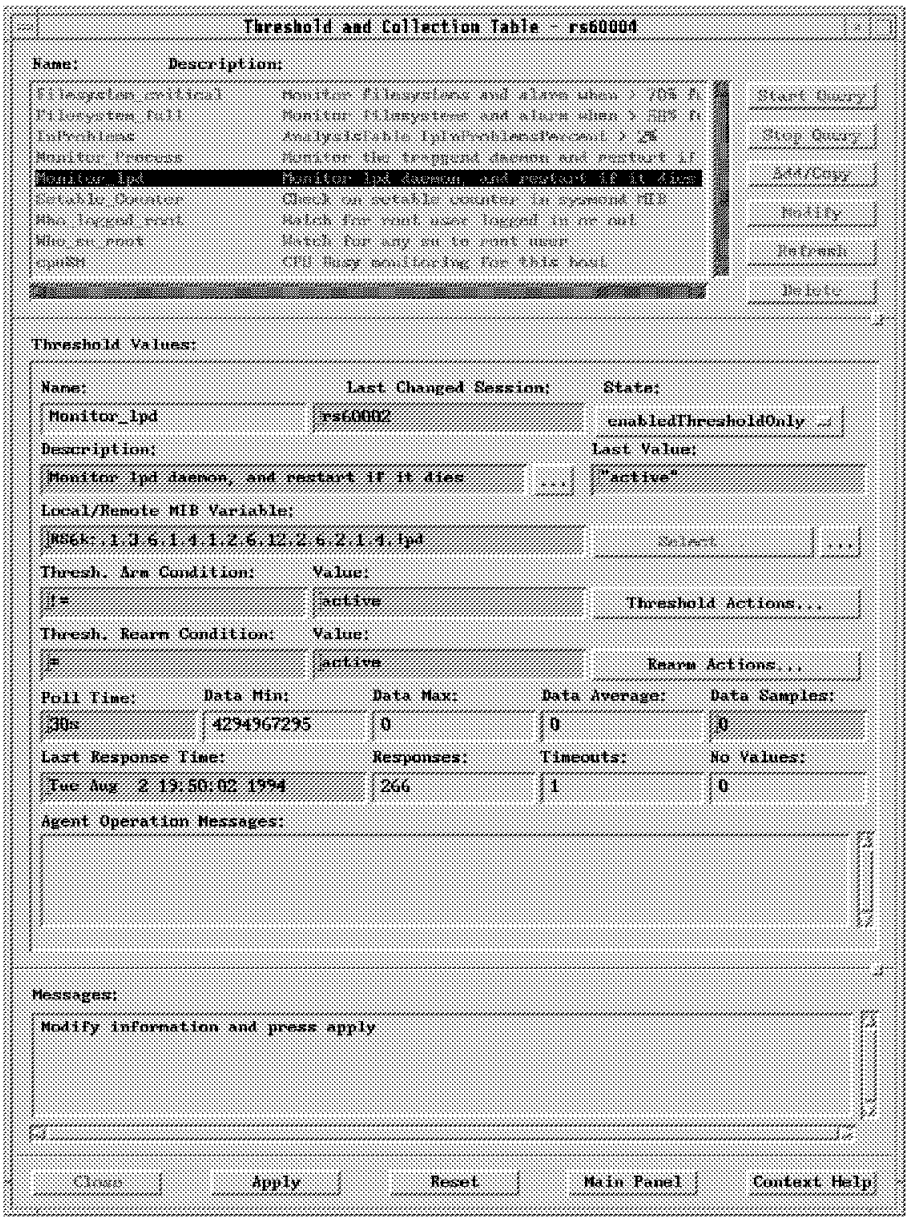


Figure 140. Threshold Table Entry for Monitoring Status of lpd Daemon

The MIB variable to be monitored is specified in the Local/Remote MIB Variable field. It is specified in three parts:

name: This is the name or address of an IP node, or a MLM Alias Table name, followed by a colon (:). If this is not specified the MLM will retrieve the MIB variable locally (that is, from the machine running the MLM). In this case we specify a list of machines by referencing the RS6k Alias Table entry.

MIB object This is the MIB object that we want to retrieve, in dotted decimal. If you don't want to type this string in you can press the Select button, which starts the NetView for AIX MIB Browser. You can then navigate to the branch of the MIB you want and cut and paste the object ID into the field. In this case the MIB object (1.3.6.1.4.1.2.6.12.2.6.2.1.4) is part of the SIA MIB extension. It contains the status, in text form, of all subsystems. The value is either active or inoperative.

.InstanceID This identifies which instance(s) of the MIB object we are interested in. A value of .* would mean all instances. In this case we have appended .lpd as the instance ID. This means that we are only interested in monitoring the value for the lpd subsystem. We could also have specified this by its ASCII code, written in dotted-decimal (for lpd that would be .108.112.100).

AIX-Specific MIB Objects

Note that the MIB object in this case shows status derived from the Subsystem Resource Controller, which is a part of AIX that provides monitoring and control of important processes. If we were doing this using, for example, an HP 9000 running Systems Monitor for HPUX we would need to use the Process Table instead of the subsystem table.

The threshold condition will be met when the result of the MIB variable above is *not equal to active*. That is, the lpd daemon is inoperative.

When the threshold condition is met, the MLM will execute the threshold actions. These are to send trap 1000 to NetView for AIX and also to execute the command `startsrc -s lpd`, to automatically restart the daemon. You can see the panel where we specified these actions in Figure 141.

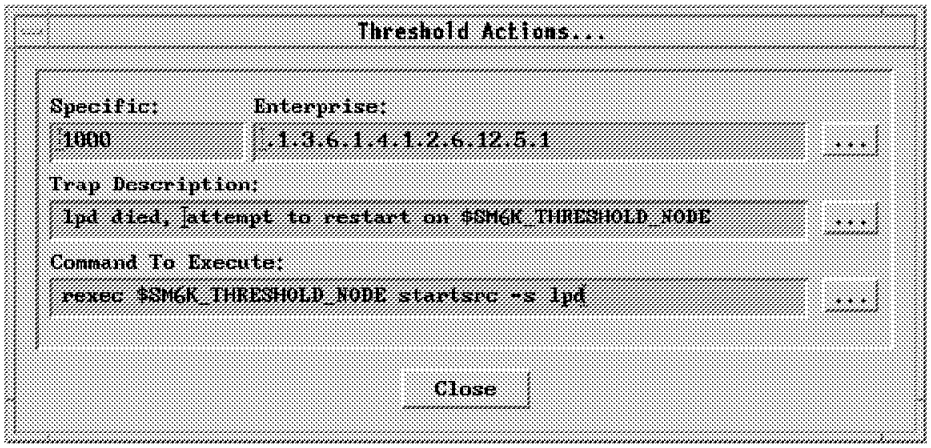


Figure 141. Threshold Actions for lpd Threshold Table Entry

These actions were entered in the Threshold Actions window, which can be displayed by clicking on the **Threshold Actions** button, seen in the main window shown above in Figure 140 on page 167.

The Enterprise field contains the enterprise id for the trap. For Systems Monitor for AIX threshold traps this value defaults to: 1.3.6.1.4.1.2.6.12.5.1.

In our configuration the Mid-Level Manager is running on node rs60004. We are using the RS6k alias, so the Threshold Table entry is monitoring the status of the daemon on *both* rs60002 and rs60004. Therefore, it is not possible to enter `startsrc -s lpd` as the command to execute since although this command would restart lpd on rs60004 successfully, but would fail to restart the daemon on rs60002. We have therefore used rexec with the variable SM6K_THRESHOLD_NODE, which will resolve to the name of the remote host on which the threshold occurred (the node on which the lpd daemon died).

This variable is one of several that can be used in the command that is to be executed, as well as in threshold and rearm trap descriptions. A complete listing of these, together with a description, can be displayed by pressing the **Context Help** button, and clicking on the Threshold Actions field when the ? prompt appears.

The most useful variables are:

SM6K_HOSTNAME The hostname of the Mid-Level Manager node (rs60004 in this scenario).

SM6K_THRESHOLD_NODE The node that caused the trap, that is the node on which the threshold occurred (rs60002 or rs60004).

SM6K_DOMAIN_NAME The qualified domain name of the Mid-Level Manager node (rs60004.itso.ral.ibm.com in this scenario).

SM6K_HOST_ADDRESS The IP address of the Mid-Level Manager node (9.24.104.27).

SM6K_THRESHOLD_VAR_VALUE The value of the variable being thresholded.

SM6K_THRESHOLD_VAR_OLDVALUE The previous value of the variable being thresholded.

SM6K_THRESHOLD_VALUE The threshold value specified in the Value field of the Threshold and Collection Table.

In order for rexec to work and restart lpd without prompting for a user ID and password, you must set up $HOME.netrc, on the Mid-Level Manager node, which contains the user ID and password to be used at the remote host. An alternative to using rexec in this case would be to use the Command Table function of Systems Monitor.

A sample entry for $HOME.netrc is shown in Figure 142. Note also that root authority will be required to restart the lpd process.

```
machine rs60002 login root password query
```

Figure 142. Sample $HOME.netrc

This file must reside in the user's home directory and only permit read-write access (that is, specify file permissions of `-rw-------`).

A rearm action was also configured to send a trap to NetView for AIX, to inform the operator when the lpd daemon comes up again. In this case trap 1001 was specified. This configuration can be seen in Figure 143 on page 170.

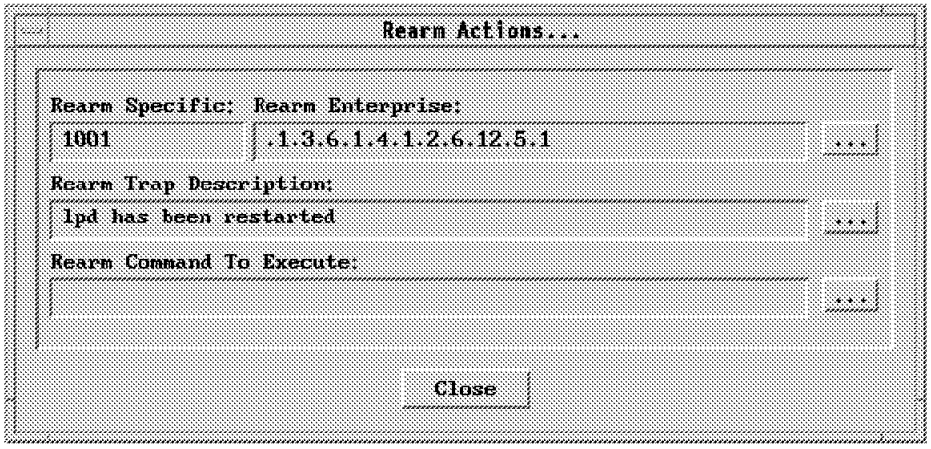


Figure 143. Rearm Actions for lpd Threshold Table Entry

So What Do We Mean by Rearm?

Very often a threshold, once it has been exceeded, will stay exceeded. In this case, we may not want to generate threshold events at every poll of the Threshold Table. The *rearm* facility allows us to avoid this. A new threshold event will not be sent until the rearm condition has been met.

Considerations if Using an SLM in Place of the MLM: In general the Threshold Table operates in the same way on the SLM as on the MLM. However, because it can only poll its own node, the equivalent Threshold Table entry for an SLM is a little simpler.

Firstly, there is no need to specify the node or alias name as a prefix to the MIB object to monitor. Therefore the SLM version of the Local/Remote MIB Variable field (see Figure 140 on page 167) would not have the RS6k: prefix.

Secondly, when defining an automatic command response to the threshold, there is no need for it to be remotely executed (because, by definition, the threshold being checked is on the same node). Therefore the rexec command shown in Figure 141 on page 168 can be simplified to `startsrc -s lpd`. This is also a more secure arrangement, because there is no need to set up the trusted host environment needed by rexec.

7.1.2.3 NetView for AIX Basic Event Configuration

NetView for AIX events can be configured by selecting **Options**, then **Event Configuration**, and then **Trap Customization: SNMP...** from the menu bar. This will bring up the Event Configuration window with a list of enterprise names and IDs. There are three entries in this table for Systems Monitor for AIX traps, as follows:

- SM/6000_Session with an enterprise ID of 1.3.6.1.4.1.2.6.12.3.1.2
- SM/6000_Threshold with an enterprise ID of 1.3.6.1.4.1.2.6.12.5.1
- SM/6000 with an enterprise ID of 1.3.6.1.4.1.2.6.12

The majority of the traps that come from Systems Monitor for AIX are generated by the Threshold Table and therefore have an enterprise ID of 1.3.6.1.4.1.2.6.12.5.1. Other types of Systems Monitor for AIX traps include

interface status traps and traps from the File Monitor Table, which are SM/6000_Session traps.

In order to configure threshold traps, we must select the **SM/6000_Threshold** entry in the table and then add the appropriate traps. If Systems Monitor for AIX traps are not configured for NetView for AIX in this way, then the trapd daemon, running on NetView for AIX, which receives the traps from the Mid-Level Manager will not understand them. If this is the case, the event message will display the following information:

```
? Trap found with no known format in trapd.conf
```

This is not particularly enlightening.

However, more meaningful messages can be displayed by configuring the traps in a number of ways:

- Display helpful messages on the event card to assist network operators with resolving problems.
- Bring up pop-up windows, to attract the attention of network operators.
- Execute a command.
- Configure the category, status and severity of the event.

Examples of Using NetView for AIX V3 (GG24-4327) contains several examples of configuring NetView for AIX events. With NetView for AIX Version 4 you have a further option for event configuration using event rulesets. These are described in depth in *Examples of Using NetView for AIX V4*, SG24-4515.

Event configuration is illustrated in Figure 144 on page 172, showing the event configuration for trap 1000, which is generated when the status of the lpd daemon changes.

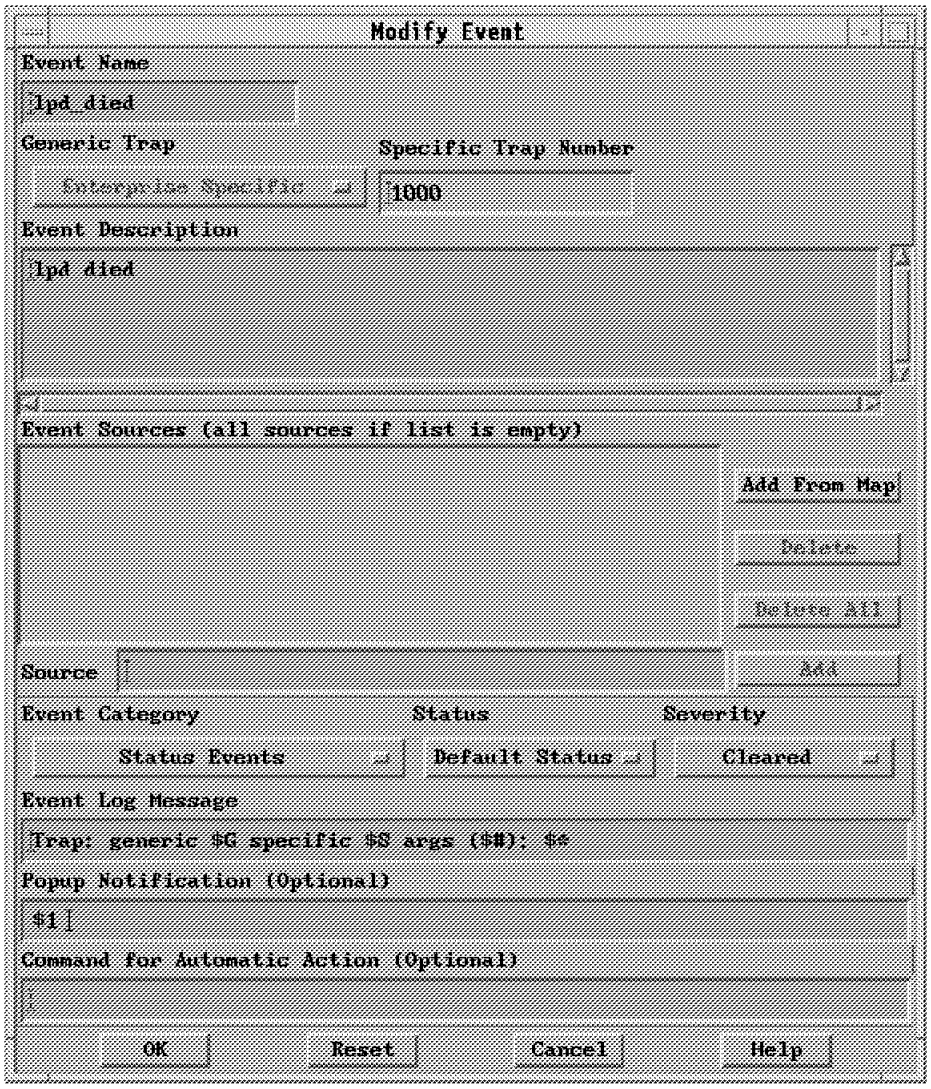


Figure 144. Event Configuration for lpd Died Trap

It can be seen in this configuration that we have left the event log message (the message that will appear on the event card) to default. This default event log message (Trap: generic $G specific $S args ($#): $*) contains a lot of information:

- $G displays the generic trap number.

  Since the lpd died event is an enterprise specific event, the generic trap number is 6.

- $S displays the specific trap number.

  The specific trap number specified in the Threshold Table for this trap is 1000.

- $# displays the number of variables in the trap.

  Every Systems Monitor for AIX threshold event has 13 variables, which are described in more detail in 7.1.2.4, “Event Card Variables” on page 174.

- $* displays all the trap variables as name-type:value strings.

The resulting event card that is generated when the lpd daemon on rs60002 dies is shown in Figure 145 on page 173.

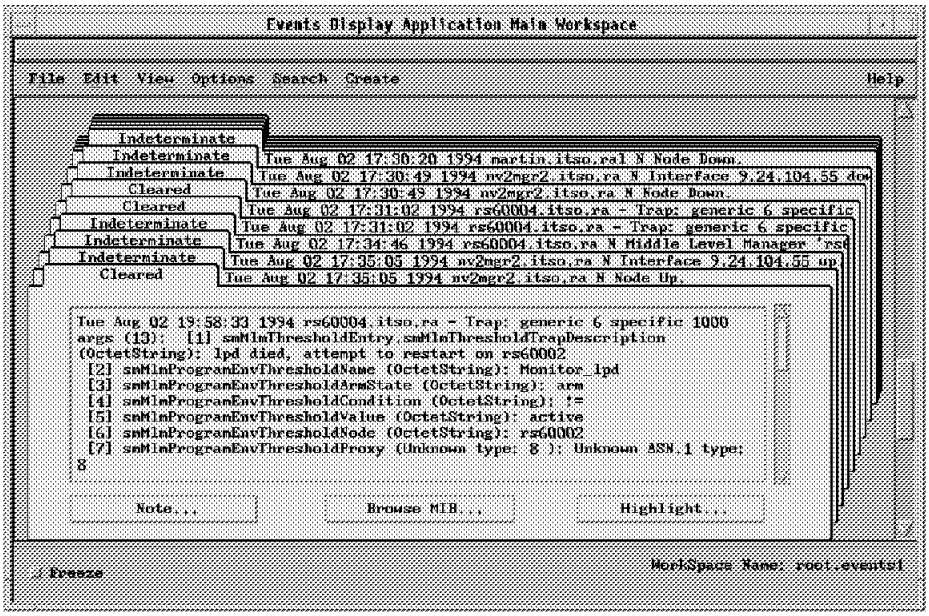


Figure 145. Event Generated When lpd Dies

Although this certainly contains a description of the event, it is probably not presented in the way our user would like to see it. We make it a little friendlier in 7.1.2.5, “Improving the Presentation of NetView for AIX Event Cards” on page 177.

In our event configuration we also defined a message to pop up (Figure 144 on page 172). The pop-up message is specified as $1, which means that the first variable within the trap is to be displayed. For MLM traps this variable is the trap description specified in the Systems Monitor for AIX Threshold Table configuration, (seen in Figure 141 on page 168). The resulting pop-up window is shown in Figure 146.

Figure 146. Pop-Up Window for lpd Died Trap

Any text can be specified in the pop-up notification field and it will appear in the pop-up window. Alternatively, as in this case, variables can be passed through from Systems Monitor. The other variables available will be discussed in more detail in 7.1.2.4, “Event Card Variables” on page 174.

The configuration for our rearm trap, 1001 (which is generated when the lpd daemon is restarted), was performed in the same way as the threshold trap.

The event card that is generated when the lpd alive trap is generated for node rs60002 is shown in Figure 147 on page 174.

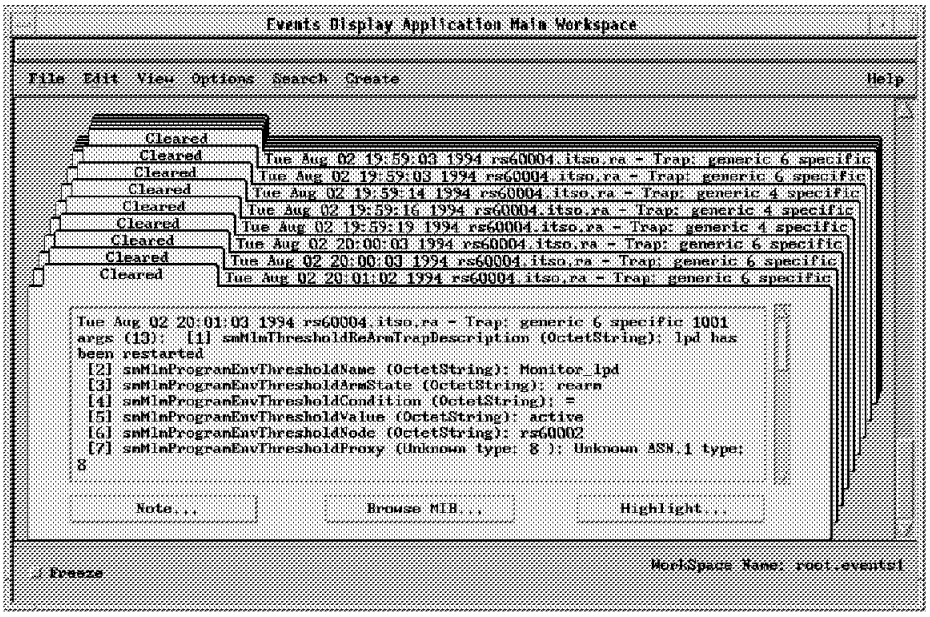


Figure 147. Event Generated When lpd Restarts

7.1.2.4 Event Card Variables

Although SNMP traps are generally simple things, they may contain a variable bindings area which is a list of MIB object/type/value triplets. Systems Monitor for AIX uses this to pass a wealth of information within the traps it sends.

We have already seen how this information may be included on event cards by using $1 to show the event description. In fact, for Systems Monitor for AIX threshold events, the trap contains a total of 13 MIB variables. If we use a value of $* in the event log field we will see all these variables in the event card. The following three figures show the effect of doing this.

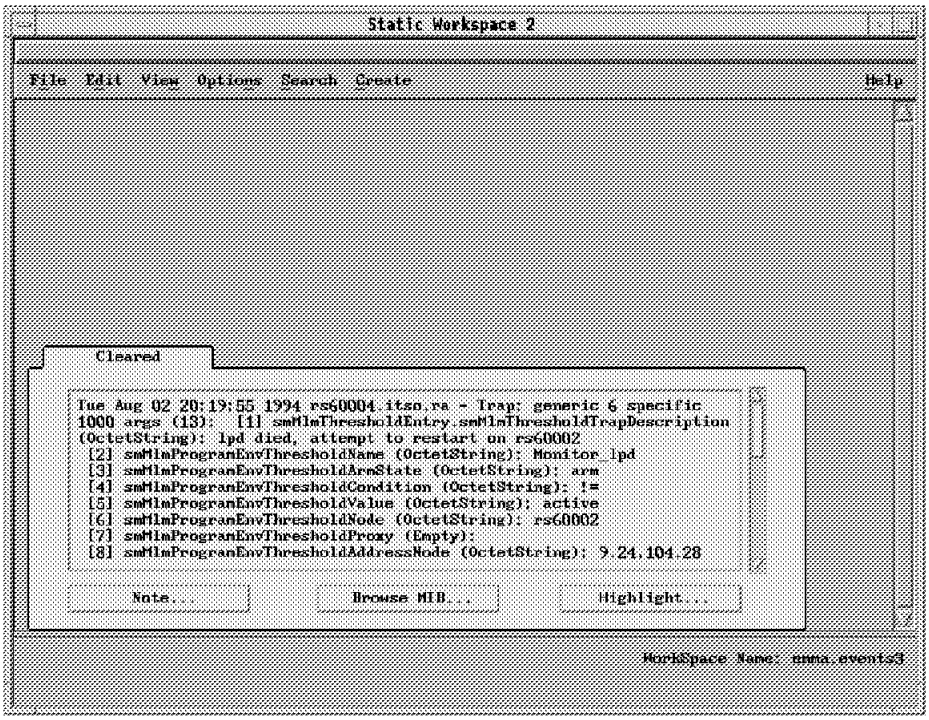


Figure 148. Detailed Systems Monitor for AIX Threshold Event Card. Event card generated when lpd dies (part 1).

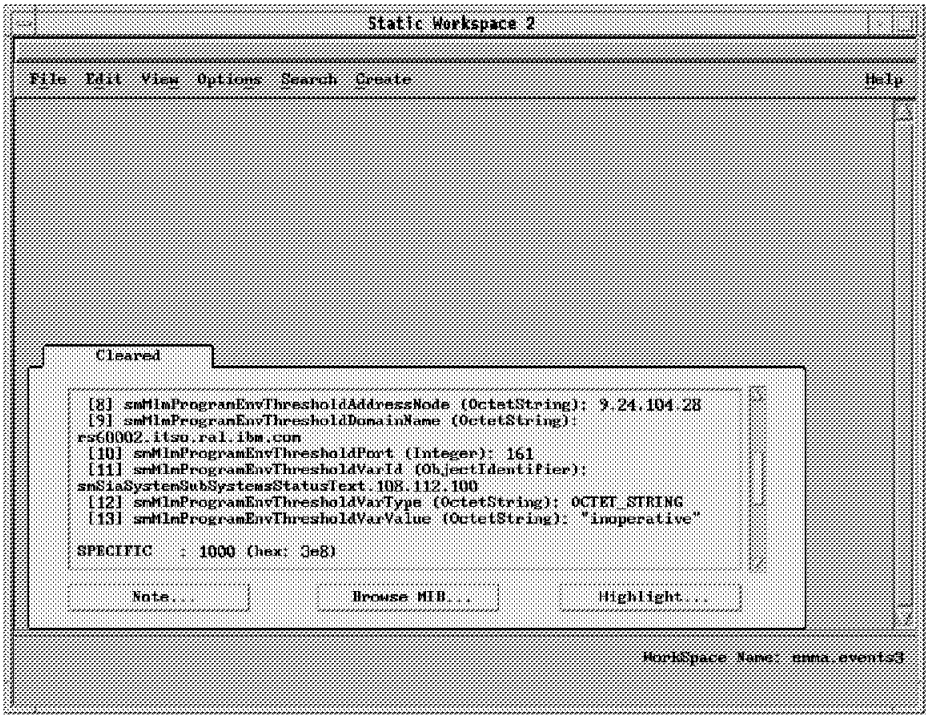


Figure 149. Detailed Systems Monitor for AIX Threshold Event Card. Event card generated when lpd dies (part 2).

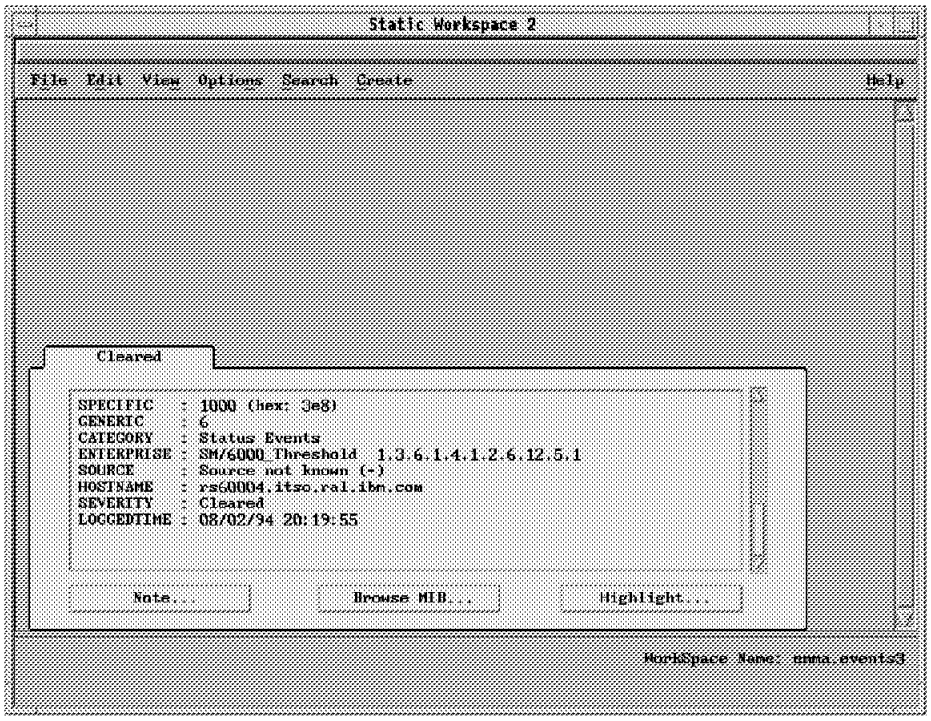


Figure 150. Detailed Systems Monitor for AIX Threshold Event Card. Event card generated when lpd dies (part 3).

A full description of the 13 variables is in *Systems Monitor for AIX Users Guide* in the MLM and SIA Reference section.

The most useful of these variables are:

- $1 = Systems Monitor for AIX trap description

  Simply specifying $1 in the Event Log or Pop-Up notification fields will cause the pop-up window to display the Systems Monitor trap description information and nothing else.

- $6, $9, $8 = Hostname/IP address of the node on which the threshold occurred ($9 is the fully-qualified hostname).

In addition to the variables from within the trap, Event Configuration give us other useful variables to use including the following:

- $A

  This corresponds to the hostname or IP address of the node that generated the trap, which for threshold traps is the MLM.

- $G

  Generic trap number.

- $S

  Specific trap number.

- \n

  This will add a new line.

- \t

  This will add a tab.

7.1.2.5 Improving the Presentation of NetView for AIX Event Cards

Event cards can be customized to display as little or as much information as required, using the variables that have just been introduced.

We replaced the default Event Log Message field that we used previously (Figure 144 on page 172) with the following text:

```
Threshold Trap received from mid level manager node $A for node $6.\n\nTrap
descr: $1 \nThreshold Name: $2 \nThreshold Arm State $3
```

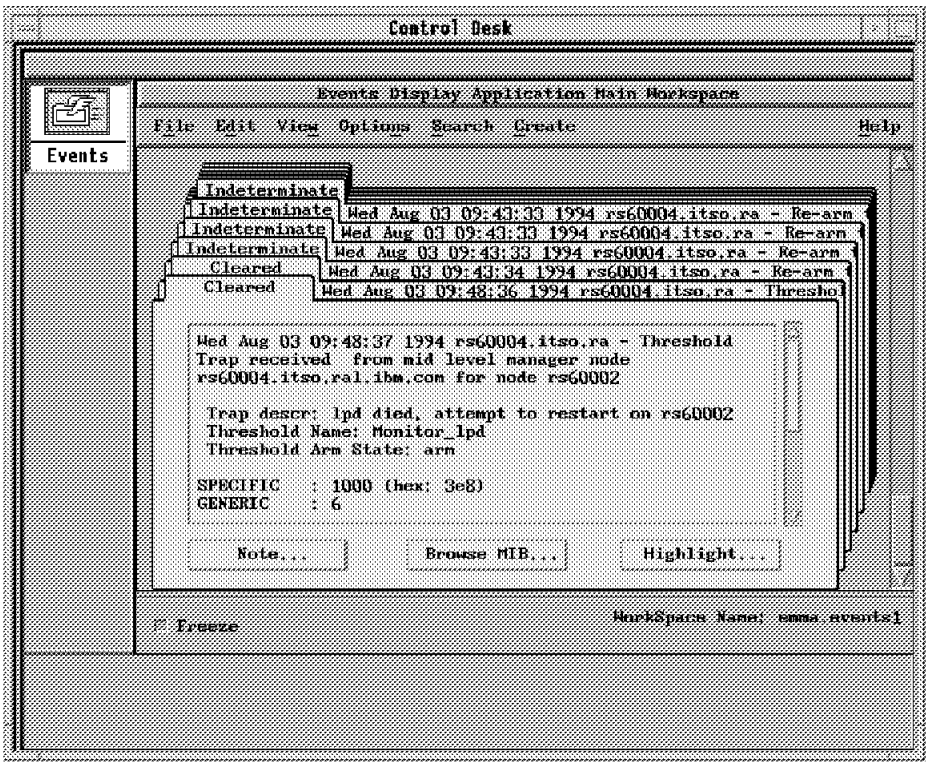


Figure 151. Customized Event Card

There are further examples of using these variables throughout this section.

7.1.2.6 Filtering Traps

In the scenario described above, every time the lpd daemon fails, a trap is sent to NetView for AIX and the Mid-Level Manager attempts to restart the daemon. You may decide that you do not want your operators to be informed every time the daemon fails (perhaps you do not want them to know at all and are quite happy for the automatic resolution process to restart the daemon).

This can be achieved by adding a filter to block all lpd traps so they are not sent to NetView for AIX. The MLM Filter Table entry to do this is shown in Figure 152 on page 178.

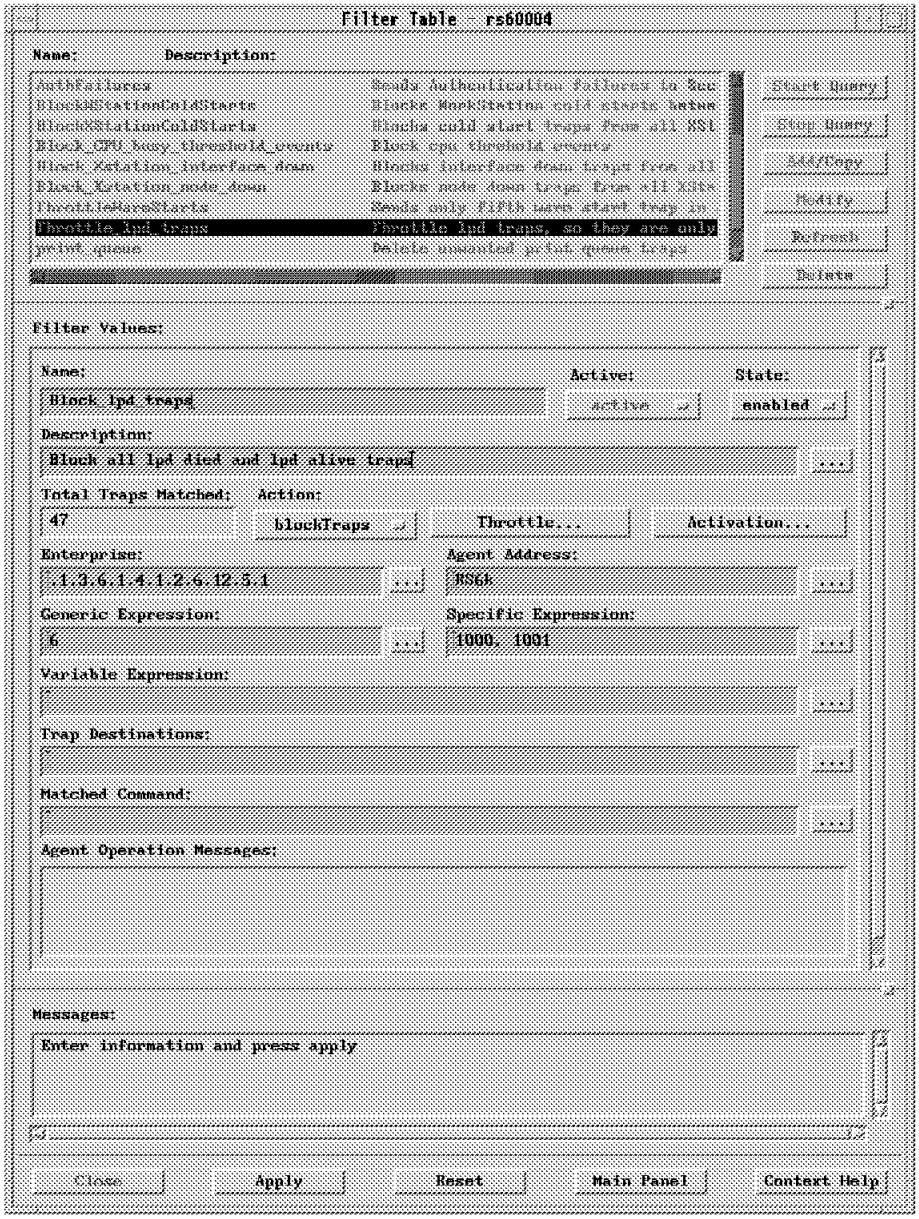


Figure 152. Filter Table Configuration to Block All lpd Traps

The Total Traps Matched field indicates the number of traps that have matched the filter and the Activation field can be used to specify the time period that the filter is active. For example, you may decide that you only want to block lpd traps at off-peak hours when less use is being made of the printers.

7.1.2.7 Throttling Traps

Blocking the traps may seem a rather drastic action. What happens, for example, if there is a serious problem with the daemon and it repeatedly fails? In this case if the Mid-Level Manager kept trying to restart it, the NetView for AIX operators would never know about the problem until users started complaining.

A compromise is to throttle the traps so that they are only sent to the NetView for AIX system if, in the example shown below, the lpd daemon fails more than three times in 30 minutes.

The filter we used to achieve this is shown in detail in Figure 153 on page 179.

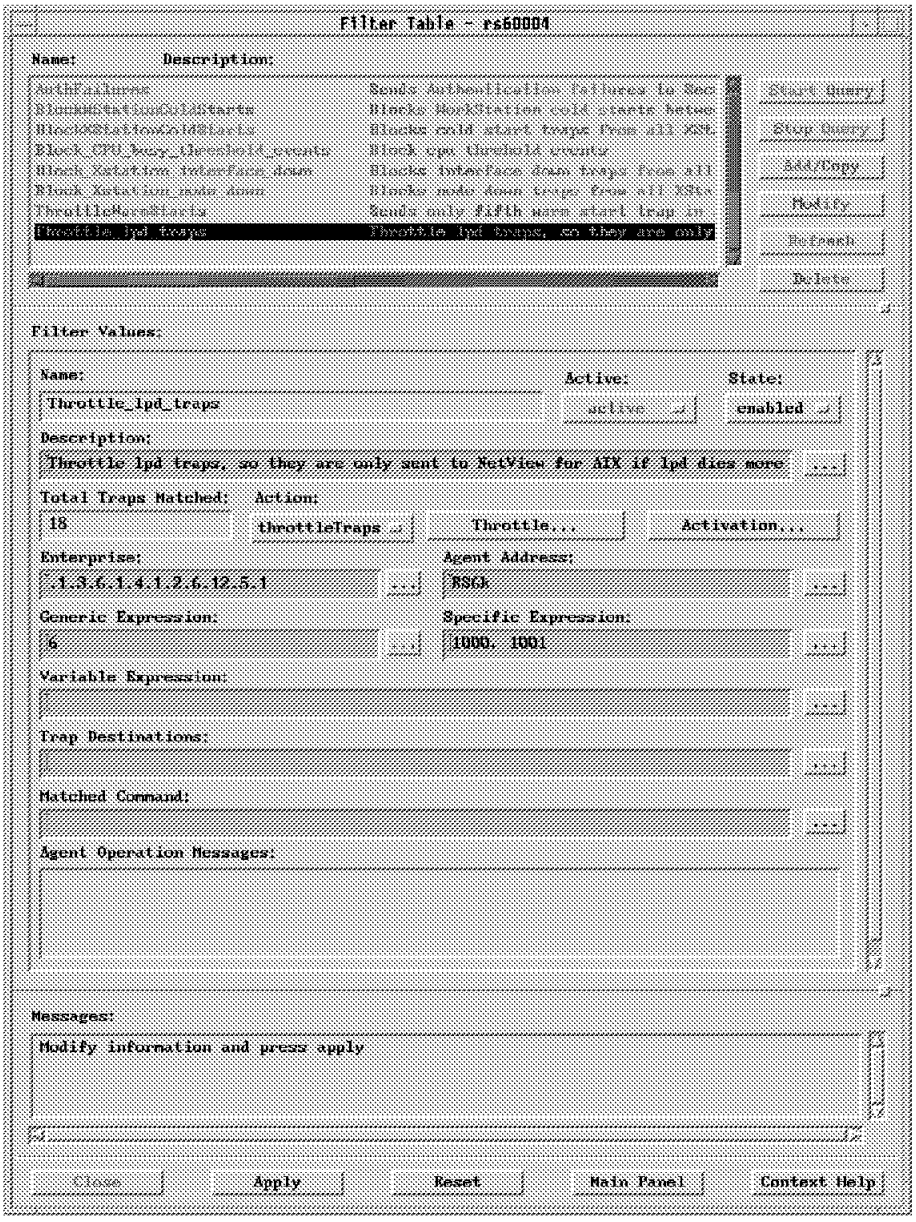


Figure 153. Filter Table Configuration to Throttle lpd Traps

In this example the filter has been configured to throttle, rather than block, traps. The throttle configuration is shown in Figure 154 on page 180.

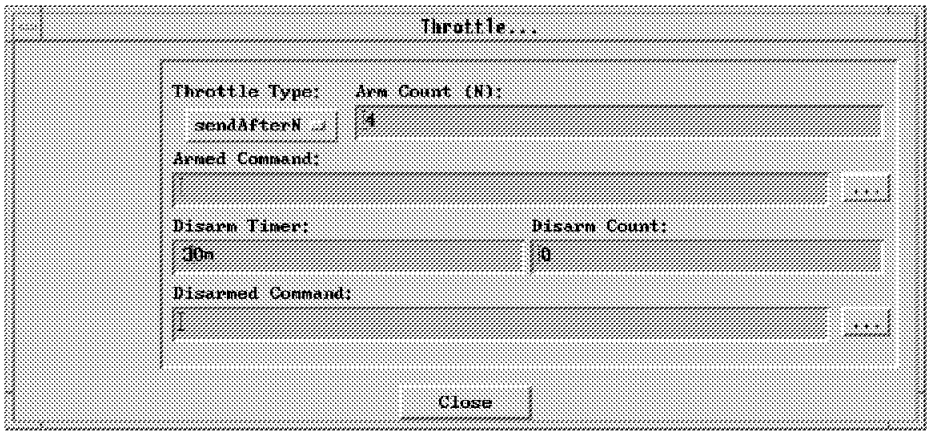


Figure 154. Throttle Configuration for lpd Traps

You can see that the Disarm Timer is set to 30 minutes. This timer specifies the time period to elapse before the throttle is disarmed. The time period is the time since the throttle is started. That is, since the first trap is matched *not* the time that the throttle is activated.

So in our scenario the throttle will be started as soon as the MLM receives the first lpd died trap. The throttle will remain armed for 30 minutes, after which it will reset. Therefore traps will only be sent to NetView for AIX if three further traps are generated in that 30-minute time frame.

However, care must be taken when applying filters to ensure that the traps you want to be sent to the NetView for AIX node really get there. For example, with this filter, if the lpd daemon dies and *stays* dead, no trap will be sent to the NetView for AIX operator because we set a rearm condition. This causes no traps to be sent at all until a poll detects that the lpd daemon is back up. We should remove the rearm specification to be sure that this throttle works correctly (Figure 144 on page 172).

An alternative approach is to set the throttle to sendFirstN and set N to 1. This will cause the first lpd trap that is generated to be sent to NetView for AIX, but subsequent traps will match the filter and will not be sent.

Considerations if Using an SLM in Place of the MLM: The SLM filter table works identically on the MLM, except that it only handles traps originating on its own node. The same filter definitions could be employed and the same caution with respect to use of filters in combination with rearm applies.

7.1.3 Monitoring Status of Print Queues

Simply monitoring the status of the lpd daemon does not provide complete printer management. The status of the printer queues (that is whether they are up or down) and the number of jobs in the queue can also impact the performance of the printers in terms of the time it takes for output to appear in the printer room.

If a printer queue goes inactive the printer administrator will want to know about it to allow him to try and fix the problem as quickly as possible; ideally before any users notice.

This can be done by running a shell script on a regular basis to search the output of the lpstat command for any occurrences of the word DOWN. The script will then print the names of the queues that are down, as well as copying them to a file called /tmp/.qmem. The script we wrote to do this can be seen in Figure 155 on page 181.

```
#!/bin/ksh -f

QUEUES=`lpstat | awk 'BEGIN {}
/DOWN/ { queue[$1]="down" }
       { }
END {
 for(i in queue)
   print i
}'`

if [ -z "$QUEUES" ]
then
    cat /dev/null > /tmp/.qmem
else
   echo $QUEUES
   echo $QUEUES > /tmp/.qmem
fi
```

Figure 155. Shell Script for Monitoring Print Queue Status, monitorq2

If we want to use Systems Monitor to automatically monitor print queue status using this script, we have to configure the following:

- Command Table (SIA)
- Threshold Table (MLM)
- NetView for AIX Event Configuration

  (We will not be discuss this in detail here, as it was covered in the last section.)

7.1.3.1 Command Table Entry to Monitor Print Queue Status

We added an entry to the Systems Monitor for AIX Command Table to run our shell script so that the output can be found in the MIB variable CommandTableDisplayStringResult.prt_queue. Because the output of the command is a text string and the instance ID is the name of the Command Table entry (in our case prt_queue). The Command Table entry can be seen in Figure 156 on page 182. You can see that the shell script is called monitorq2 and is stored in the /tmp directory.

Note

Make sure you test the shell script before creating the Command Table entry. In particular, ensure that the monitorq2 shell script has executable permission.

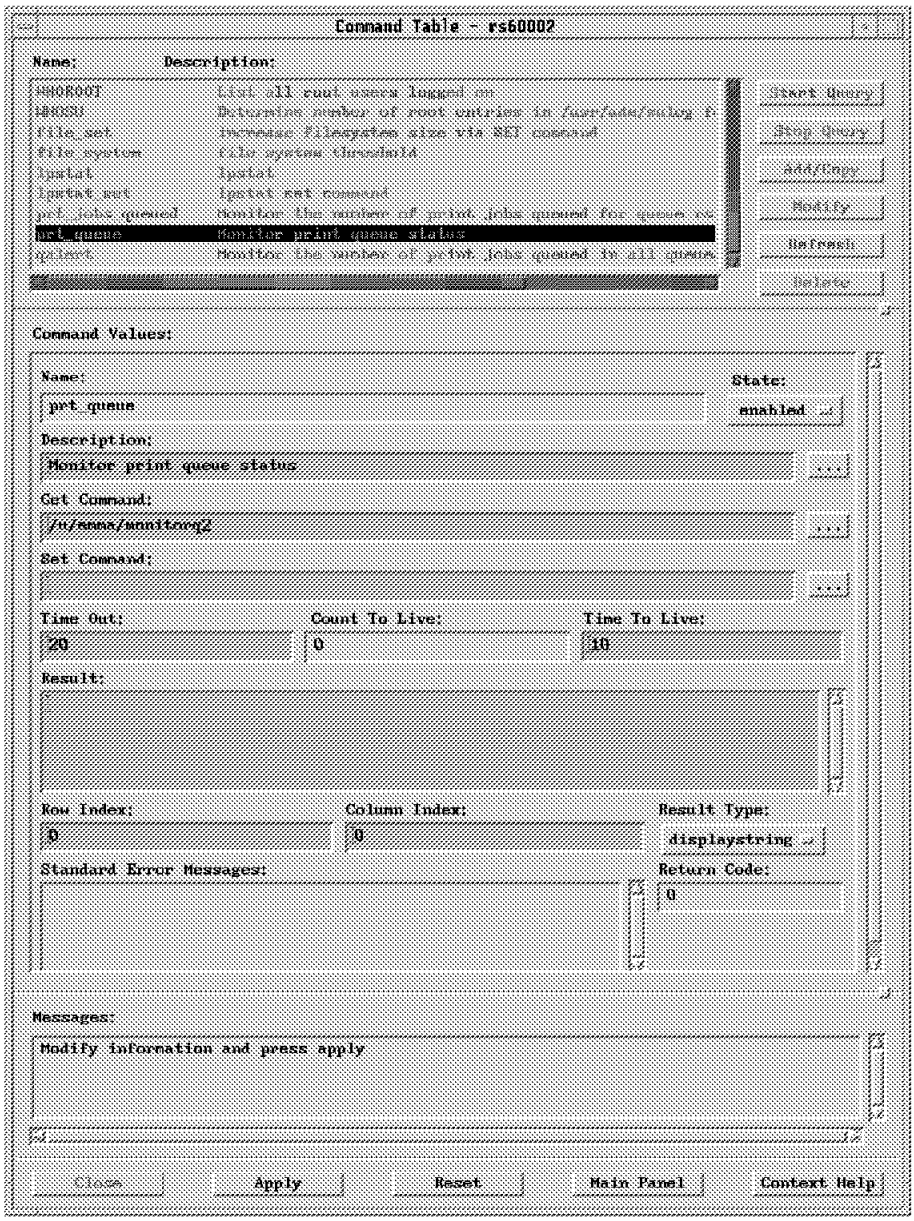


Figure 156. Command Table Configuration to Monitor Print Queue Status

7.1.3.2 Threshold Table Entry to Monitor Print Queue Status

The Threshold Table can be used to poll the Command Table entry we have just created, thereby testing the print queue regularly. It can send a trap to NetView for AIX if the command output is not equal to NULL; that is, if one or more queues are down. The Threshold Table entry can be seen in Figure 157 on page 183.

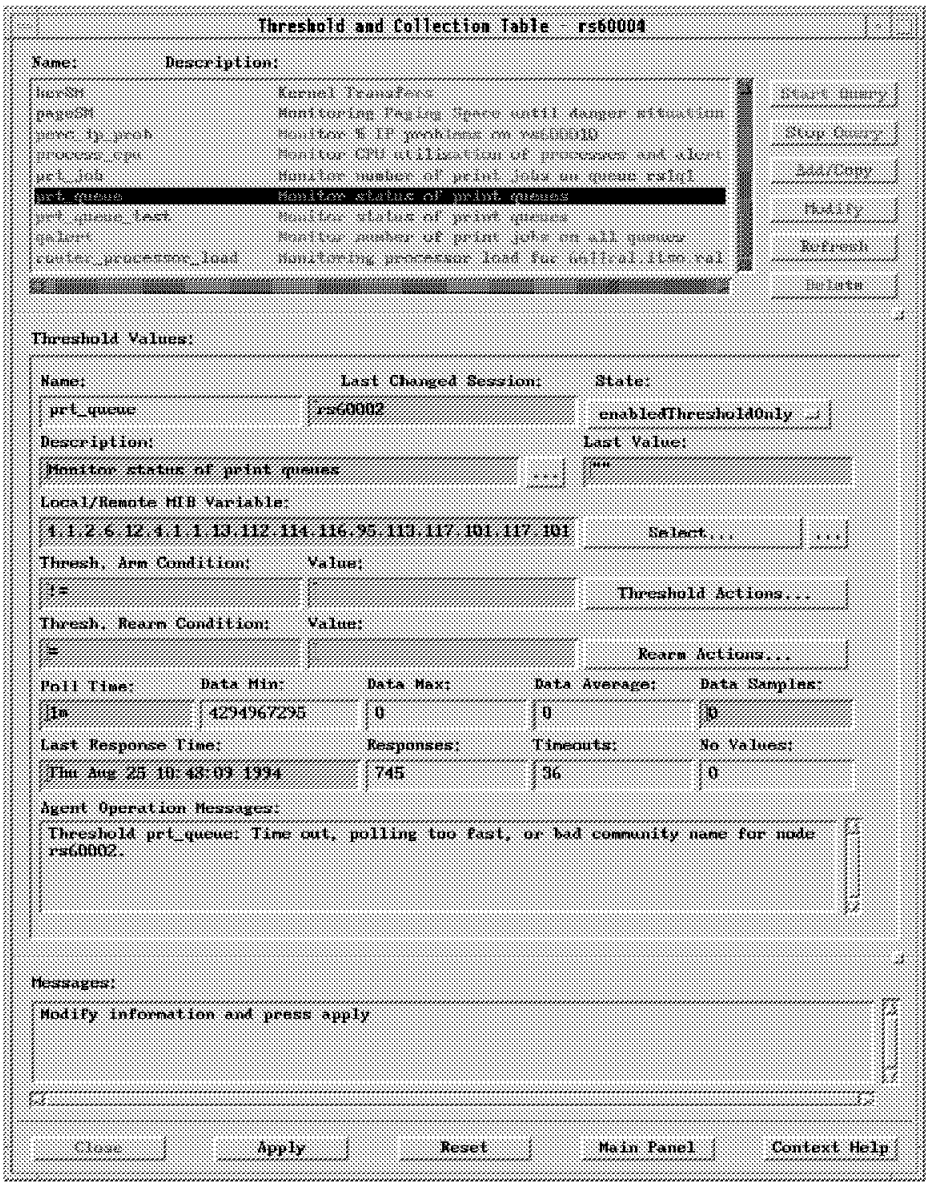


Figure 157. Threshold Table Configuration to Monitor Print Queue Status

The output from the Command Table is a string and therefore the MIB variable that we want to threshold on is the one that contains the displaystring results for the Command Table. This MIB variable has an object ID of:

.1.3.6.1.4.1.2.6.12.4.1.1.13

As we only want the result of the Command Table entry called prt_queue we must append this to the object ID, using one of the following options:

1. By appending the name of the Command Table entry, (prt_queue) to the MIB object ID, that is: .1.3.6.1.4.1.2.6.12.4.1.1.13.prt_queue.
2. By appending the dotted decimal form of the instance ID (the table entry name as ASCII characters). This is the version that we used in Figure 157. We used the MIB Browser to find out what the dotted decimal translation was.

The MIB variable is prefixed with the name of the node to be monitored, which in this scenario is rs60002.

See page 167 for further explanation of how to specify the Local/Remote MIB Variable field.

We configured the Threshold Actions as shown below. When the threshold condition is met, trap 100 is sent to NetView for AIX with the trap description shown.

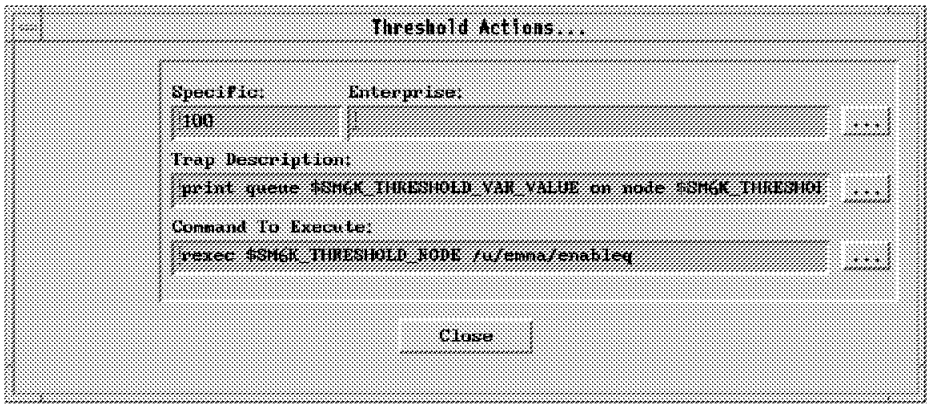


Figure 158. Threshold Actions for Print Queue Status

As you can see, we used two Systems Monitor for AIX variables in this configuration.

- $SM6K_THRESHOLD_VAR_VALUE is the value of the variable being thresholded, that is the name of the queue or queues that are down.
- $SM6K_THRESHOLD_NODE is the name of the node on which the queues are located.

We also specified a command in the Command To Execute field. This will cause a shell script called enableq to run on the node where the queues are located. The script looks at /tmp/.qmem for a list of queues that are down and attempts to restart them, thereby automating the process as much as possible and reducing the requirement for operator action (/tmp/.qmem was written by our monitoring script).

The enableq script is listed below:

```
#!/bin/ksh -f

QUEUES=`cat /tmp/.qmem`

for i in $QUEUES
do
  qadm -U $i
done
cat /dev/null > /tmp/.qmem
```

Figure 159. Shell Script to Automatically Enable Queues

Be Careful

Make sure that the necessary files and scripts are located on the correct nodes. The shell scripts enableq and monitorq2 must reside on the remote node. That is, the node where the print queues are located (rs60002 in this scenario) and not the Mid-Level Manager.

The trap that is sent to NetView for AIX is shown in Figure 160.

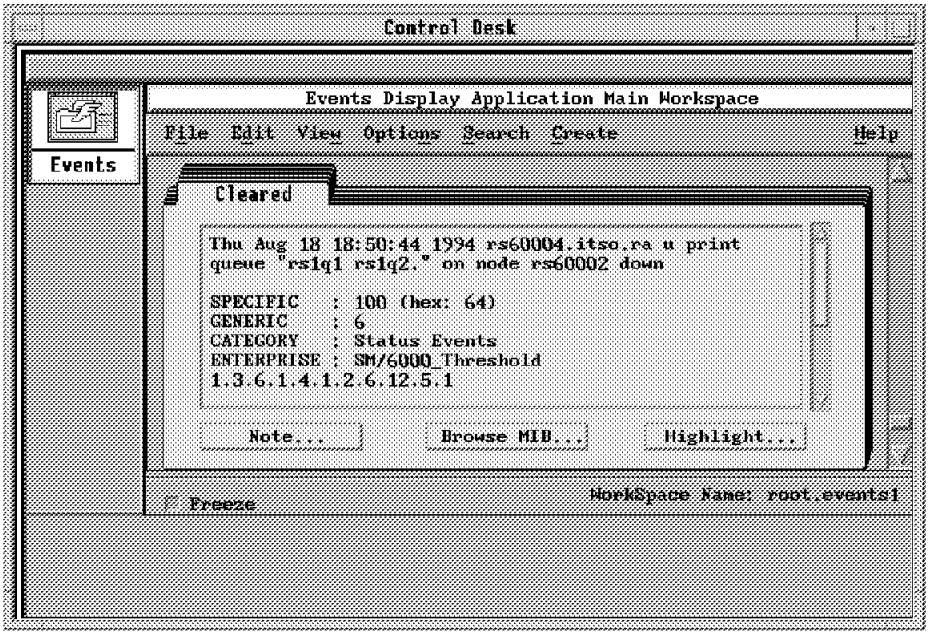


Figure 160. Print Queue Status Event

In this threshold configuration a rearm condition has been set, which will rearm the threshold condition when the output from the command table is NULL (when all the queues are back up again). A rearm trap could also be configured if required, although in this example we decided not to set one.

Considerations if Using an SLM in Place of the MLM: This example would operate in the same way if an SLM was used instead of an MLM, except:

- The MIB variable to be monitored would not be prefixed with the node name (rs60002), because the SLM would have to be running on rs60002 (it can only monitor the node it is on).
- There would be no need to use rexec to execute the enableq script.

7.1.3.3 To Set a Rearm Condition or Not?

In the previous two examples of monitoring the lpd daemon and print queue status, a rearm condition was specified. This means that when the lpd daemon dies, or a print queue goes down, a threshold trap will be generated because the threshold condition has been met. However, another threshold trap will not be sent until the rearm condition has been met, since this has the effect of rearming the threshold condition.

It is not necessary to set a rearm condition, unless you want one. If no rearm is specified, a threshold trap will be sent at every poll that the threshold is met. We have already described one case in which it was preferable not to set a

rearm condition (7.1.2.7, “Throttling Traps” on page 178). In general, a rearm action helps prevent the operator being overloaded with traps. For example, if we monitor file system utilization and specify a threshold condition of >80, a trap will be sent every time the file system utilization exceeds 80%. If the high utilization persists, you will end up with traps coming in at the same frequency as the poll interval (unless a rearm is set).

As before, an alternative way to handle this situation would be to use the throttling capability of the Filter Table, but only if the rearm action is disabled. The lesson to learn is that these two functions should be used together only with great care.

7.1.4 Monitoring Number of Jobs in the Queue

We have now built monitors and automation to handle failure of lpd and of a print queue. However, the number of jobs that are waiting in a queue can impact performance and therefore it would be useful to know when this number has risen beyond an acceptable level. At this point, the system administrator could intervene and may be able to distribute the jobs among other printers and queues, thereby avoiding potential problems.

There are two examples included here, the first showing how to configure the Command and Threshold Tables to monitor specific queues, the second showing how all queues can be monitored with only one Threshold Table entry.

7.1.4.1 Command Table Configuration

This configuration shows the command required to count the number of jobs in a queue called rs1q1 and is shown in Figure 161 on page 187 below:

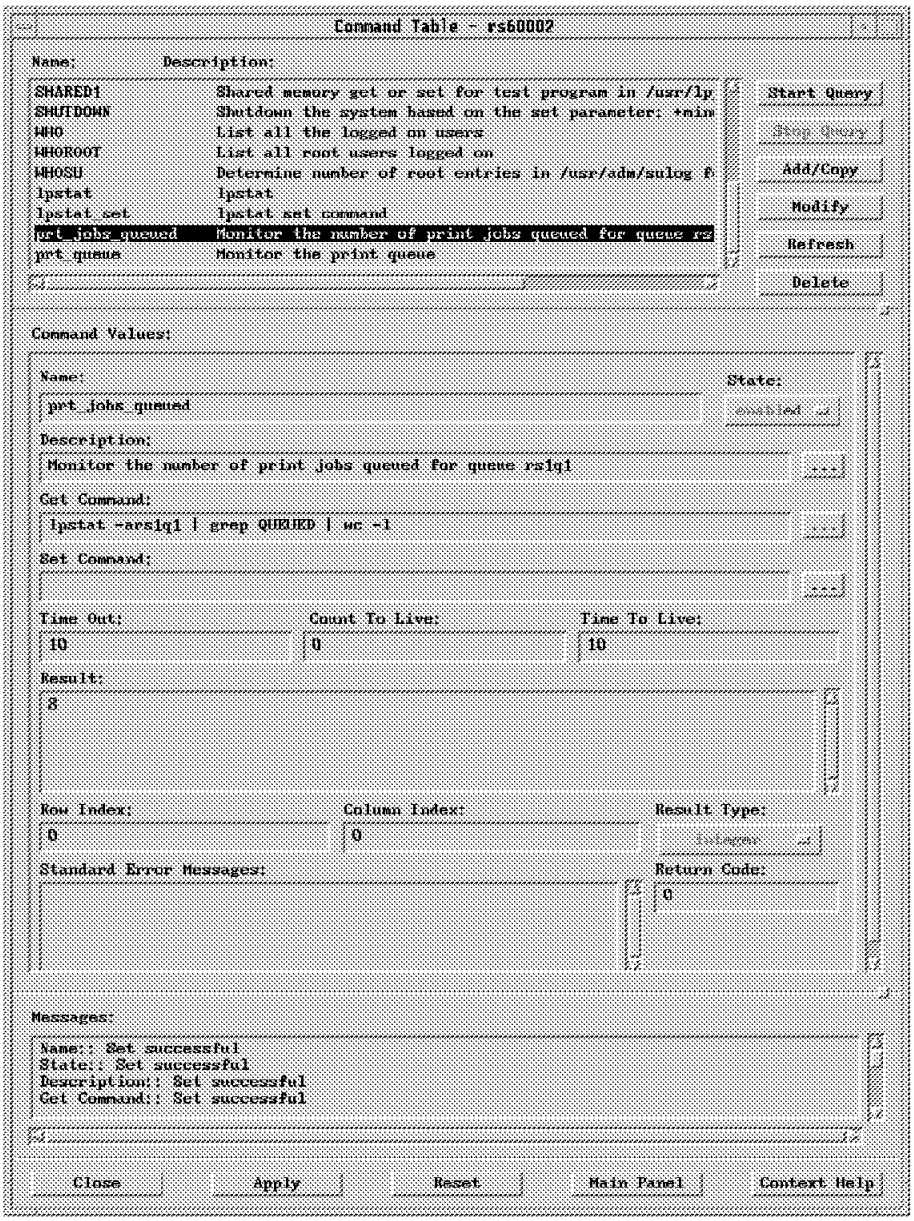


Figure 161. Command Table Entry for Number of Print Jobs Queued

It is necessary to specify the queue, otherwise the command will simply count up the total number of queued jobs. This is not very helpful, as we want to know when, for example, there are eight jobs in *one* queue and not when there are eight jobs in eight queues.

An SNMP GET will monitor the output of the `lpstat -ars1q1` command, for all instances of the word QUEUED and count the number of instances that it finds. Therefore the result is an integer and the result type field is set to integer.

Note

It is essential to check that this field is set correctly, otherwise the command will fail and the next stage will fail to work correctly.

7.1.4.2 Threshold Table Entry

We can use this table to poll the MIB variable that contains the result of the command that we have just added to the command table. The object ID for the Command Table integer result is:

.1.3.6.1.4.1.2.6.12.4.1.1.14

We are only interested in the integer result from the command called prt_jobs_queued. We therefore specify this as the instance ID, using the dotted-decimal form (Figure 162). See page 167 for further explanation of how to specify the Local/Remote MIB Variable field.

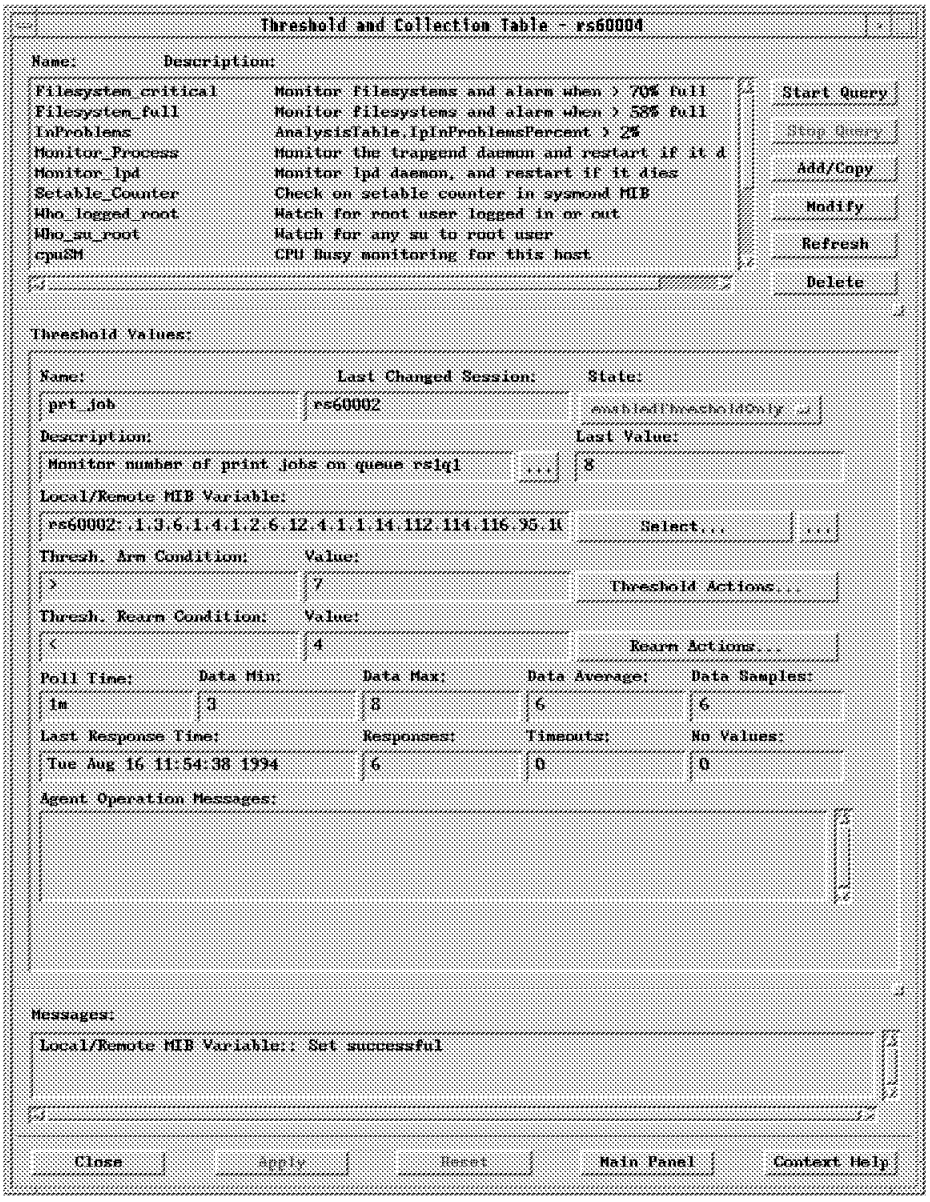


Figure 162. Threshold Table Entry for Print Jobs Queued

This entry will poll the MIB variable that contains the output from the Command Table entry added previously. If the variable has a value greater than 7 (rs1q1 has 8 or more jobs waiting), then the threshold condition will be met and a trap

will be sent to NetView for AIX. The configuration for this trap can be seen in Figure 163 on page 189.

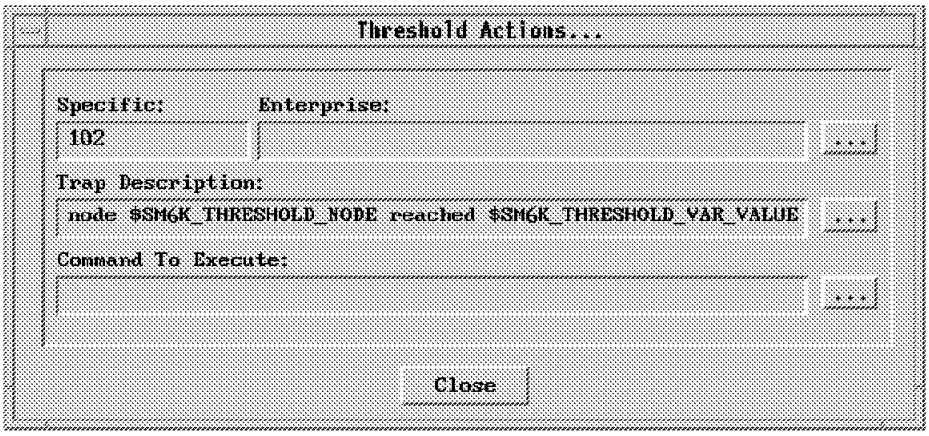


Figure 163. Threshold Table Actions for Print Jobs Queued

The trap description utilizes some of the variables described in page 169 and the complete description is as follows:

```
Number of print jobs on queue rs1q1 on node $SM6K_THRESHOLD_NODE reached
$SM6K_THRESHOLD_VAR_VALUE
```

When the number of jobs in the queue falls below 4, the rearm condition will be met and a rearm trap will be sent to NetView for AIX. The configuration can be seen in Figure 164.

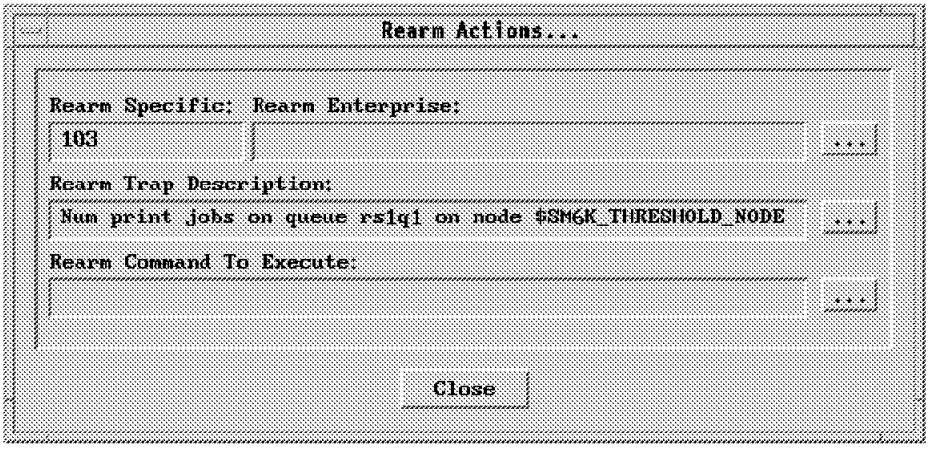


Figure 164. Threshold Table Rearm Actions for Print Jobs Queued

The threshold and rearm traps that are generated are shown in Figure 165 on page 190 and Figure 166 on page 190.

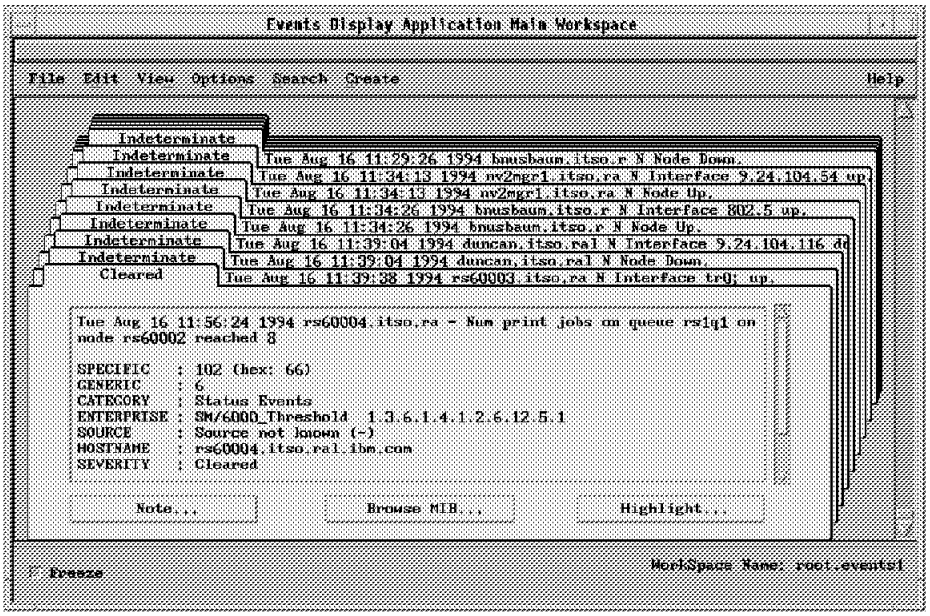


Figure 165. Threshold Event for Print Jobs Queued

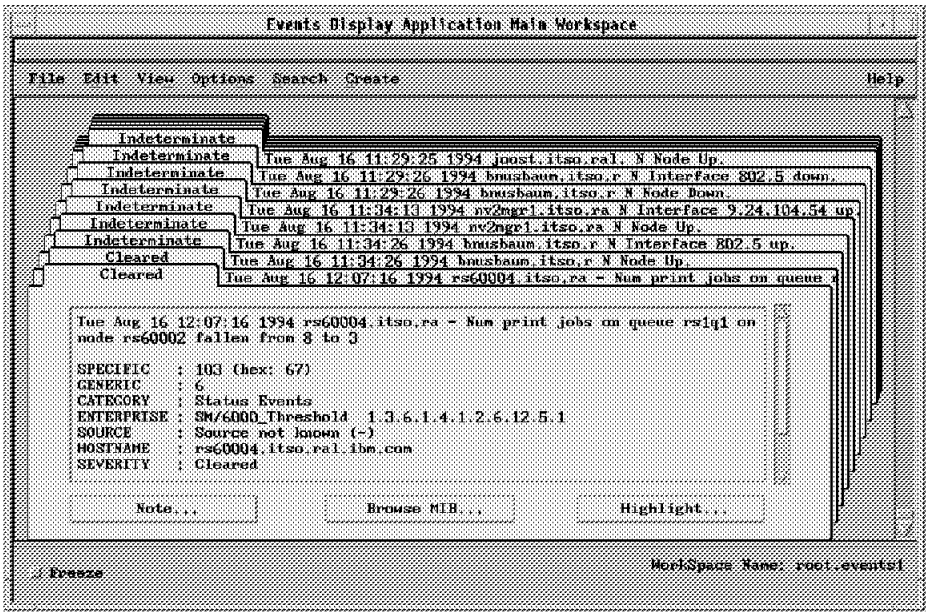


Figure 166. Rearm Event for Print Jobs Queued

Of course, we had to add entries into NetView for AIX event configuration to make the traps appear as neat as this.

This example has shown how to monitor the number of jobs in a specific queue and alert the operator by means of an event if too many are queued. However, to handle *all* queues, it would be necessary to add an entry in the Threshold Table for every queue.

A smarter way to monitor the number of jobs in a queue is to use the shell script shown in Figure 167 on page 191.

```
#

lpstat | awk 'BEGIN { currq="none"; }
$1~"QUEUED" { count[currq]++; next; }
            { currq=$1;  }
END {
  for(i in count)
    if(count[i]>7)
      print i
}'
```

Figure 167. Shell Script to Monitor Number of Jobs in Queue

This script will monitor the output of the lpstat command and count up the number of queued jobs there are for each queue. If this number exceeds 7, then the script will print the name of the queue.

An entry is then added to the Command Table to run the script, but now the output of the command is a queue name rather than a number; therefore the result type field must be set to displaystring, as shown in Figure 168 on page 192.

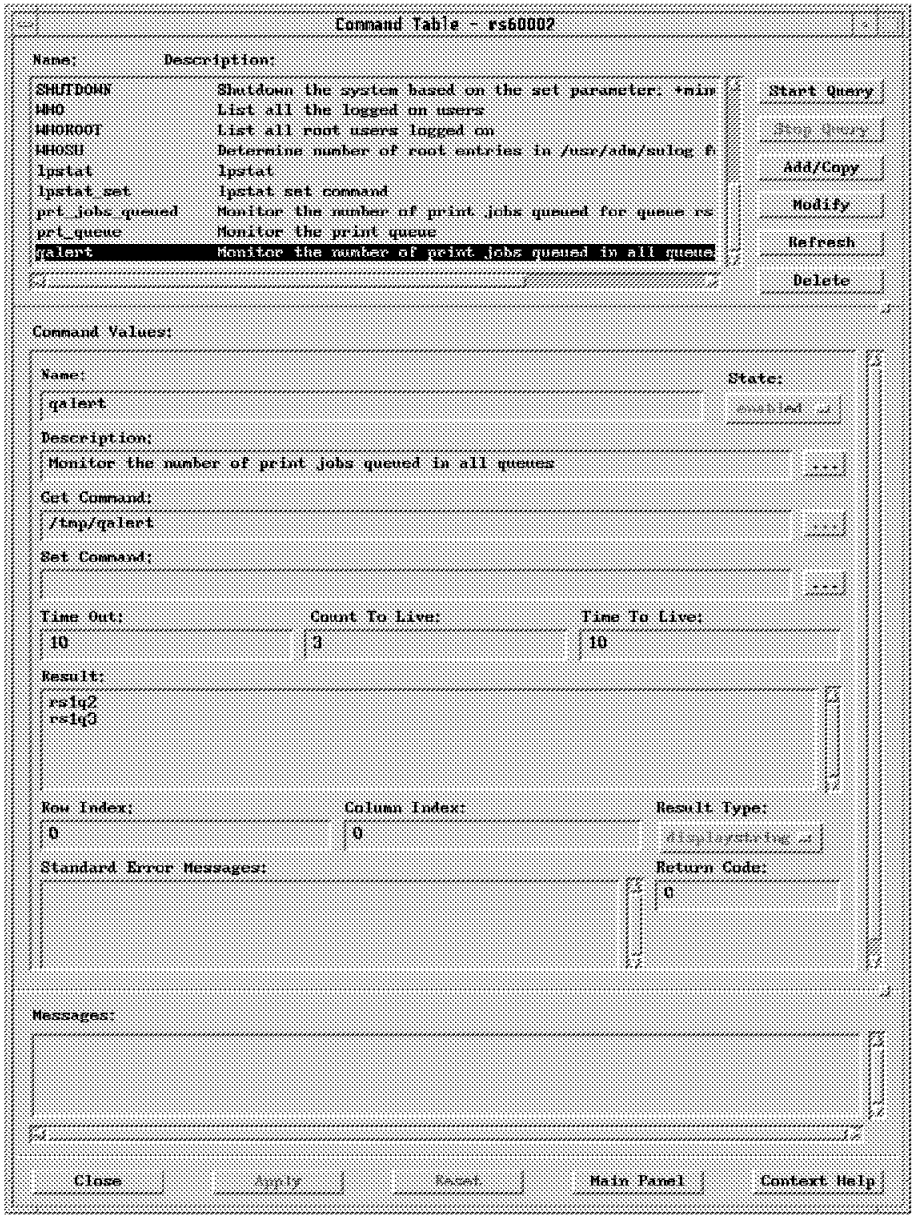


Figure 168. Command Table Entry for Qalert Script

We created a Threshold Table entry to poll this. If there is no output, it indicates that there are no queues with more than seven jobs waiting to be run. Therefore the threshold condition is configured to be exists, so that the threshold will be armed and a trap will be sent if there is output from the shell script. We set the rearm condition to changes so that we do not see a lot of duplicate events. Notice that this may delay a problem notification by one poll period as follows:

1. Queues A and B have more than 7 jobs queued - We get an event telling us about them.
2. Time goes by - We get no further events (because the command result has not changed) until...
3. Queue C now also has 8 jobs queued (it's a bad day for printing). This causes the threshold to rearm.

4. At the end of the next polling period, we get an event telling us about all three queues.

Figure 169 shows the Threshold Table entry:

Figure 169. Threshold Table Entry for Querying Output from Qalert Script

We have to define the Threshold entry to send a trap and configure that trap in NetView for AIX, as we have previously described. The event, that will then be generated when two queues (rs1q2 and rs1q3) each have more than seven jobs waiting, is shown below:

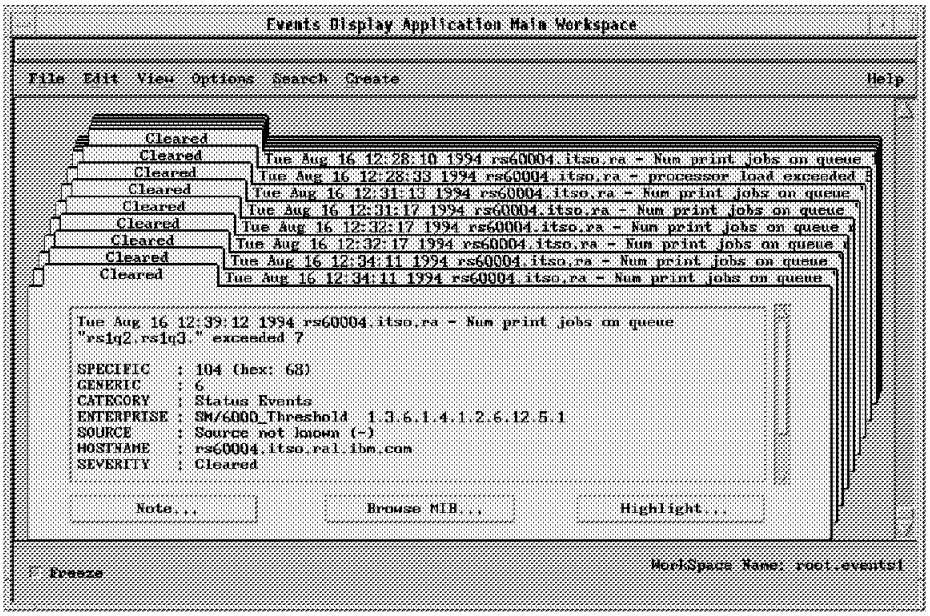


Figure 170. Event Generated When Two Queues Meet a Threshold Condition

Considerations if Using an SLM in Place of the MLM: This example would operate in the same way if an SLM was used instead of an MLM, except that the MIB variable to be monitored would not be prefixed with the node name (rs60002), because the SLM would have to be running on rs60002 (it can only monitor the node it is on).

7.1.5 Alternative Trap Destination

You may decide that you would like all the traps that relate to printing to be sent to the operators in charge of running the printers. We can easily achieve this by specifying a filter to match all the printing traps and then specify an alternative trap destination, as shown in Figure 171 on page 195. An activation time and deactivation time can also be configured for this filter, so that it will only apply on specified days and at particular times of the day.

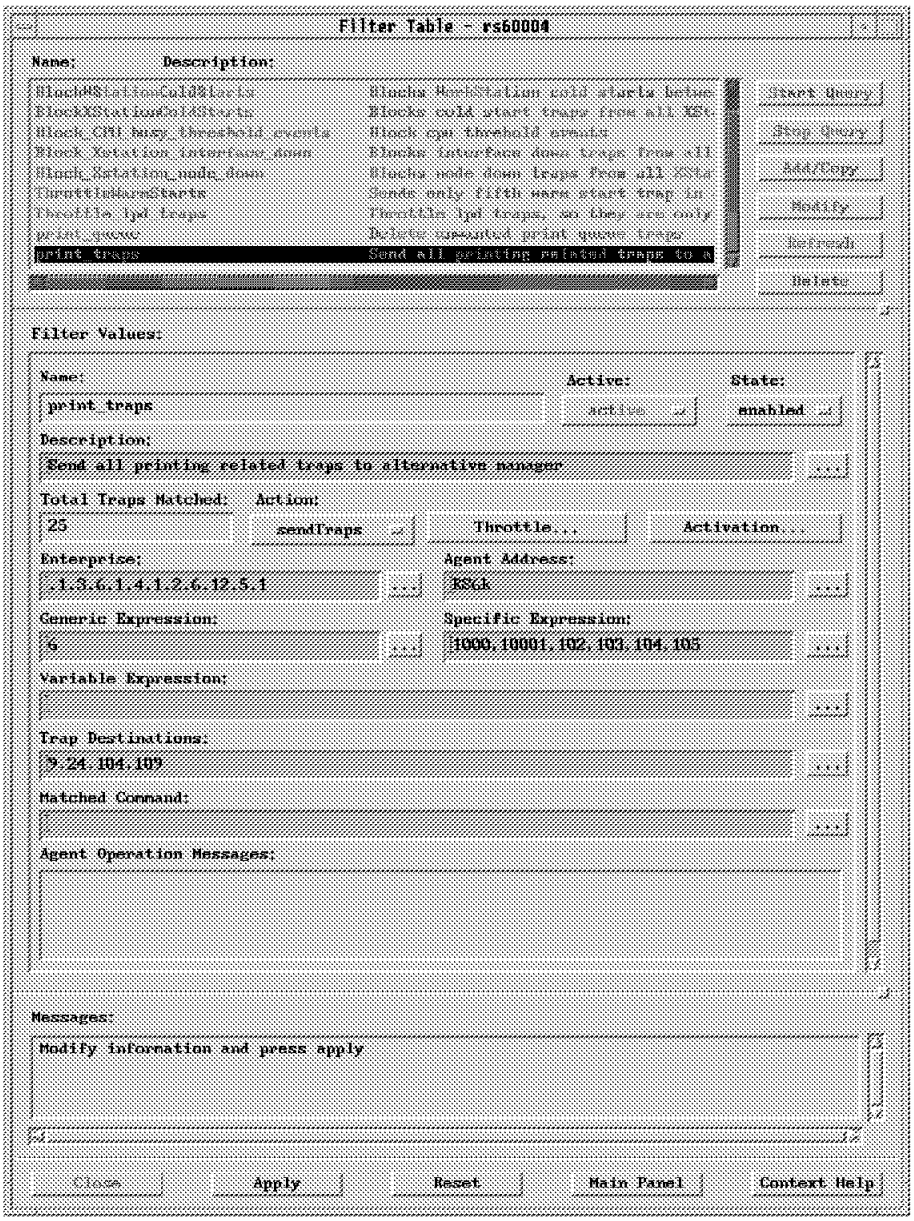


Figure 171. Alternative Trap Destination

7.1.6 Monitoring /etc/hosts for Data Changes

In any IP network with more than a handful of nodes, there is a need to use node names, in place of IP addresses. For smaller networks, static name resolution may be used, in which case every node on the network will have a file containing a list of hostnames and their corresponding IP addresses. The host uses this list to do its own name resolution. On the RISC System/6000 this file is called /etc/hosts. As administrators of the system we would like to keep these different copies of /etc/hosts identical.

In the following example, we show how to monitor changes in the actual data content of the file /etc/hosts. The File Monitor Table entry can be seen in Figure 172 on page 196 and it can be seen that the monitor type is set to *dataChange*. Once again, the entry is configured to run a command when the monitor type is met, that is, when the data in the file is changed. This could be

any command, but we have chosen to send a message to root user on the local system in this example. A trap will also be sent to NetView for AIX when the content of the file is changed. In this case trap 22, the File Modified trap, is sent.

Figure 172. Entry in File Monitor Table for Monitoring /etc/hosts

An example of the event that will appear at NetView for AIX is shown below in Figure 173 on page 197.

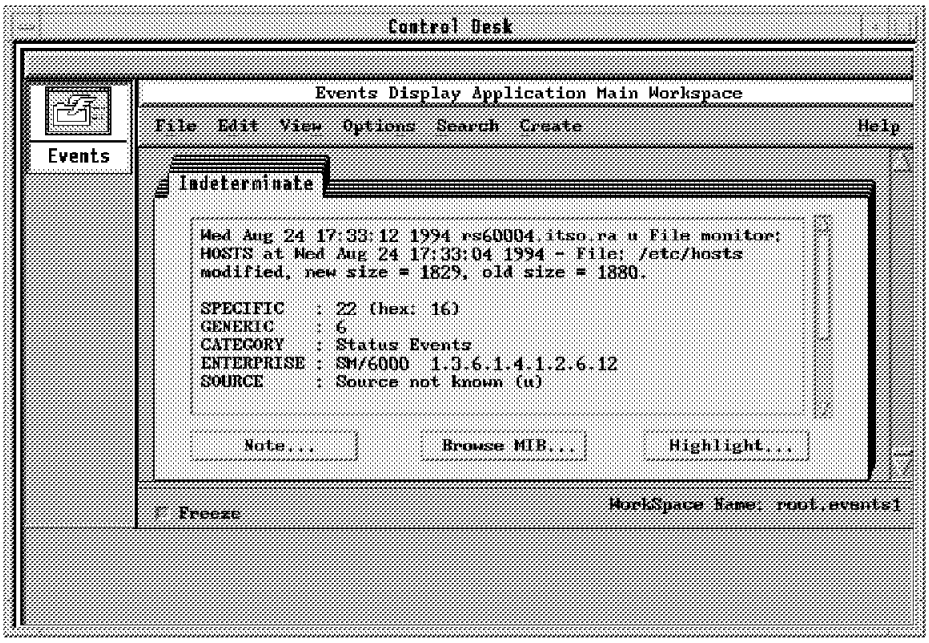


Figure 173. Event Generated When /etc/hosts Is Modified

The default event configuration for this event can be seen in Figure 174 on page 198.

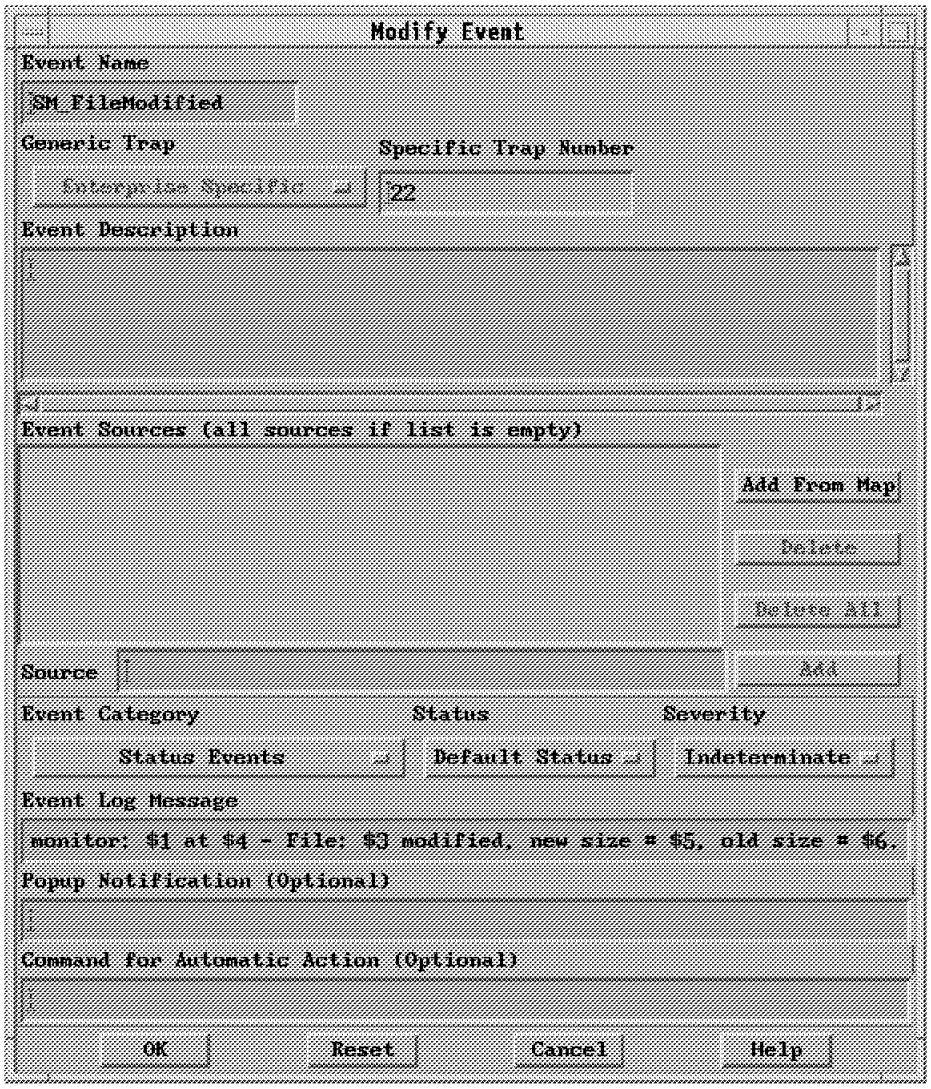


Figure 174. Event Configuration for File Modified Event

You can see in the previous two figures that there are five variables passed through for this event. They are as follows:

$1 The name of the entry in the File Monitor Table

$3 The name of the file that was changed

$4 The date and time at which the file was modified

$5 The new size of the file

$6 The old size of the file

7.1.7 Monitoring /etc/resolv.conf to Verify It Exists

As an IP network grows and becomes more dynamic the static name resolution of /etc/hosts becomes cumbersome. This leads the network administrator to implement a name service, such as the Domain Name Service (DNS).

The file /etc/resolv.conf must exist on every client in a domain name serving environment. In this environment, only one node, the domain name server, has a list of all the hostname to IP address mappings and all the other nodes in the

network have a file which points to the server. On the RISC System/6000 this file is called /etc/resolv.conf and it is important to monitor the existence of this file, since if it is deleted for any reason, the local node will no longer be able to resolve hostnames.

An entry can be added to the local hosts File Monitor Table to check for the existence of the file. The entry configuration can be seen in Figure 175.

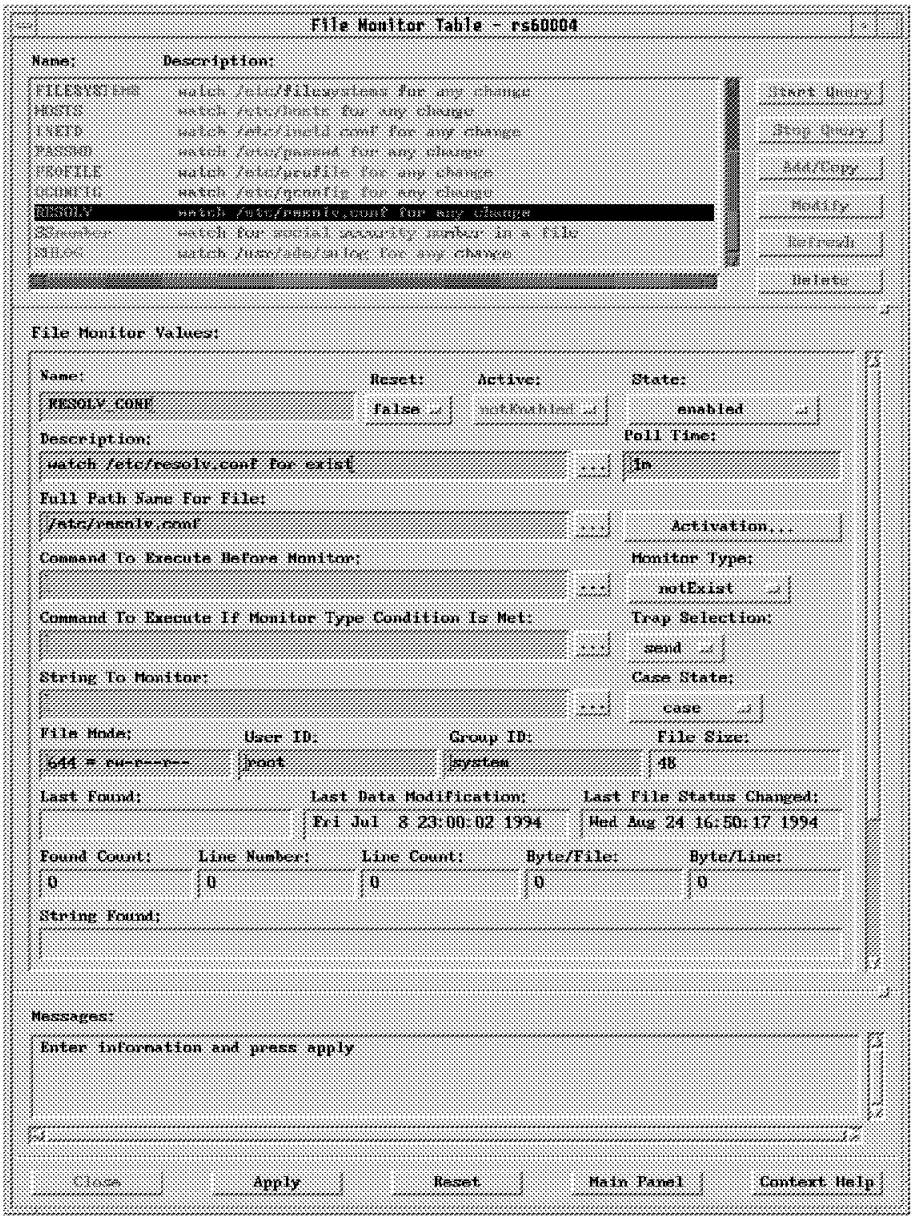


Figure 175. Monitoring Existence of resolv.conf File Monitor Table Entry

In this scenario although we want to monitor for the existence of the file /etc/resolv.conf, we actually configure the table entry to have a monitor type of notExist. The file should always exist on a system, but if it is deleted for any reason, then a trap, with specific number of 24, will be sent to NetView for AIX. The format of this trap can be seen in Figure 176 on page 200.

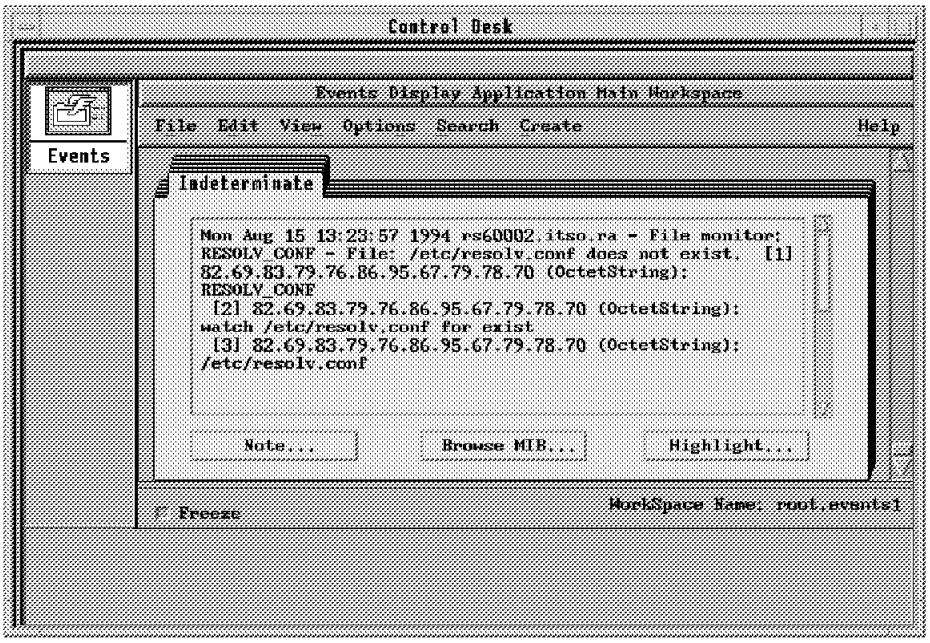


Figure 176. Event Card Indicating the Non-Existence of /etc/resolv.conf

The configuration for trap 24 which generates the event card, is shown in Figure 177.

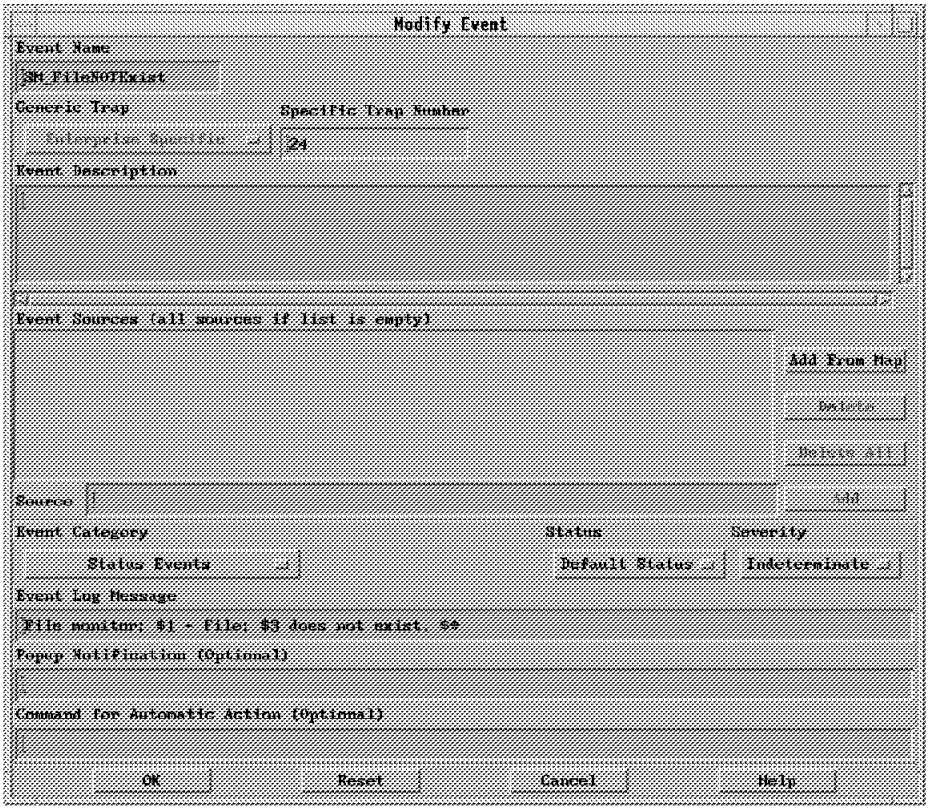


Figure 177. File Does Not Exist Event Card Configuration

In order to monitor whether or not a file exists and generate an event when the file is created, the same procedure detailed above can be followed, with the file monitor type set to Exist rather then notExist. In this case a trap will be

generated when the file is created with a specific trap number of 25. The format of the trap is shown in Figure 178 on page 201.

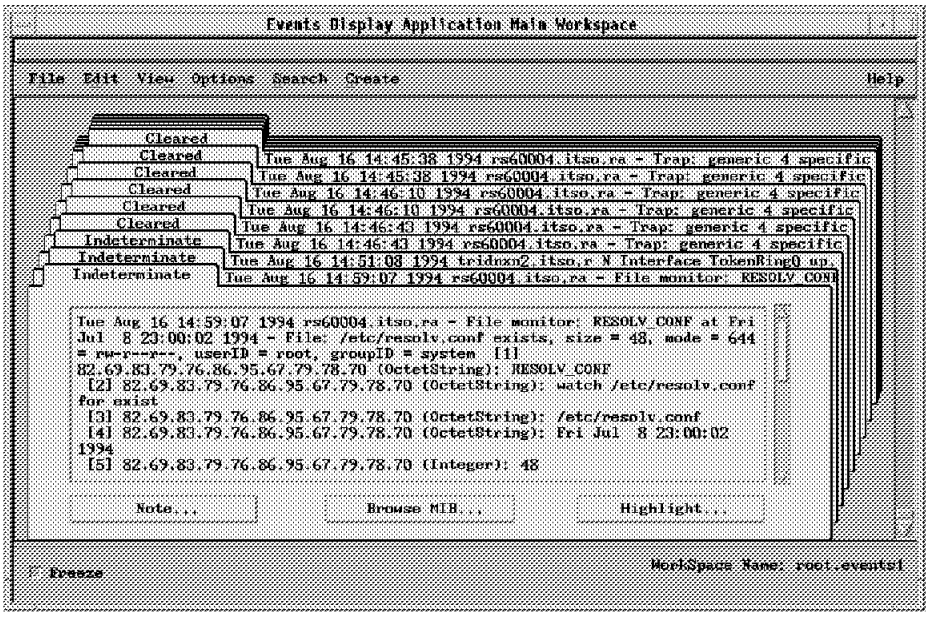


Figure 178. File Exists Event Card

7.1.8 SNA Session Status Monitoring

The RISC System/6000 SNA subsystem has very similar monitoring requirements to the printing process described previously (7.1.1, “Monitoring Printer Status” on page 162). In this example we show how the Command Table can be used to monitor session status for LU6.2 session.

We set up SNA monitoring for the environment shown in Figure 179 on page 202.

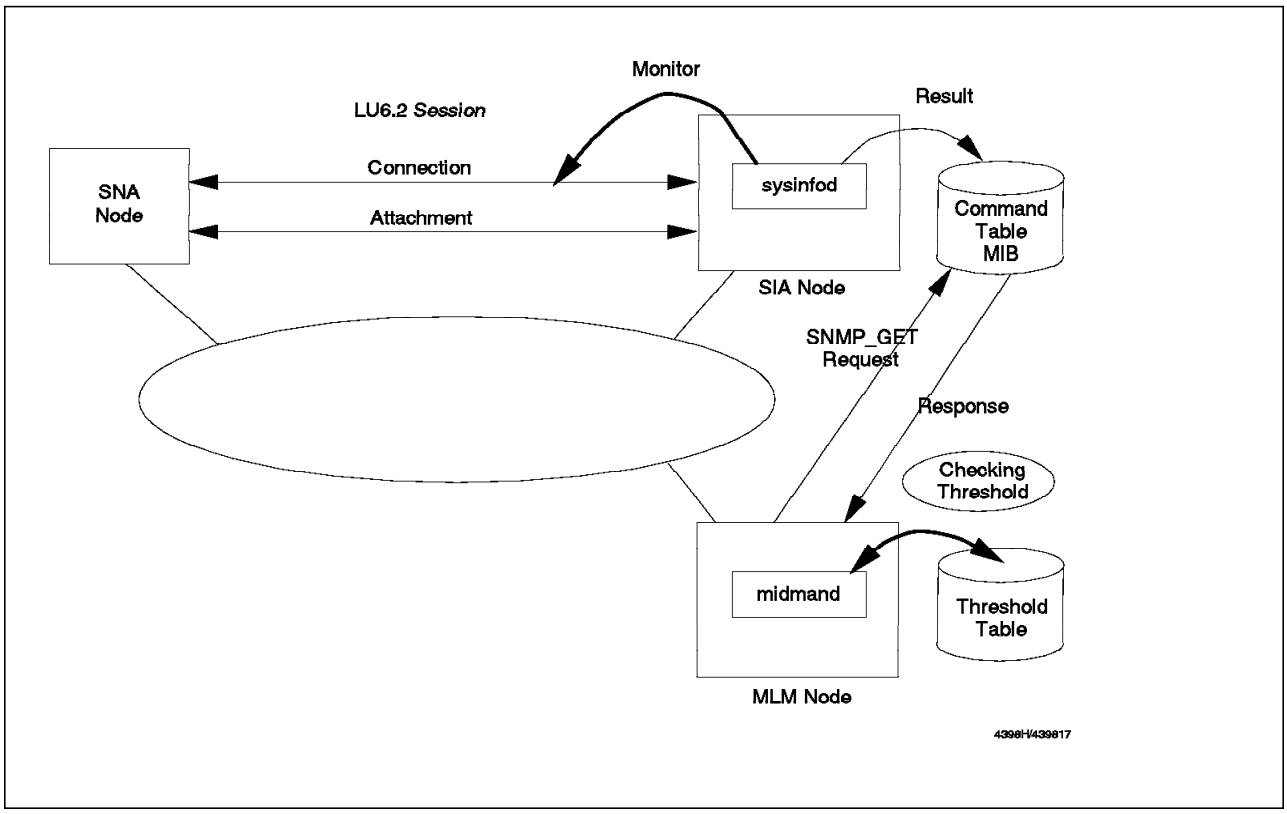


Figure 179. System Environment for Monitoring SNA LU6.2 Sessions

The command normally used for checking SNA session status is `lssrc -ls sna`. In our sample environment the output of this command is as shown:

```
 "sna" Program, Process ID 33542  active
NVIXTA01             Attachment    - active
NVIXLCMDS1           Connection    - active
```

Figure 180. Sample Output of lssrc -ls sna

We can see from this that SNA Services is active, with an attachment and a single LU6.2 session established.

We could monitor the SNA subsystem alone using the standard SIA `smSiaSystemSubSystems` MIB objects (see also 7.1.2, "Monitoring Status of the lpd Daemon" on page 165). However there is no SIA MIB for monitoring SNA sessions. Therefore we decided to use the Command Table.

First we created a shell script to parse the result of the `lssrc -ls sna` command:

```
#!/bin/ksh

USAGE="SNAMON Attachment_Name Connection_Name"

lssrc -ls sna > /tmp/snatmp

ATT_NAME=$1
CON_NAME=$2

SNA_DOWN=`cat /tmp/snatmp | awk 'NR == 1 {print $NF}'`

if [[ $SNA_DOWN != "active" ]]
then
   print 2
   exit
else
   ATT_DOWN=`cat /tmp/snatmp | grep $ATT_NAME | awk '{print $NF}'`

   if [[ $ATT_DOWN != "active" ]]
   then
      print 2
      exit
   else
      CON_DOWN=`cat /tmp/snatmp | grep $CON_NAME | awk '{print $NF}'`

      if [[ $CON_DOWN != "active" ]]
      then
         print 2
         exit
      fi
   fi
fi

print 1

rm /tmp/snatmp
```

Figure 181. Sample Shell Script for Monitoring SNA Sessions

If there is any trouble on the managed SNA LU6.2 sessions this shell script returns 2 and if SNA sessions are working correctly, it returns 1. Figure 182 on page 204 shows the configuration of the Command Table for executing this script.

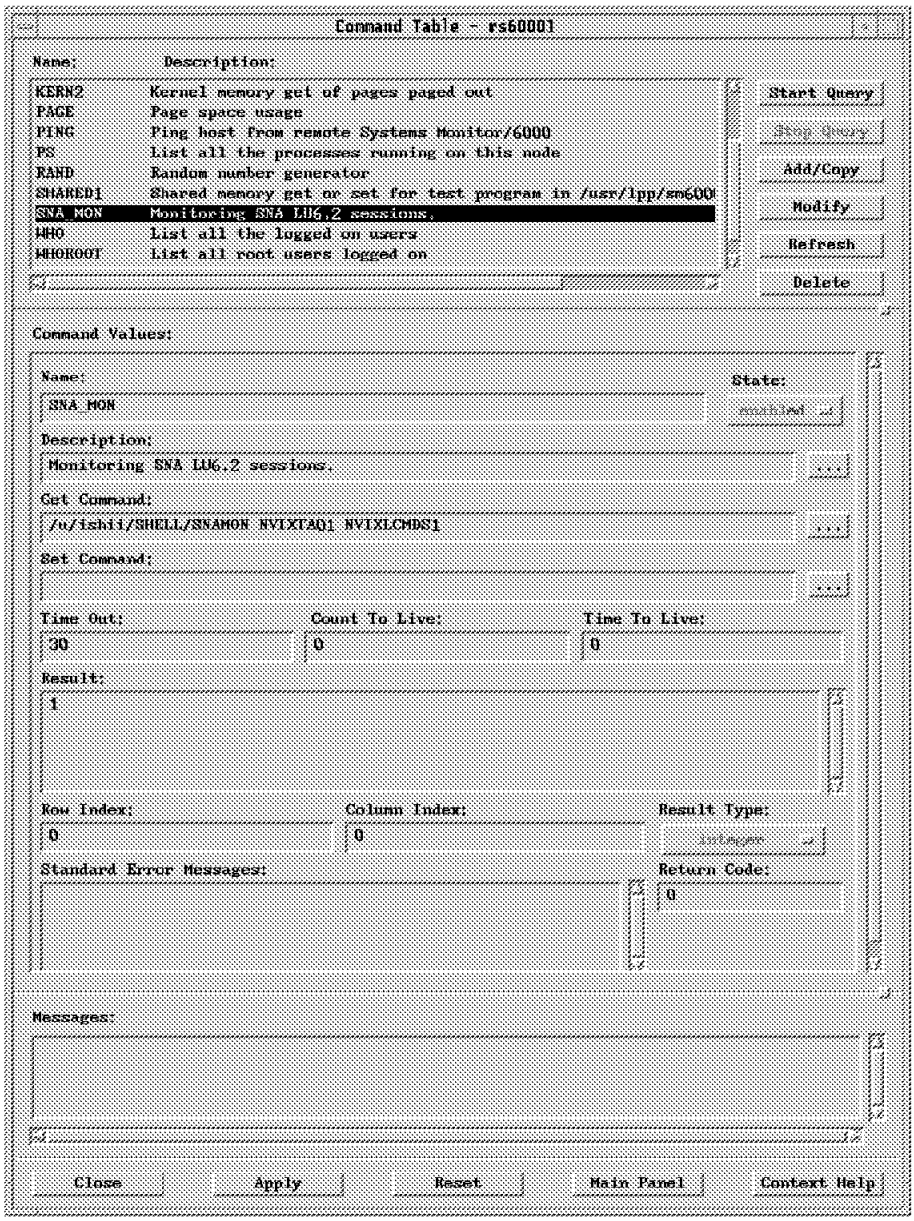


Figure 182. SNA LU6.2 Monitor Command Table Entry

We created a Threshold Table entry to repeatedly poll this command. The threshold was configured to automatically attempt to restart the SNA session, and to report the session failure to the NetView for AIX operator. Figure 183 on page 205 and Figure 184 on page 205 show the Threshold Table configuration panels. The MIB variable we are polling, .1.3.6.1.4.1.2.6.12.4.1.1.14.SNA_MON is the integerResult field of the SIA Command Table. See page 167 for further explanation of how to specify the Local/Remote MIB Variable field.

Figure 185 on page 206 shows the shell script that we used to attempt the LU6.2 session restart.

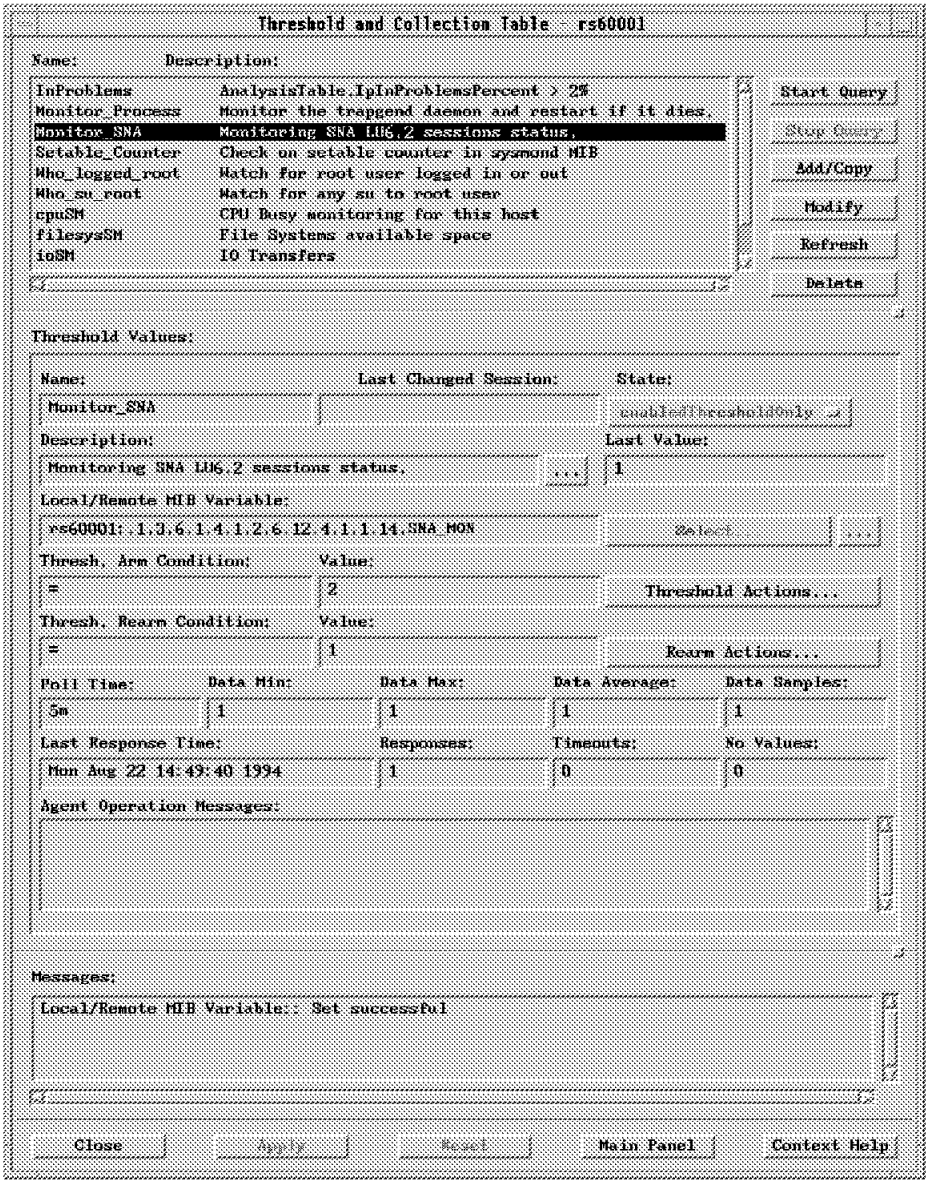


Figure 183. SNA LU6.2 Sessions Monitoring Threshold Table Entry

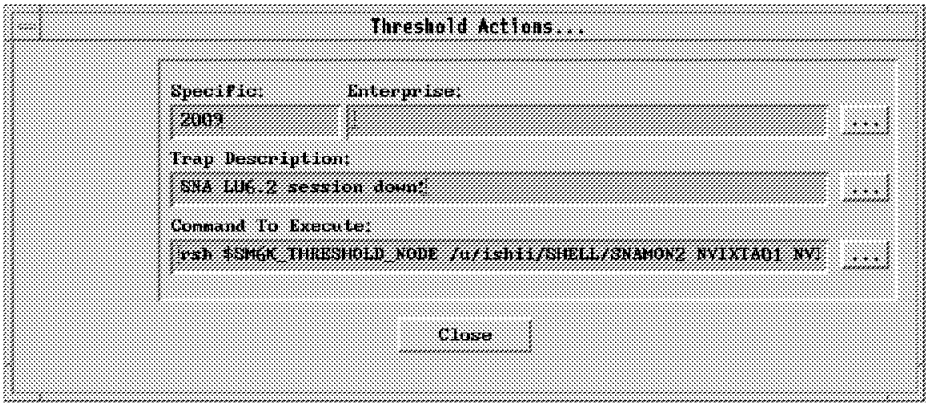


Figure 184. SNA LU6.2 Monitoring Threshold Action

```
#!/bin/ksh

USAGE="SNAMON2 Attachment_Name Connection_Name"

lssrc -ls sna > /tmp/snatmp2

ATT_NAME=$1
CON_NAME=$2

SNA_DOWN=`cat /tmp/snatmp2 | awk 'NR == 1 {print $NF}'`

if [[ $SNA_DOWN != "active" ]]
then
   startsrc -s sna
   startsrc -tattachment -o$ATT_NAME
   startsrc -tconnection -o$CON_NAME
else
   ATT_DOWN=`cat /tmp/snatmp2 | grep $ATT_NAME | awk '{print $NF}'`

   if [[ $ATT_DOWN != "active" ]]
   then
      startsrc -tattachment -o$ATT_NAME
      startsrc -tconnection -o$CON_NAME
   else
      CON_DOWN=`cat /tmp/snatmp2 | grep $CON_NAME | awk '{print $NF}'`

      if [[ $CON_DOWN != "active" ]]
      then
         startsrc -tconnection -o$CON_NAME
      fi
   fi
fi

rm /tmp/snatmp2
```

Figure 185. Shell Script for Restarting SNA Sessions

Considerations if Using an SLM in Place of the MLM: This example would operate in the same way if an SLM was used instead of an MLM, except:

- The MIB variable to be monitored would not be prefixed with the node name (rs60001), because the SLM would have to be running on rs60001 (it can only monitor the node it is on).
- In the threshold action definition there would be no need to use rsh to remotely execute the SNAMON2 shell script to restart the connection.

Notes

This sample was written for SNA Services/6000. It would have to be modified for use with SNA Server.

7.2 Security Monitoring

The distributed computing environment offers some special security challenges which have only begun to be seriously addressed in recent times. Systems such as DCE and the IBM Network Security Program seek to implement a unified security structure across the distributed environment. However, in most real configuration security is still specific to each individual machine. Furthermore, even with a fully distributed security structure, each machine needs a local super-user (or *root*) user ID for maintenance purposes. If an imposter gains knowledge of the root password, or works out a procedure to log into a system as super-user, he has the capability to destroy everything installed on the system.

In this section we show three examples of how Systems Monitor for AIX can help you to detect attempts to break super-user security:

- 7.2.1, “Who su Configuration” shows an example of how to monitor each switch to super-user by ordinary user IDs.
- 7.2.2, “Monitoring /etc/passwd for Status Changes” on page 213 shows monitoring of the password file for any alterations.
- 7.2.3, “Monitoring for Failed Login Attempts” on page 217 shows monitoring of failed login attempts using the File Monitor string searching function.

7.2.1 Who su Configuration

Whenever a user attempts to su to root, an entry is added to the file `/usr/adm/sulog`, with details of whether or not the su was successful, the date and time that the attempt was made, and the user ID making the attempt. This example shows how the Systems Monitor Command Table and Threshold Table can be used to monitor these attempts and notify NetView for AIX operators when they occur.

7.2.1.1 Who su Command Table Entry

We added an entry to the Command Table to monitor attempts to su to root, as shown in Figure 186 on page 208. This entry counts up the number of entries in the file /usr/adm/sulog where users have been both successful and unsuccessful to su to root.

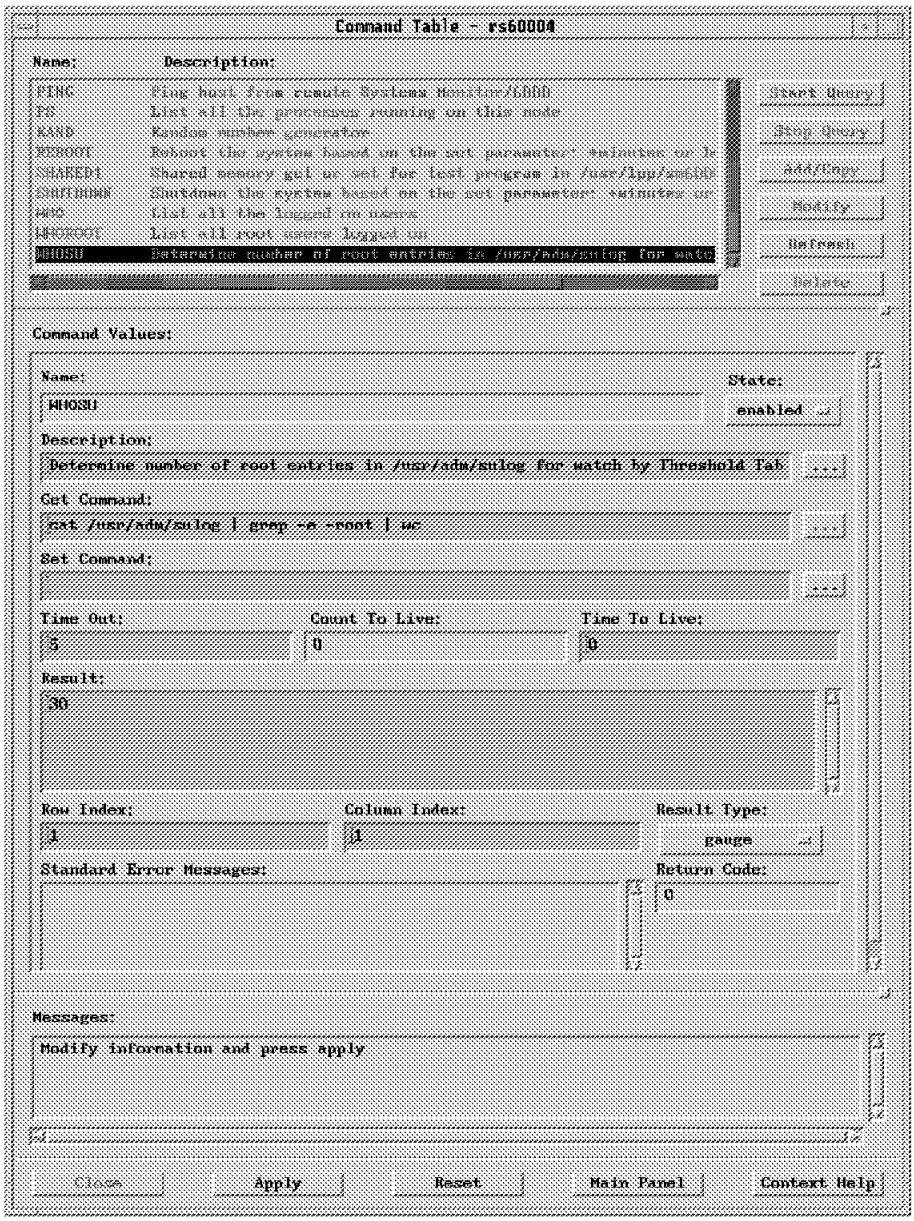


Figure 186. Who su Command Table Entry

As you can see, the last time the command was run, there were 30 entries in the file (that is, 30 users had attempted to su to root). This number alone is not very interesting. However, an entry can be made in the Threshold Table to run this command on a regular basis, (every 5 minutes, say) and if the value of 30 changes, indicating that another user has attempted the su command, a trap will be sent to NetView for AIX.

7.2.1.2 Who su Threshold Table Entry

We want to threshold on the result that is generated when the whosu command is run. It can be seen from Figure 186, that the Result Type field for the WHOSU Command Table entry, is gauge. Therefore we shall threshold on the MIB variable `smSiaCommandGaugeResult`, that has an object ID of `1.3.6.1.4.1.2.6.12.4.1.1.16`. The NetView for AIX MIB browser can be used to determine the object ID of a MIB variable, as shown in Figure 187 on page 209.

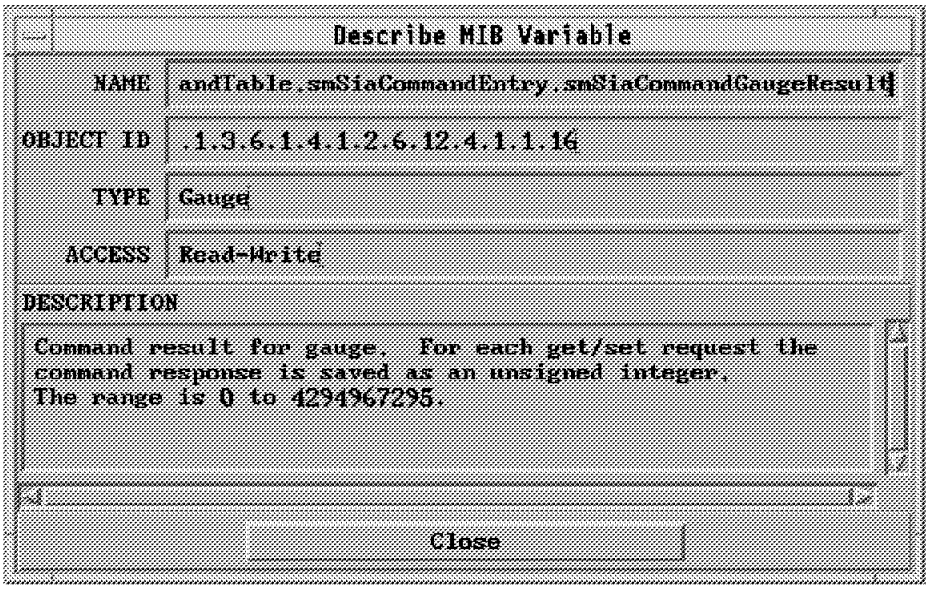


Figure 187. Object ID of MIB Variable smSiaCommandGaugeResult

We cannot enter just the MIB object ID as the variable to be thresholded on in the Threshold Table. We need to append the instance ID for the particular instance that we are interested in, which in this case is the WHOSU entry in the Command Table. Therefore we must append the value WHOSU to the object ID, as shown in Figure 188 on page 210.

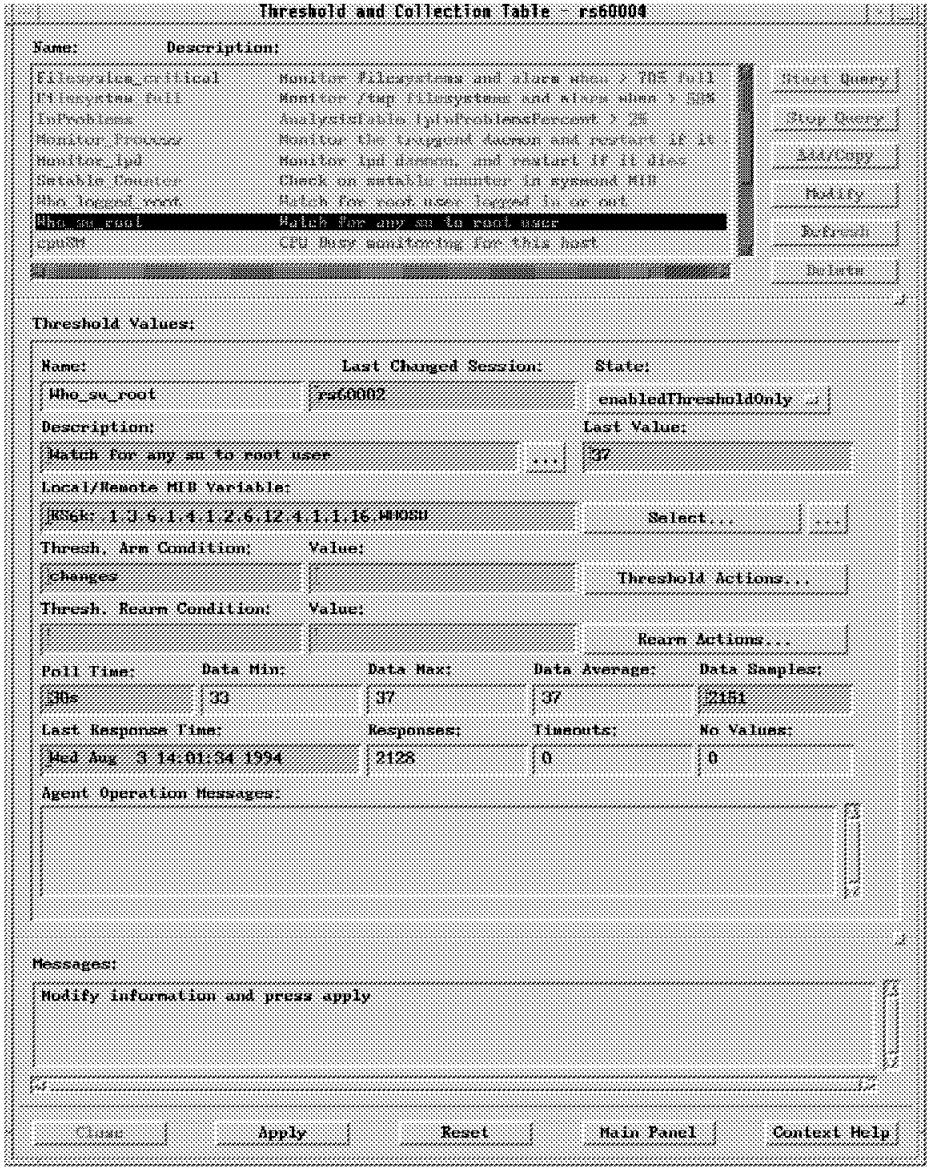


Figure 188. Who su Threshold Table Entry

See page 167 for further explanation of how to specify the Local/Remote MIB Variable field. This entry will poll the MIB variable smSiaCommandGaugeResult every 30 seconds for hosts rs60002 and rs60004 (that is, the hosts contained in the alias RS6k).

The actions that we defined to be taken when the MIB variable changes can be seen in Figure 189 on page 211.

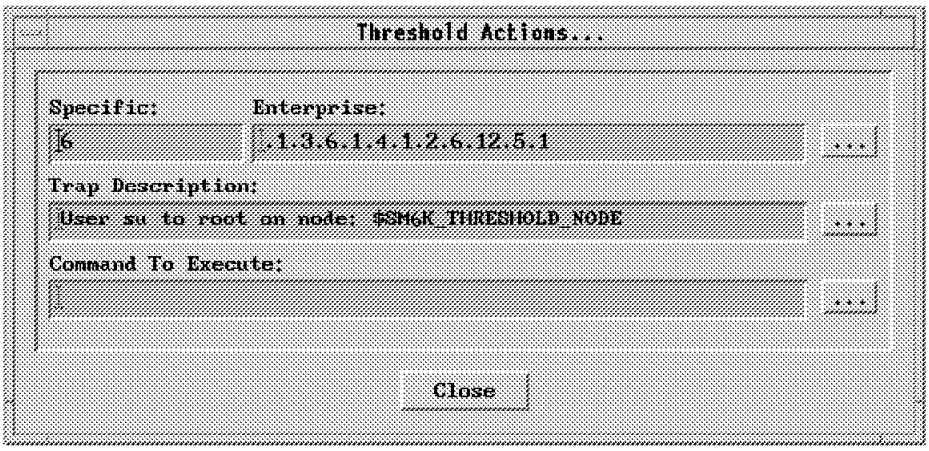


Figure 189. Who su Threshold Actions

The action that will be taken when the threshold condition has been met, is to send a trap, with an enterprise of .1.3.6.1.4.1.2.6.12.5.1 (which indicates that it is a Systems Monitor for AIX threshold trap) and a specific trap number of 6.

The next step will be to configure this event at NetView for AIX.

Considerations if Using an SLM in Place of the MLM: This example would operate in the same way if an SLM was used instead of an MLM, except that the MIB variable to be monitored would not be prefixed with the alias name (RS6k), because the SLM would have to be running on each node being monitored (it can only monitor the node it is on).

7.2.1.3 Who su Event Configuration

The NetView for AIX event configuration for this event is shown in Figure 190 on page 212.

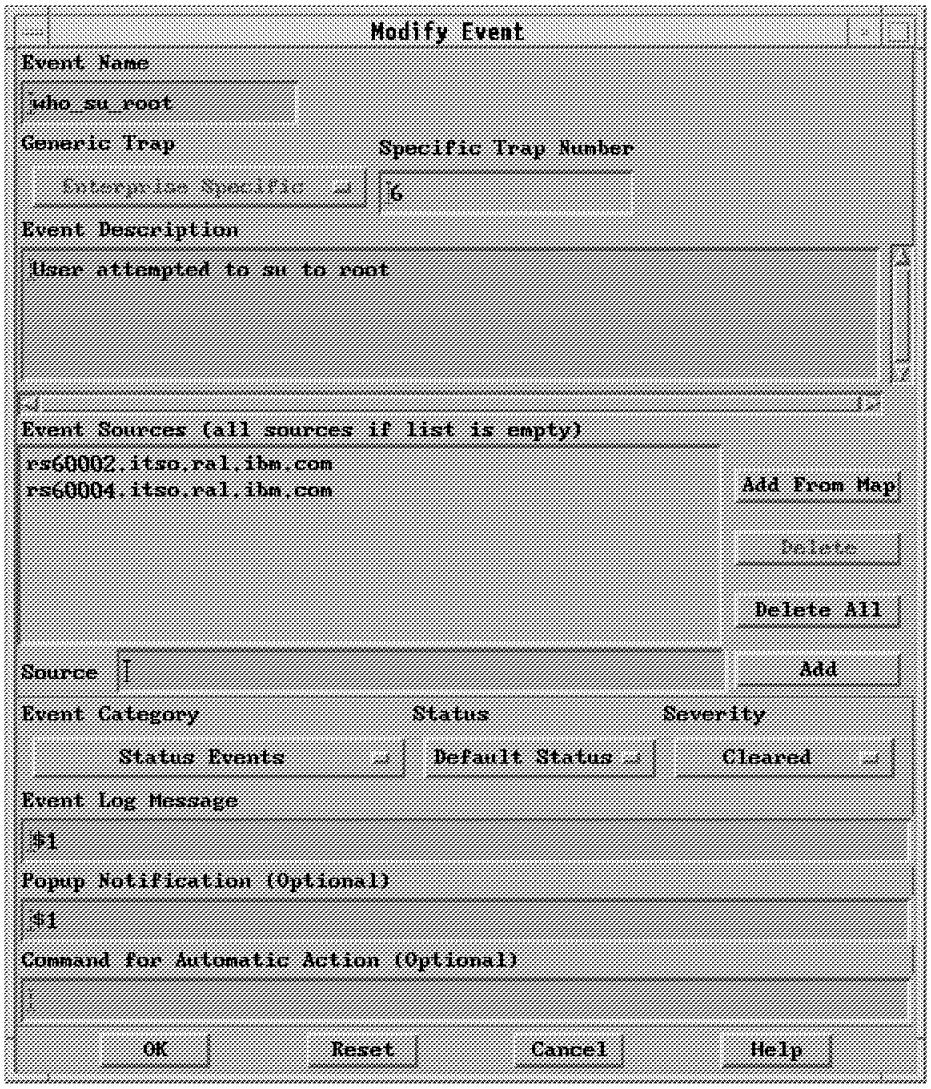


Figure 190. Who su Event Configuration

When the event is configured it is possible to specify the event sources that it should apply to. A blank field indicates that all source names or IP addresses are valid, but in this scenario the event will only be generated for nodes rs60002 and rs60004. The event log message and the pop-up notification have an entry $1, which corresponds to the Systems Monitor threshold trap description. Hence, both the event card and the pop-up window will contain the information that was specified in the Threshold Table trap description field in Figure 189 on page 211.

The pop-up window that is generated is shown in Figure 191 on page 213.

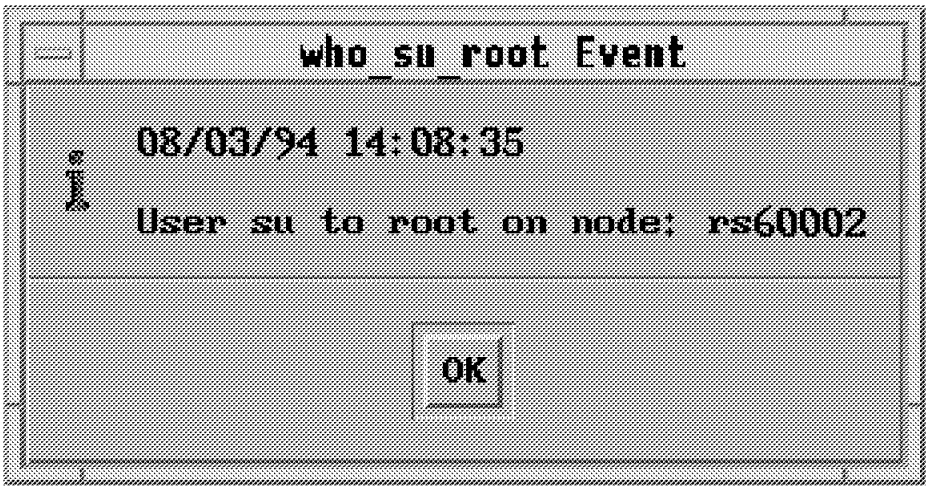


Figure 191. Pop-Up Window

7.2.2 Monitoring /etc/passwd for Status Changes

In this example we are monitoring the status of the file /etc/passwd. This is the file containing information about all the user IDs on the system. If somebody has access to this they can gain control of the system. For example, it would be possible to alter the home directory of root and thereby cause damage by installing a bogus .profile file.

In this example we use the Systems Monitor for AIX File Monitor table so that if any changes are made in the permissions or ownership of /etc/passwd a message will be sent to the root user and a trap will be sent to NetView for AIX. The destination of the trap is determined by the trap entry in the file /etc/snmpd.conf (see 2.2, “Configuring SNMP” on page 25 for SNMP configuration examples). For file status changes the SIA sends specific trap 23.

The entry that was made in the File Monitor Table is shown in Figure 192 on page 214.

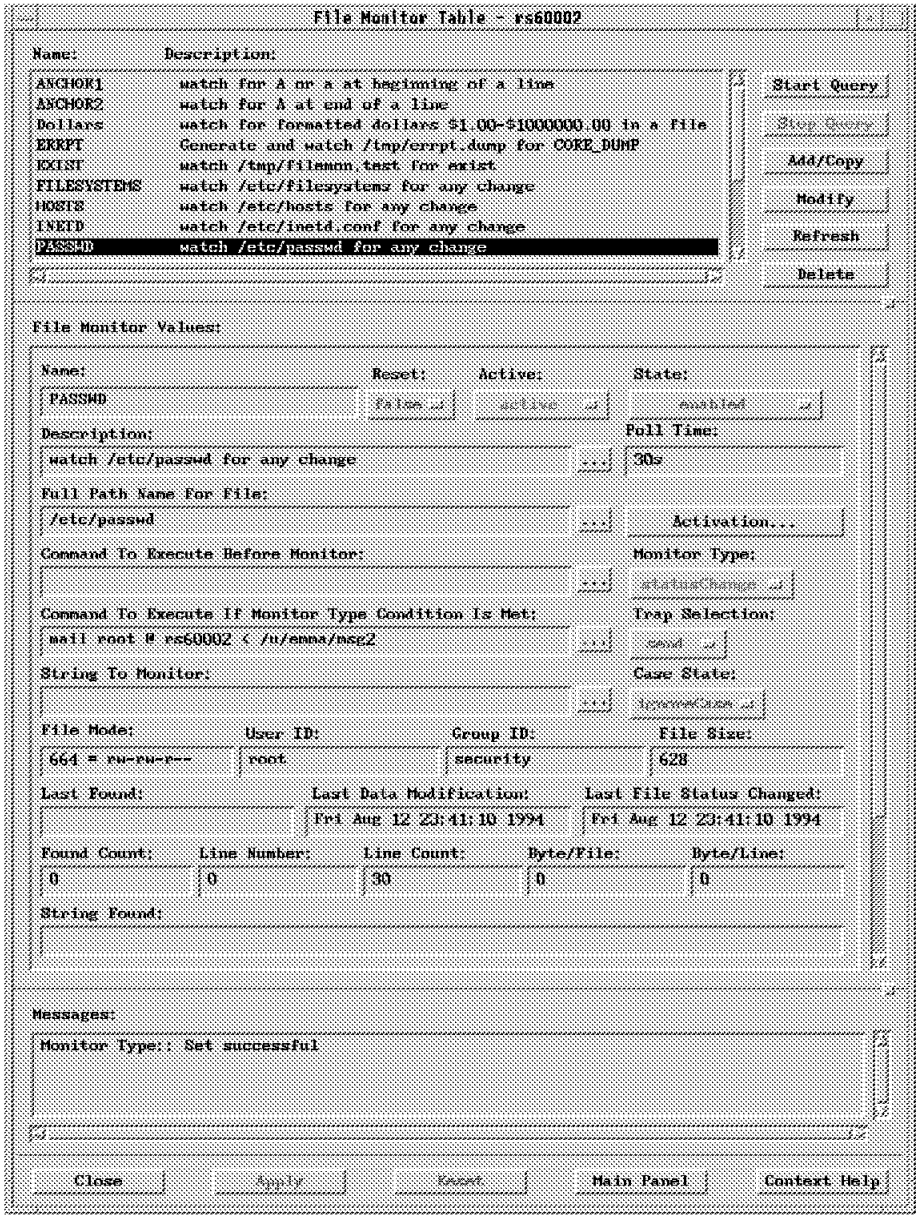


Figure 192. Entry in File Monitor Table for Monitoring /etc/passwd

You can see that the default file mode for /etc/passwd, is 664, that is:

- Read-write access for user root
- Read access for group security
- Read access for all other users and groups

By default, this file is owned by user root in group security.

Since we are interested in monitoring status changes, the Monitor Type field is set to statusChange. We want to send a trap to NetView for AIX if the status of the file is changed, so the Trap Selection field is set to send.

7.2.2.1 Changing Ownership of Passwd File

If a user attempts to modify the ownership of the /etc/passwd file, this will be reflected on the File Monitor Table, when it is re-queried. An example of this is shown in Figure 193.

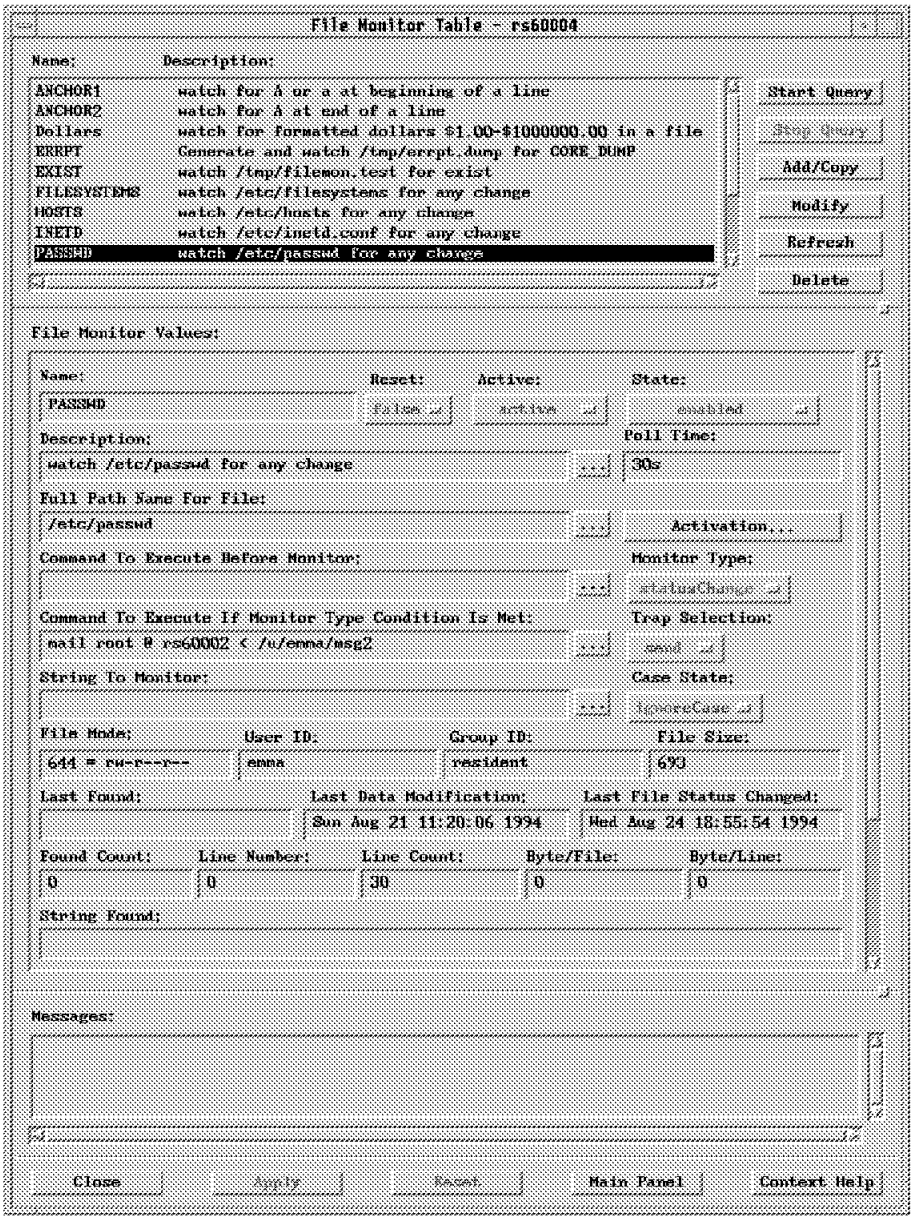


Figure 193. Querying File Monitor Table

We can see that user emma modified the file on Wednesday August 24th at 15:15 and that the file is now owned by user emma, who is a member of the group called resident.

When this ownership change is made an event will be sent to NetView for AIX. By default the event card will carry all this detailed information, including values for the old user ID, group ID and mode, as well as the new values. The event card can be seen in Figure 194 on page 216.

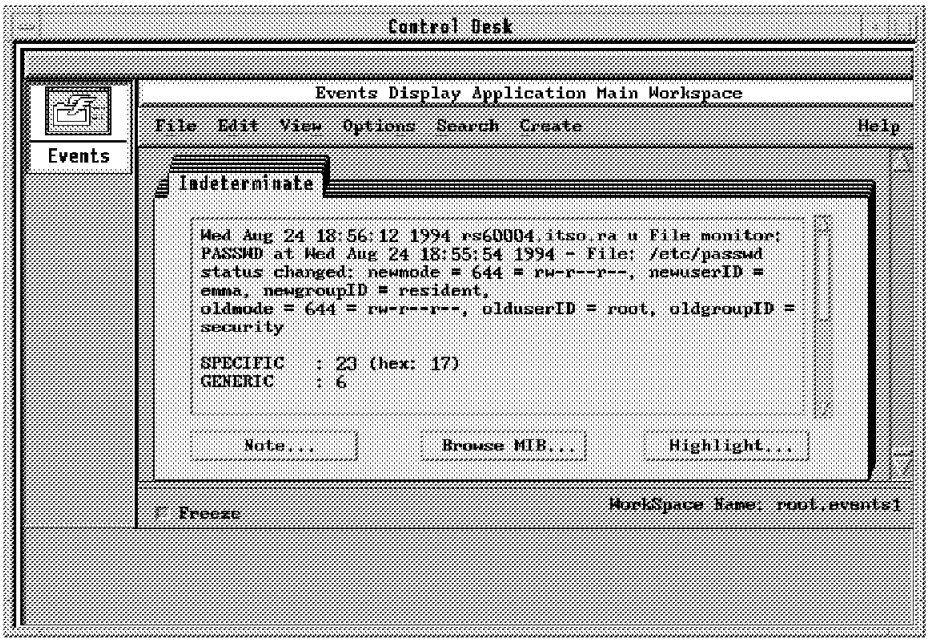


Figure 194. Default File Status Change Event Card

In this scenario the event that is generated is not a Systems Monitor for AIX threshold event, but a pre-defined event with a specific trap number of 23. The enterprise ID is *SM/6000* (enterprise ID 1.3.6.1.4.1.2.6.12). This event is called SM_File_Changed and will not need to be added like the threshold events since it already exists. The event configuration may be modified, but the specific trap number may *not* be changed.

The default event configuration, which resulted in the event card in Figure 194, has an event log message configured to be:

```
File Monitor: $1 at $4 - File: $3 status changed: newmode = $5, newuserID =
$6, newgroupID = $7, \noldmode = $8, olduserID = $9, oldgroupID = $10.
```

We can see that these variables are different from those included in the Systems Monitor for AIX threshold traps and by comparing the event card and the event configuration, we can see that:

- $1 is the name of the entry in the File Monitor Table.
- $3 is the name of the file that was changed.
- $4 is the date and time that the change took place.
- $5 is the new mode of the file, that is the new file permissions.
- $6 is the user ID of the new user who owns the file.
- $7 is the group ID of the new user who owns the file.
- $8 is the old file permissions.
- $9 is the user ID of the old owner.
- $10 is the group ID of the old owner.

In this scenario, the file permissions did not change and therefore variables $5 and $8 are the same.

The event card may be configured to include less detailed information if required.

7.2.2.2 Changing Permissions of /etc/passwd

As we saw earlier, the default permissions of the passwd file, are to only allow read-write access for the root user, all other users are only able to read the file. If any user attempts to change these attributes, a trap will be sent to NetView for AIX and since this is still a status change, trap 23 will be sent. The event card that will be generated can be seen in Figure 195. In this case, the variables $6, &7, $9 and $10 which represent the old and new user ID and group IDs remain unaltered.

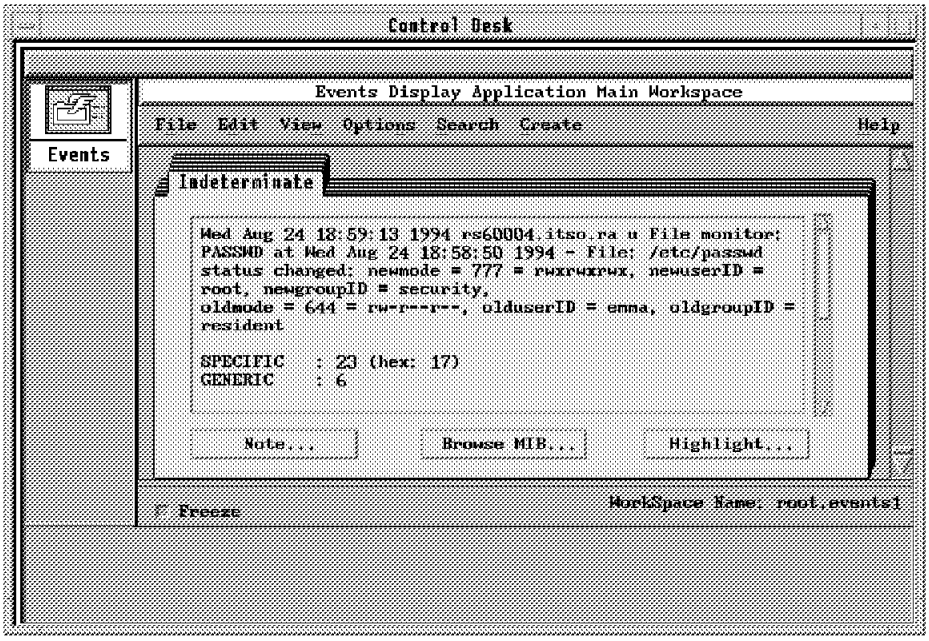


Figure 195. File Permissions Event Card

7.2.3 Monitoring for Failed Login Attempts

One ″brute force″ method to break through system security is to try to guess the root password. Unless the attacker is very lucky it will normally take many failed attempts before he discovers the password. In this example we show a technique for intercepting such attacks, by monitoring for failed login attempts.

AIX maintains a list of failed login attempts in file /etc/security/failedlogin. This file is in a binary format that is not very easy to handle. However there is an accounting command, `fwtmp`, provided with AIX that will format it as an ASCII file. The same command can turn the ASCII version back into binary.

The monitoring process we want to set up is therefore:

1. Format /etc/securityfailedlogin into ASCII form.
2. Check for occurrences of the string ″root″.
3. Send a trap to the NetView for AIX operator if any are found.

All of these steps can be performed by a single SIA File Monitor Table entry. The table entry we created is shown in Figure 196 on page 218.

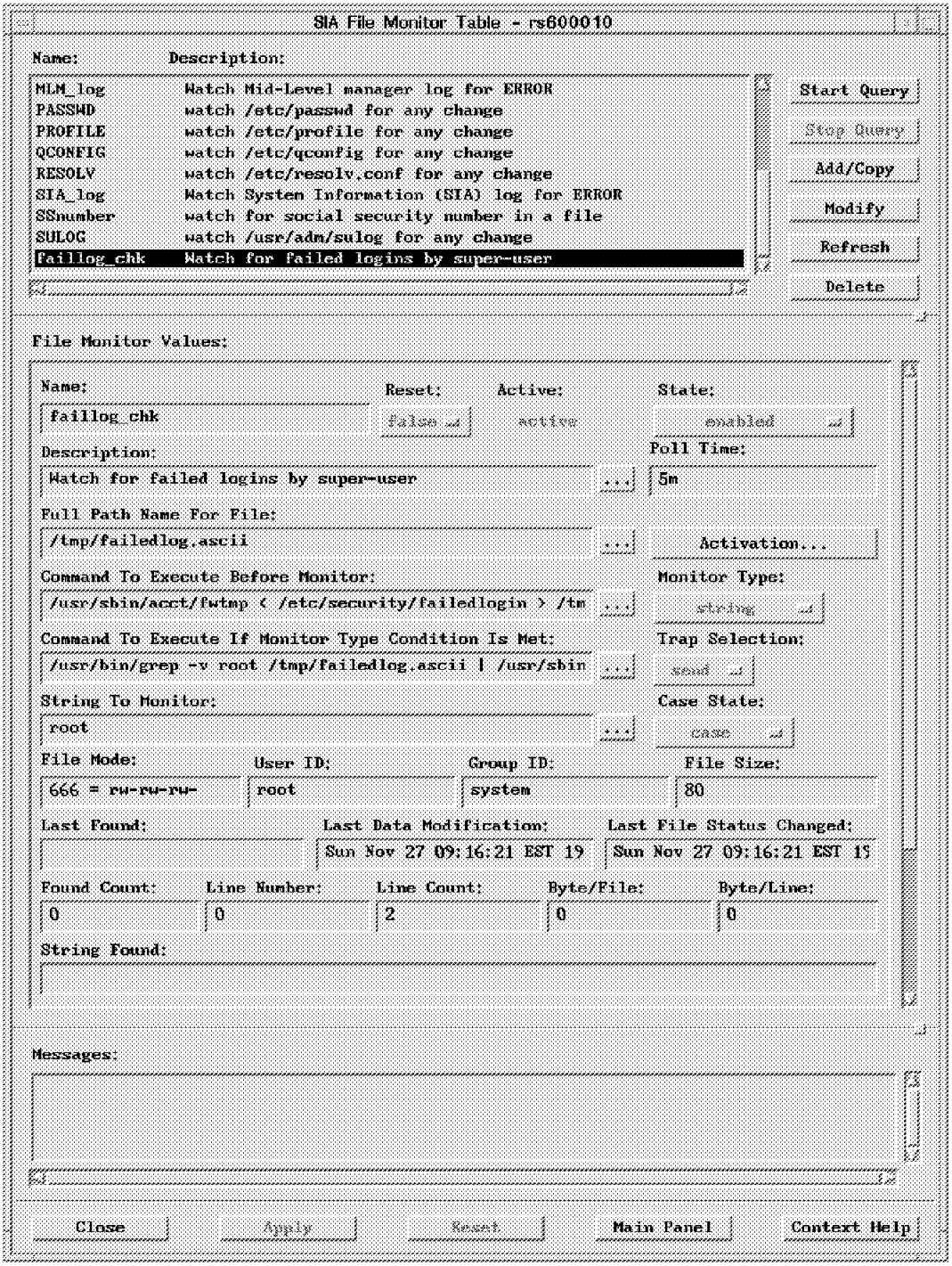


Figure 196. File Monitor Table Entry for Failed Logins

The entry is set to test the file every 5 minutes. When the polling interval expires, the command specified in the Command to Execute Before Monitor field is run. In this case the command is:

```
/usr/sbin/acct/fwtmp < /etc/security/failedlogin > /tmp/failedlog.ascii
```

This uses the fwtmp utility to *create* the file that we want to monitor (/tmp/failedlog.ascii).

Once the initial command has completed, the ″normal″ file monitoring cycle begins. You can see that we have defined ″string″ in the Monitor Type field, meaning that we want to check for the occurrence of a specific string in the file. We have selected **send** to cause a trap to be sent when the string is present.

The problem with this approach is that the original source file, /etc/security.failedlogin, still contains the records of root's previous failures, so at the next polling cycle we will create another set of traps reporting the same events. We circumvented this problem by using the Command to Execute When Monitor Type Condition is Met field to execute the following command:

```
grep -v root /tmp/failedlog.ascii | /usr/sbin/acct/fwtmp -ic > /etc/security/failedlogin
```

This writes all *except* the records containing ″root″ into the failedlogin file in binary form. In this way, at the next polling cycle any root failed login records are sure to be new ones.

The event card resulting from this monitor detecting a failed attempt to login as root is shown in Figure 206 on page 229.

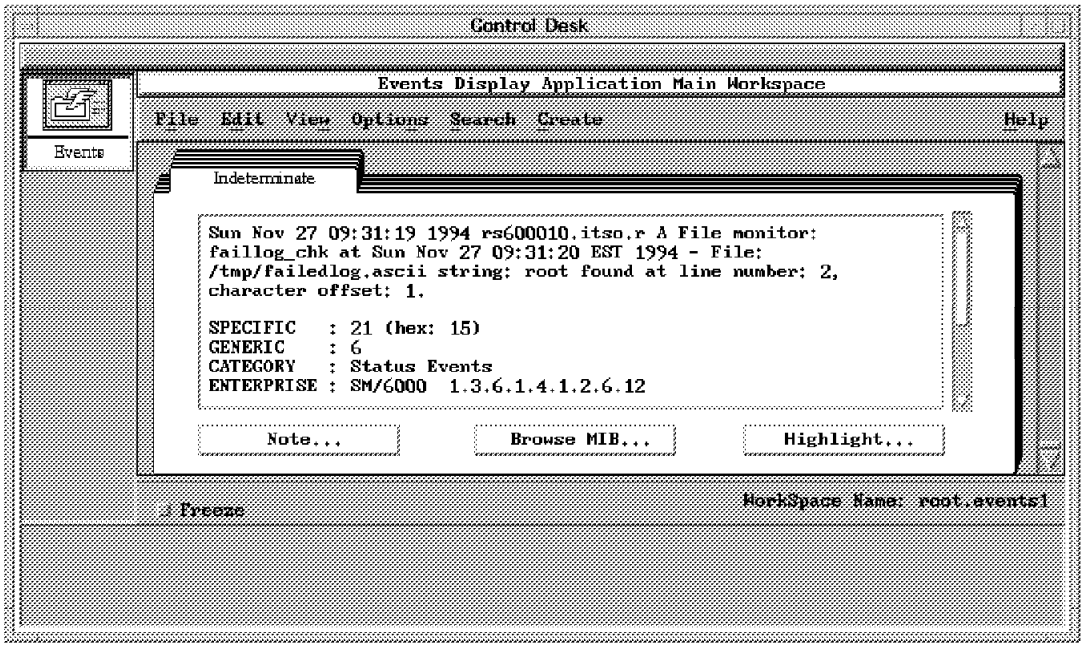


Figure 197. Event Card Showing Failed Login Attempt

From the card we can see that the text string monitor uses a different specific trap ID from the file status change monitor (21 instead of 23). The default event configuration for trap 21 is as follows:

```
File Monitor: $1 at $4 - File: $3 string: $10 found at line number:  $11,
character offset: $12.
```

Where:

- $1 is the name of the entry in the file monitor table.
- $3 is the name of the file being monitored.
- $4 is the date and time the file was last modified.
- $10 is the name of the string that was found.

- $11 is the line number at which the string was found.
- $12 is the character offset.

7.3 Performance Monitoring

In this section we will look at approaches to monitoring system and network performance. The distributed polling mechanism of the MLM, combined with the extended system monitors provided by the SIA make Systems Monitor for AIX a powerful tool for this task. Performance monitoring and thresholding is important to allow administrators to be alerted to potential problems in a timely manner.

You will find the following examples here:

- 7.3.1, “Monitoring Paging Space” uses the Alias Table, the Threshold Table and the SIA to show health monitoring for a remote system.
- 7.3.2, “Monitoring Processor Utilization” on page 225 uses the Threshold Table to monitor and collect performance information from a network node (a 6611).
- 7.3.3, “Monitoring File System Utilization” on page 227 uses the Threshold Table to monitor the disk resources on a distributed system.
- 7.3.4, “Automatic Response to File System Full Errors” on page 231 extends the previous example, using the Command Table to automatically allocate more disk resource.
- 7.3.5, “Representing Systems Monitor Thresholds As NetView Symbols” on page 235 shows how the graphical EUI of NetView for AIX can be used to highlight performance thresholds.
- 7.3.6, “Monitoring Excessive CPU Utilization for Processes” on page 242 uses the Threshold Table to monitor the most active processes on a system.
- 7.3.7, “Monitoring the Percentage of IP Datagrams in Error” on page 246 uses the Analysis Table to show how to derive a calculated figure by combining MIB variables.
- 7.3.8, “NFS Performance Monitoring” on page 250 shows another application of the Command Table to examine NFS statistics

7.3.1 Monitoring Paging Space

Our objective in this example was to monitor the percentage of paging space used on a group of file servers (nodes: rs60001, rs60002, rs6003, rs60004, rs600010 and rsserver). We wanted to alert the NetView for AIX operator when utilization exceeded 60%. This allows the system administrator to fix the problem before it causes processes to fail.

The monitoring is done by the Mid-Level Manager using the Threshold Table and Alias Table functions. It reports any exceptions by sending traps to NetView for AIX. The nodes being monitored have either the Systems Information Agent or Systems Monitor/6000 V1 installed. The MIB variable that is being monitored is a Systems Monitor MIB variable with the same object ID in both versions.

The following steps are needed to set this scenario up:

1. Add an entry for the group of file servers in the MLM Alias Table.

2. Add an entry to the MLM Trap Destination Table so that traps are sent to the NetView for AIX machine.
3. Add an entry to the Threshold Table, to monitor the paging space and send a trap when the 60% target is exceeded.
4. Configure the threshold and rearm events on NetView for AIX.

7.3.1.1 Configuring the Alias Table

This table is used to specify an alias name that will be used to describe a group of nodes. These alias names will then be used in other Mid-Level Manager tables, in this case the Threshold Table. The alias we used for this example is shown in Figure 198.

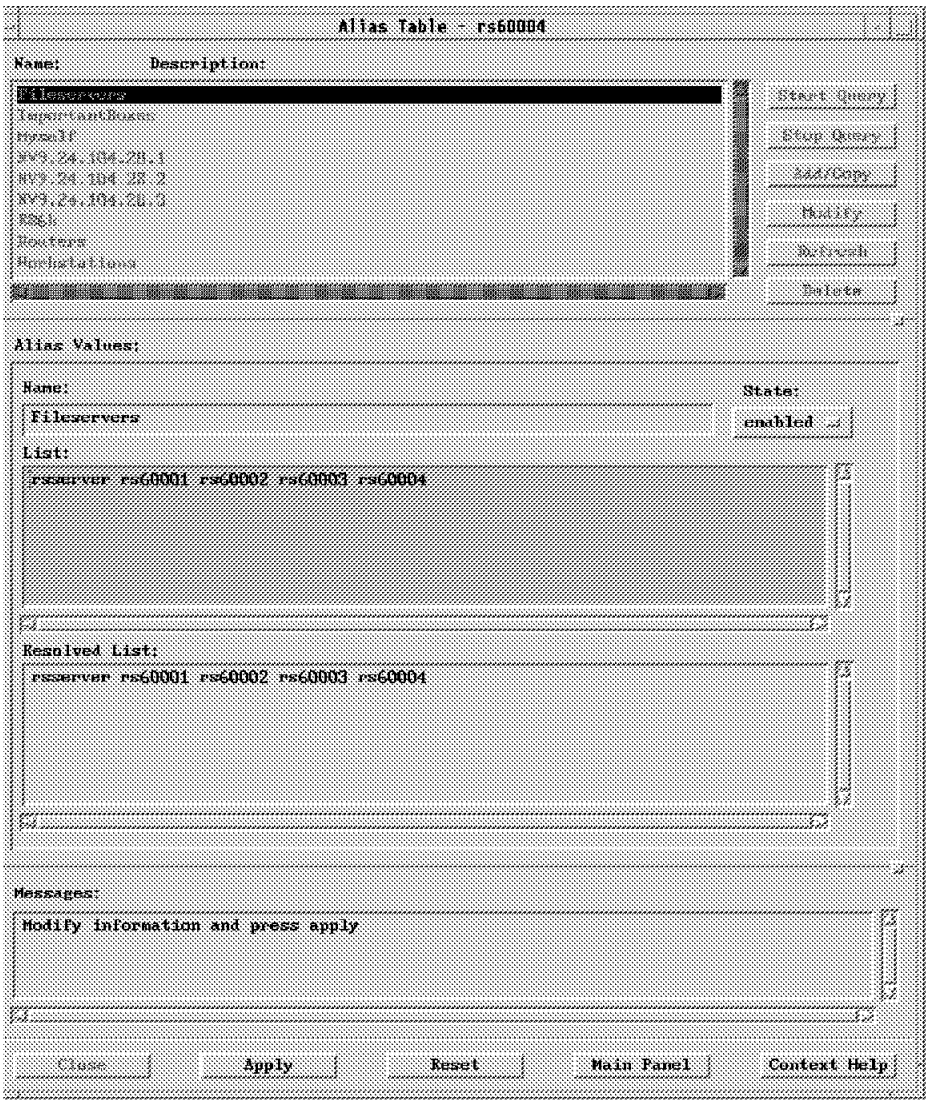


Figure 198. Entry in the Alias Table for the Alias "Fileservers"

The Name field corresponds to the name of the alias and the List field contains the list of nodes or IP addresses that are associated with that alias. These can be names or addresses, but when you enter them, the MLM verifies that they are acceptable and displays them in the Resolved List part of the panel.

Considerations if Using an SLM in Place of the MLM: The SLM does not provide an Alias Table, so this step would not apply to an SLM solution. You would have to install the SLM agent on each file server, instead.

7.3.1.2 Configuring the Trap Destination Table

We did not have to make any changes to this table, because when NetView for AIX discovered the Mid-Level Manager it defined a trap destination entry pointing to itself.

7.3.1.3 Threshold Table Entry

We added an entry to this table to monitor the paging space MIB variable on the group of file servers (defined in the alias group File servers). Figure 199 shows this entry.

Figure 199. Entry in the Threshold Table to Monitor Paging Space

The MIB variable that is being monitored (the Local/Remote MIB Variable field in the panel) is one of the standard SIA MIB variables. It reports the percentage of

paging space used. See page 167 for further explanation of how to specify the Local/Remote MIB Variable field. The values that can be seen in the Last Changed Session and Last Value fields should not be edited. These fields are filled in when you select **Start Query** and are the result of the SNMP GET that is sent to the specified node or group of nodes.

We are using the alias that we added in the Alias Table previously. It is not necessary to use an alias, the MIB variable can be specified alone, in which case the thresholding will only be done on the local node, that is the Mid-Level Manager. Alternatively a node name or IP address could be specified rather than an alias, to indicate on which node the thresholding is to be done.

When the percentage of paging space used exceeds 60% on any of the nodes that are being monitored, the threshold is exceeded and a trap will be sent to NetView for AIX. The configuration panel defining the threshold actions that we configured can be seen in Figure 200.

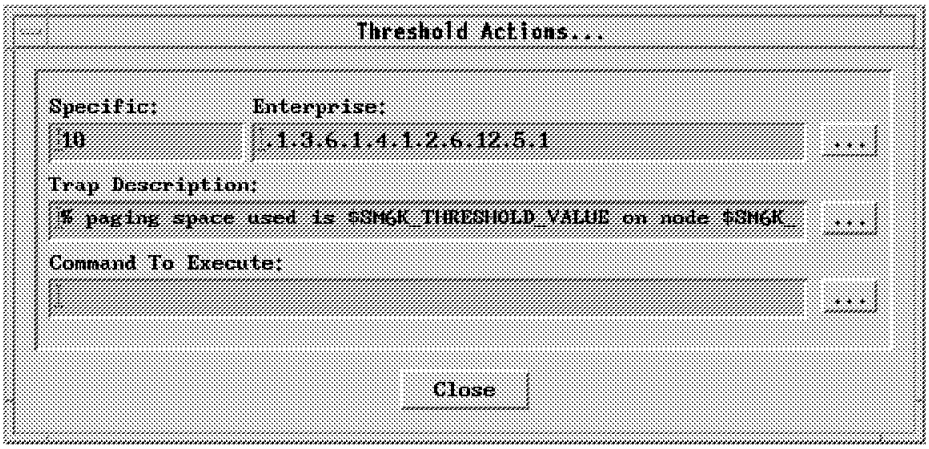


Figure 200. Threshold Actions for Paging Space Entry

You can see that we are sending a trap with an enterprise ID of .1.3.6.1.4.1.2.6.12.5.1 (the Systems Monitor Thresholds enterprise ID) and a specific trap ID of 10. The Trap Description field defines the first variable within the trap. In this case we are using some MLM environment variables to create the detailed description, see page 169 for a description of the most useful variables.

When the value drops below the rearm threshold of 60% a rearm trap will be sent. The configuration panel for the rearm action is shown in Figure 201 on page 224.

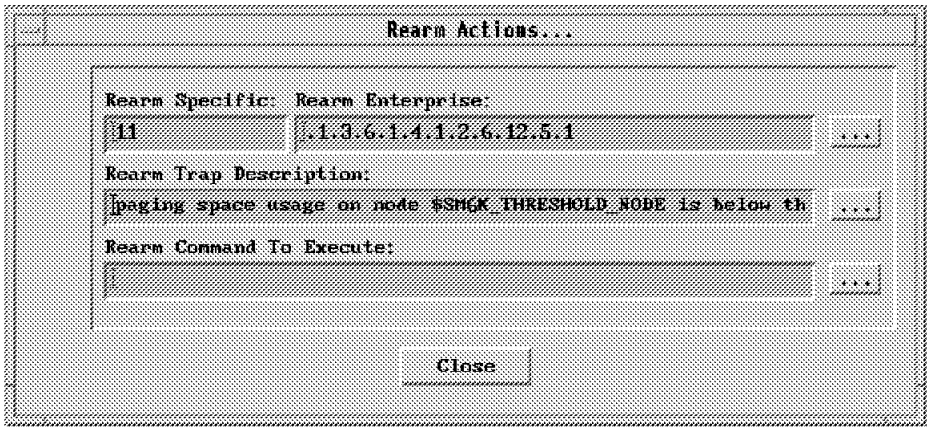


Figure 201. Rearm Actions for Paging Space Entry

Considerations if Using an SLM in Place of the MLM: This example would operate in the same way if an SLM was used instead of an MLM, except that the MIB variable to be monitored would not be prefixed with the alias name (Fileservers), because the SLM would have to be running on each node being monitored (it can only monitor the node it is on).

7.3.1.4 Paging Space Event Configuration

Once the threshold actions have been configured on the Mid-Level Manager, we have to add the traps in NetView for AIX event configuration. The paging space threshold event configuration is shown in Figure 202 on page 225.

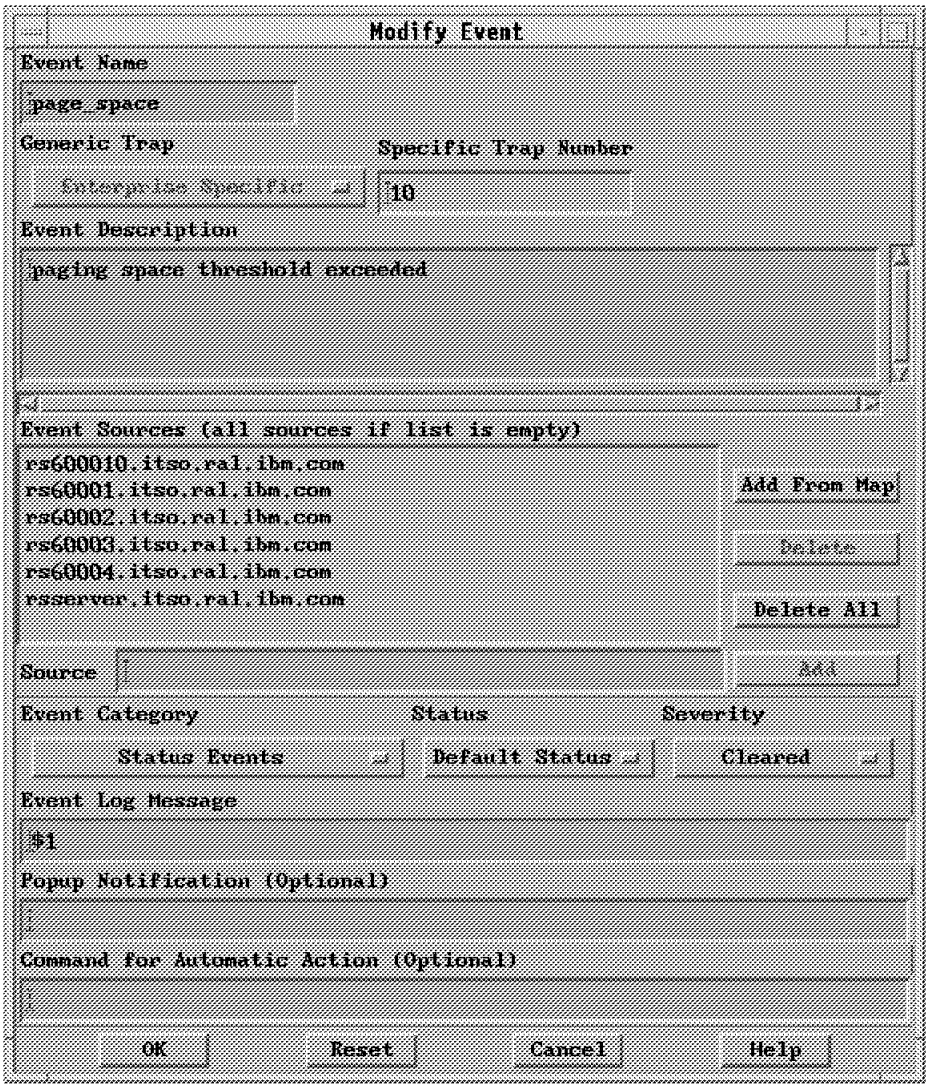


Figure 202. Paging Space Event Configuration

The Event Log Message field is set to $1, which will cause the first variable from the trap to be displayed on the event card. This will be the description that we specified in Figure 200 on page 223.

7.3.2 Monitoring Processor Utilization

It is also possible to store data as well as compare it against a threshold. The following example shows how to add an entry to the Threshold Table, to collect and threshold the processor load of an IBM 6611 Multiprotocol Router.

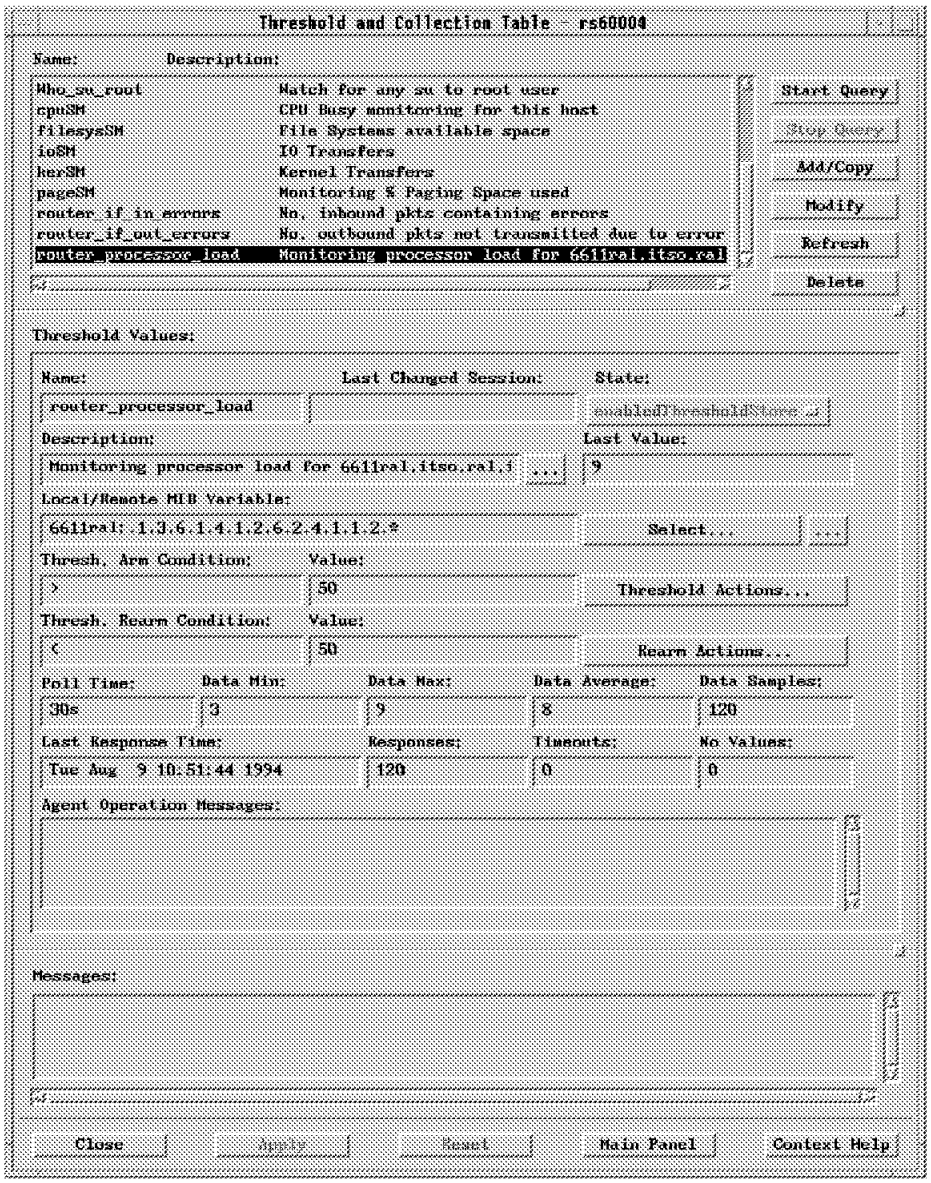


Figure 203. Threshold Table Entry for Router Processor Load

Threshold actions and rearm actions can be configured as described in the previous example, but in this example, the data is not only being compared against a threshold, but also being stored. This data can then be sent up to NetView for AIX and converted into ASCII format for input into spreadsheets or the NetView for AIX xnmgraph utility, or it can be converted into the appropriate format and fed into a relational database. 7.4, “Performance Data Collection” on page 253 shows a detailed example of this.

In this scenario, the MIB that is being queried is a 6611-specific MIB, but any MIB on any system in the network can be queried in exactly the same way. You can find the MIB object ID by pressing **Select** which starts the NetView for AIX MIB Browser. In order to do this you must first have loaded the MIB for the device you are monitoring into NetView for AIX, by selecting **Options** and then **Load/Unload MIBs** from the menu bar.

Considerations if Using an SLM in Place of the MLM: It is *not* possible to substitute an SLM for the MLM in this example, because the node being monitored is not an AIX system, but a 6611 router. The SLM can only monitor the node it is on.

7.3.3 Monitoring File System Utilization

Disk space in AIX is subdivided into file systems, giving a convenient way to allocate space. However, there is an immutable law of nature that states that no matter how big the file systems are made their contents keep growing. Monitoring file system utilization is therefore very important. A systems administrator will need to know how full file systems are becoming, so that they can delete unwanted log files or warn users to perform housekeeping before the file systems become completely full and cause serious problems.

In the following example we show two versions of a file system space monitor. In both cases we are using the MLM Threshold and Alias Tables to monitor for file system utilization above 95% on a group of nodes. The file system utilization information comes from the Systems Information Agent MIB, so each of the monitored nodes has to be running the SIA or the Systems Monitor V1 agent.

Both versions of this example will send a trap when the threshold value is breached. The difference between them is that the second example contains more detailed information within the trap.

7.3.3.1 Threshold Table Entry for Monitoring File Systems Filling Up

The configuration seen in Figure 204 on page 228 will monitor the percentage utilization of all file systems on nodes rs60002 and rs60004. The MIB variable that we are polling is one of the pre-defined variables included in the Systems Monitor for AIX MIB and the value of the variable is a number corresponding to the percentage of the file system used.

Figure 204. Threshold Table Entry for File Systems Filling Up

From this you can see that we have specified that we want to monitor all instances of MIB object .1.3.6.1.4.1.2.6.12.2.5.2.1.4 on the nodes in alias "RS6k". See page 167 for further explanation of how to specify the Local/Remote MIB Variable field. The MIB variables will be polled every one minute. We have set the threshold condition to be "greater than 95" and the rearm condition to be "less than 90". If any of the file systems on any of the monitored nodes exceeds 95% utilization, then a trap will be sent to NetView for AIX. We configured this by pressing **Threshold Actions** and filling in the panel shown in Figure 205 on page 229.

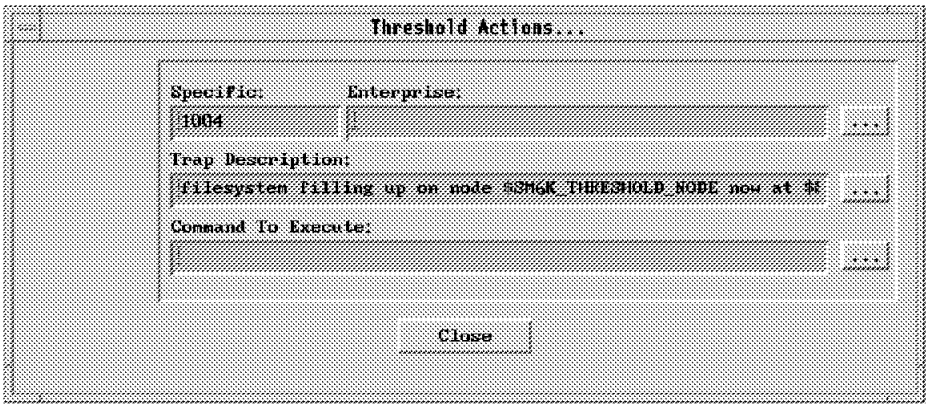


Figure 205. Threshold Actions for File Systems Filling Up

The trap we are sending will have the Systems Monitor Threshold enterprise ID, with generic trap ID 6 and specific trap ID 1004. The full text of the message we are sending in the trap is:

`filesystem filling up on node $SM6K_THRESHOLD_NODE now at $SM6K_THRESHOLD_VAR_VALUE % utilization`. We are making use of two of the variables provided by the Mid-Level Manager within this message:

- $SM6K_THRESHOLD_NODE resolves to the name of the node on which the threshold occurred (that is, the node where the file system exceeded 95%)
- $SM6K_THRESHOLD_VAR_VALUE is the value of the variable that exceeded the threshold (that is, the actual percentage utilization of the file system).

The event card that appears in NetView for AIX when this trap arrives is shown in Figure 206.

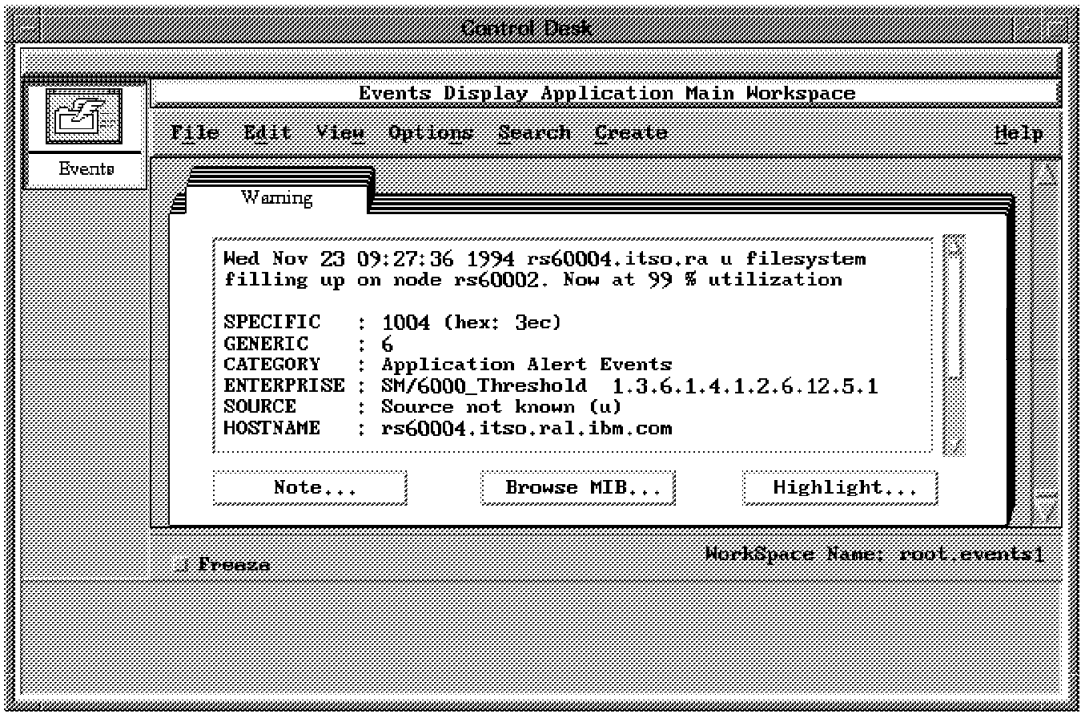


Figure 206. Event Card for File Systems Filling Up

We have also configured a rearm condition that will be activated when the utilization of the file system drops below 90%. The purpose of the rearm condition in this case is to prevent repeated threshold traps. For example, in this case we are monitoring two machines (rs60002 and rs60004, as specified by the Alias Table entry RS6k). If each of them had two file systems that were near full we would see four traps per minute.

On the other hand, when a rearm condition is specified, a trap will only be sent if a *different* file system is found to have exceeded the 95% threshold. No traps will be generated for file systems that were over the threshold at the previous poll. It is only when the rearm condition has been met that the threshold will be activated again for those file systems.

7.3.3.2 Enhancing File System Threshold Monitoring

The threshold configuration above (Figure 204 on page 228 and Figure 205 on page 229) achieves the objective of warning the NetView for AIX operator that a potential file system problem exists on a certain remote node. However it does not tell him *which* of the file systems is in trouble.

The operator can, of course, display the file system status for the node in question by selecting **Monitor**, then **System Information**, and then **Filesystems** from the menu bar. It would be better if we could include the file system name in the event card itself. We can derive the file system name by taking advantage of the way that the SIA constructs the MIB instance ID for the file systems table (see 2.3.2, “Understanding Systems Monitor MIB Instances” on page 33). The instance ID is in fact the name of the file system expressed as ASCII codes. So, for example, the MIB instance for the utilization of the /u file system would be .1.3.6.1.4.1.2.6.12.2.5.2.1.4.47.117 where the first fourteen digits are the MIB object ID for smSiaSystemFileSystemPercentUsed and 47.117 are the ASCII codes for ″/u″ in decimal.

This MIB instance is provided by the MLM in Threshold Table processing as a variable, SM6K_THRESHOLD_VAR_OID. We wrote a simple shell script, send_fs_trap, that converts the MIB object form into readable text and then sends it in a trap. send_fs_trap is listed in Figure 281 on page 316. The modified Threshold Actions panel that we used to invoke it is shown in Figure 207.

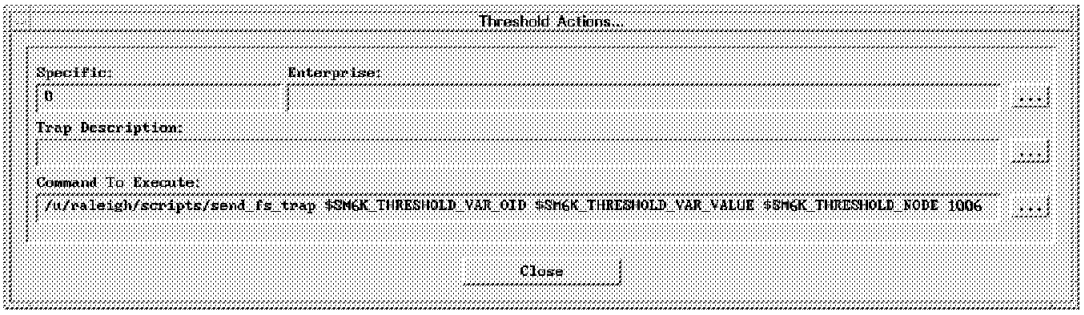


Figure 207. Threshold Actions Entry to Invoke send_fs_trap

You can see that we no longer generate a trap directly, but instead call send_fs_trap to do it. We are passing the MIB instance ID, the threshold value and the node name to the script, plus the specific trap number to use: 1006.

The event card that results when the trap generated by send_fs_trap arrives at NetView for AIX is shown in Figure 208 on page 231.

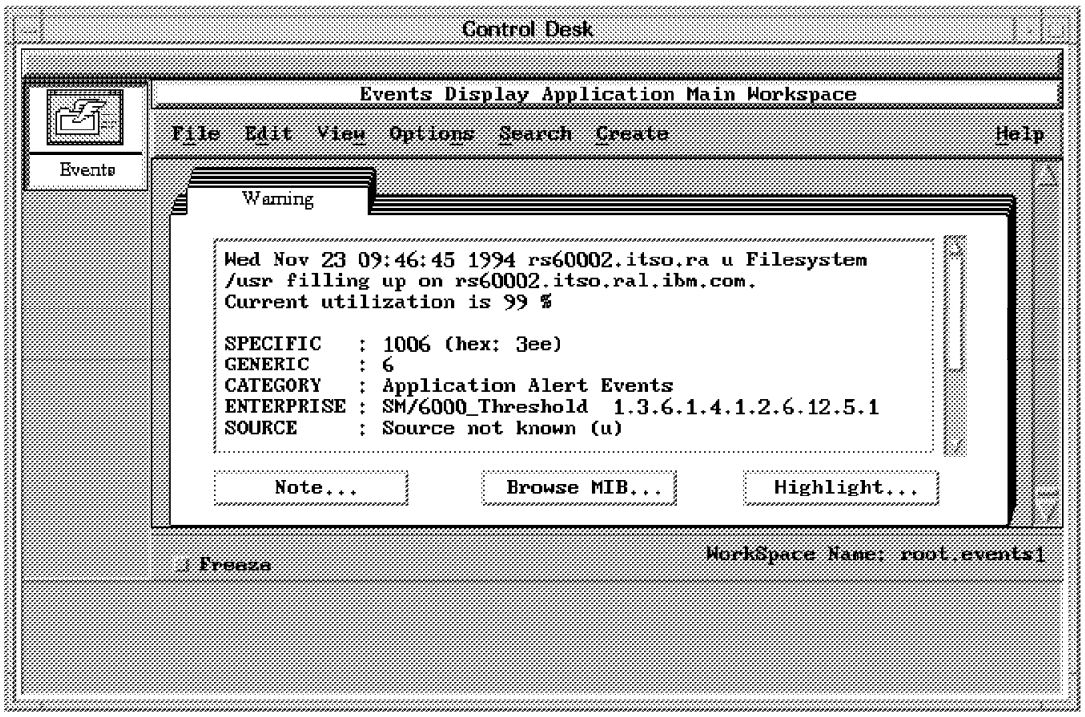


Figure 208. Event Card Including the File System Name. We had to customize the trap in NetView for AIX to make it format the trap in this way, since send_fs_trap only sends the raw data with no linking text.

Considerations if Using an SLM in Place of the MLM: This example would operate in the same way if an SLM was used instead of an MLM, except that the MIB variable to be monitored would not be prefixed with the alias name (RS6k), because the SLM would have to be running on each node being monitored (it can only monitor the node it is on).

7.3.4 Automatic Response to File System Full Errors

In the example above (7.3.3, “Monitoring File System Utilization” on page 227) we show how the MLM can poll for critical file system information and alert the NetView for AIX operator when a file system is getting dangerously full. In this example we want to go one step further so that, as well as informing the operator, the MLM will attempt to recover the situation automatically.

The command we want to issue is:

```
/usr/sbin/chfs -a size=+8192 /filesystem_name
```

This will increase the size of file system ″filesystem_name″ by 4 megabytes (AIX asks you to define space in units 512 KB, just to keep your mental arithmetic sharp).

The threshold polling is performed by the Mid-Level Manager node, but the file systems being monitored are on one or more Systems Information Agent nodes. We therefore have to issue the above command remotely. There are several options available to us for this:

- Use the rsh command

- Use the rexec command
- Use the SIA Command Table

We chose to do the last of these.

7.3.4.1 Creating and Invoking the Command Table Using SNMP SET

We created the Command Table entry on each of the SIA nodes (see Figure 209).

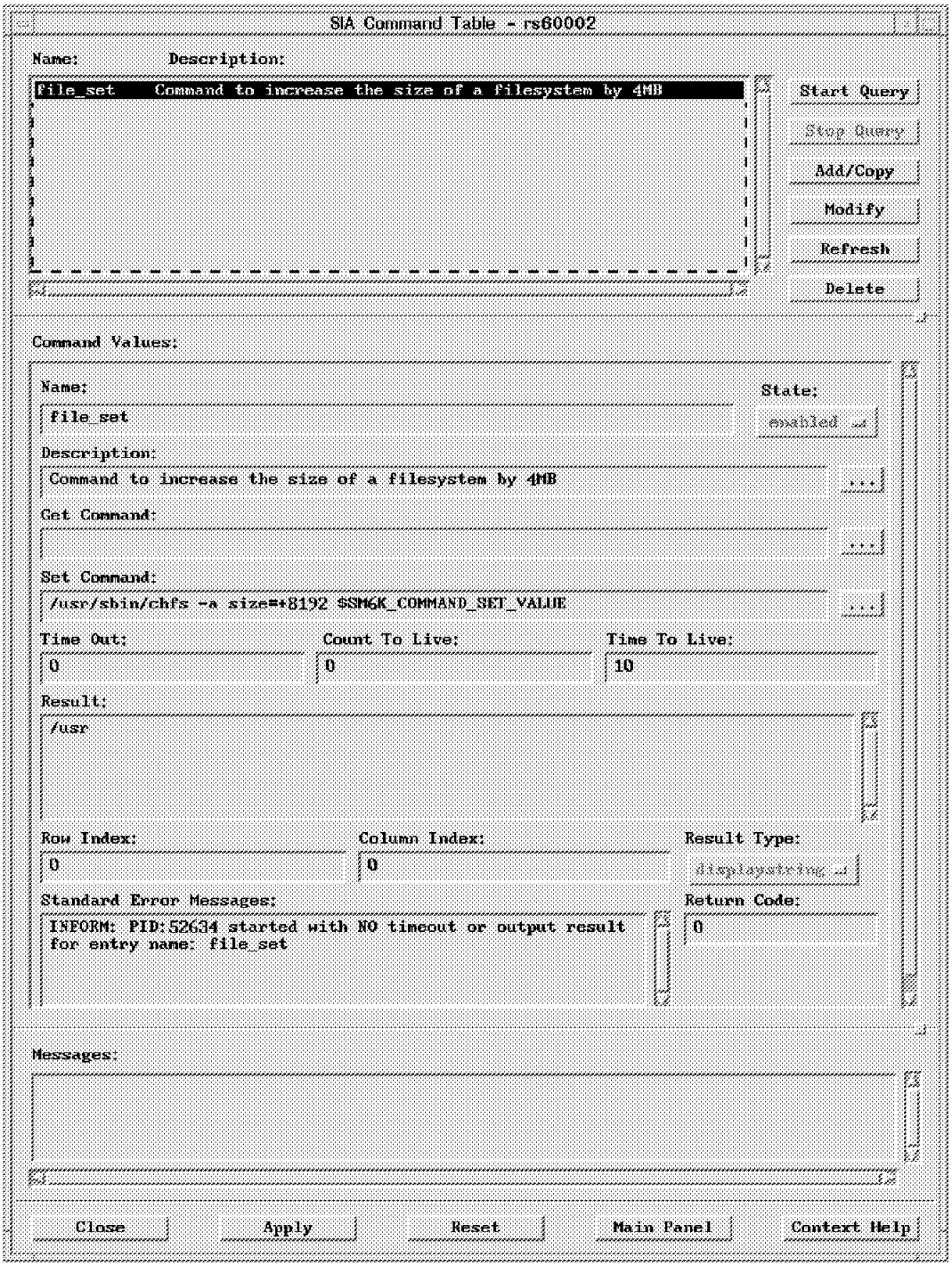


Figure 209. Command Table Entry for File System Space Increase

From Figure 209 you can see that we have filled in the Set Command field, instead of the more usual Get Command. This is because we need to pass information (the name of the file system to be expanded) to the SIA. We can specify the command parameters, by issuing a SET request to the DisplayStringResult, IntegerResult, CounterResult or GaugeResult MIB variables for the particular entry in the Command Table that we want to run. In the above example, the information we will pass is a string and therefore we must issue an SNMP SET command to the DisplayStringResult field for the entry with name file_set.

If we were doing this from the NetView for AIX machine we could use the snmpset command. Since we want to send the command from the MLM we have to use the standard AIX command: snmpinfo. The command to update file system /tmp on node rs60004 would be:

```
snmpinfo -m set -c ITSC -h rs60004 1.3.6.1.4.1.2.6.12.4.1.1.13.102.105.108.101.95.115.101.116=/tmp
```

Important

Do not put a leading . in front of the MIB object ID, as this will cause the command to fail and return an error message ″unknown variable″.

The general format of the snmpinfo -m set command is:

```
snmpinfo -m set -c community -h hostname variable.instance=value
```

In this case the variable is the SIA Command Table MIB object for DisplayStringResult, and the instance is the ID of the entry in the Command Table. The entry is called file_set, so its instance ID is 102.105.108.101.95.115.101.116 (″file_set″ in decimal ASCII codes). The value we want to set this field to is the name of the file system we want to expand, /tmp in our example above. This value is then made available within the Command Table as variable SM6K_COMMAND_SET_VALUE (see Figure 209 on page 232).

Before we can issue this command, we have to configure the MLM system to allow us to use snmpinfo in this way. The command uses file /etc/mib.defs to define the MIB objects that it can GET or SET. This file is created using the mosy command. mosy is a MIB compiler, that takes ASN.1 source files and converts them into an intermediate form that a program can more easily use. We issued the following commands to generate a copy of /etc/mib.defs with the Systems Information Agent MIB objects included:

```
cd /usr/lpp/snmpd
mosy -o /etc/mib.defs smi.my mibII.my ibm.my /usr/lpp/smsia/original/ibm-sysinfo.mib
```

7.3.4.2 Threshold Table Entry to Invoke the Command Table Using SNMP SET

Now we must add the Threshold Table entry to poll for excessive file system utilization and automatically invoke the Command Table entry we have just created.

The Threshold Table entry is identical to the one we used previously (Figure 204 on page 228), with two exceptions:

1. The Threshold Actions details are different.
2. We removed the rearm specification.

The new Threshold Actions panel is shown in Figure 210 on page 234.

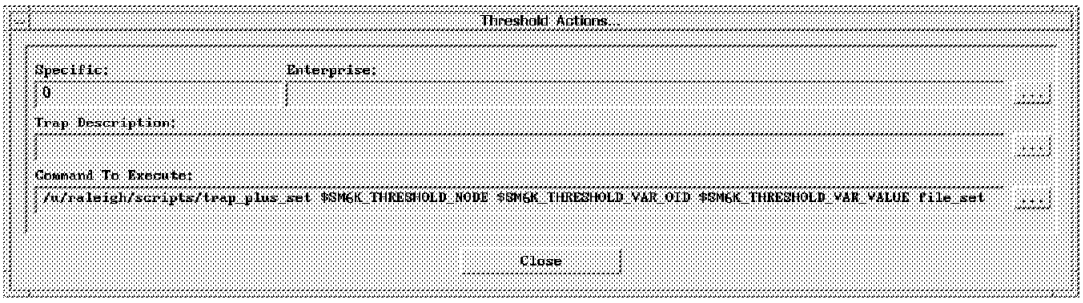


Figure 210. Command Table Entry for File System Space Increase

We are executing shell script trap_plus_set, which uses the same trick as fs_send_trap to find the name of the file system in trouble (see 7.3.3.2, "Enhancing File System Threshold Monitoring" on page 230). It then sends a trap to inform the NetView for AIX operator of the problem and issues the `snmpinfo` command to invoke the file system space increase. trap_plus_set is shown in Figure 282 on page 317.

Why did we remove the threshold rearm specification? The reason is that we want our automated response to always fix the problem. It would automatically add 4 MB to any file system that triggers the threshold at 95% full. However, if adding 4 MB did not reduce the utilization below the rearm level the threshold would not reoccur and hence the automation would be disabled.

Figure 211 shows the event display in NetView for AIX when the threshold-driven automation is in action.

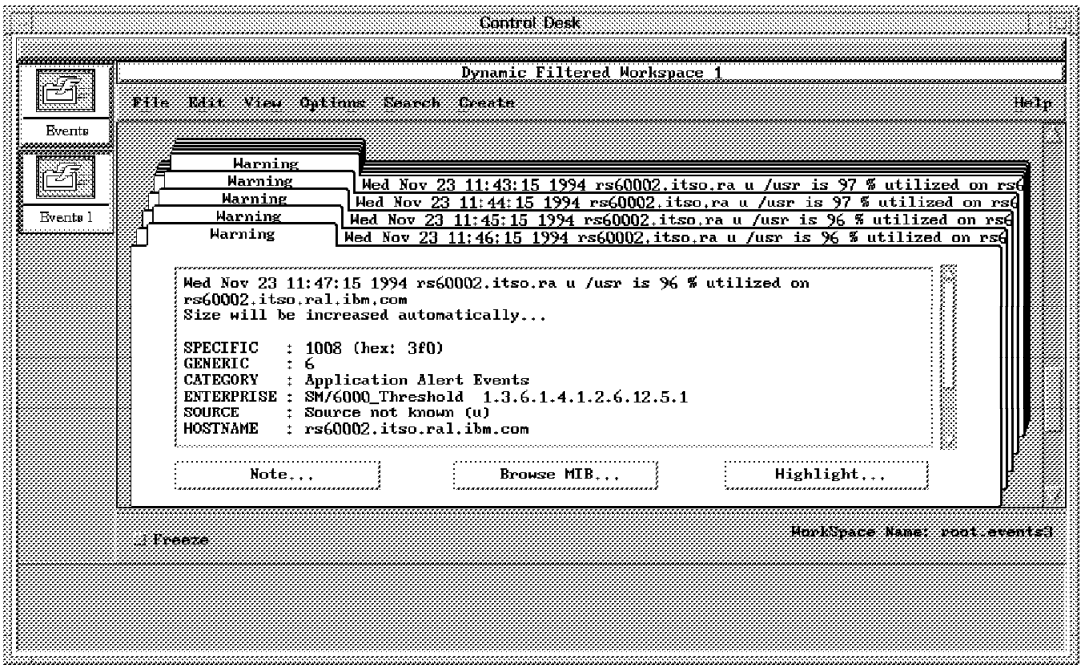


Figure 211. Events Display Showing File System Space Automation. The /usr file system is very large, so adding 4 MB did not reduce it below the 95% threshold value in one step. You can see that at each polling interval an extra 4 MB is added until utilization falls below the threshold.

Considerations if Using an SLM in Place of the MLM: This example would be much simpler if using the SLM in place of the MLM for monitoring, because there would be no need to remotely execute the chfs command. The SLM agent would be running on the node being monitored, so the command could be executed directly. This means that the Command Table entry and the automatic invocation of snmpinfo to trigger it are both unnecessary.

7.3.5 Representing Systems Monitor Thresholds As NetView Symbols

All of the performance monitoring examples so far have resulted in traps being sent to NetView for AIX, displayed as event cards. Another way to show them is to exploit the network mapping capability of NetView for AIX to update the color of symbols on the graphical views.

The Agent Policy Manager (APM) feature of NetView for AIX Version 4 automatically adds symbols to the node submap which reflect the status of Systems Monitor for AIX threshold and file monitors. You can see examples of this in Chapter 9, “Examples Using APM for Threshold and File Monitoring Tasks” on page 279.

Using APM is the recommended way (and certainly the easiest way) to get symbol color changes as a result of threshold monitoring. However, if you are not using NetView for AIX Version 4, or if you have a requirement that APM does not meet, this example may be useful.

In this example we use some sample code that has been produced in other ITSO Raleigh projects. This code, named wteuiap6, is described in *Examples Using NetView for AIX*, GG24-4327, and is further enhanced in *Examples Using NetView for AIX Version 4*, SG24-4515. The source code is available from IBM, or by anonymous FTP from node rsserver.itso.ral.ibm.com for IBM personnel.

wteuiap6 is a daemon connected to the NetView for AIX end-user interface API, which waits for messages from a command-line program, wtdriver6. The commands you can enter allow you to add objects, symbols and submaps, connect them together, and set their status.

In the following example we show how to add an icon to represent the file system utilization on node rs60002 and cause this icon to change color to yellow, representing marginal status, when the file system utilization exceeds 95% and change color to red, representing a critical condition, when the file system utilization exceeds 99%.

Initially the icon representing the file systems, must be added to the rs60002 submap. This can be done by issuing the command:

```
wtdriver6 -f rs60002 add rs60002.filesystem "Client:FileSystem" label file system
```

The format of this command is:

```
wtdriver6 -f submap add symbol_name symbol_type label label_name
```

Where:

- submap is the name of the submap the icon is being added to.
- symbol_name is the name of the symbol being added.
- symbol_type is the type of symbol being added.

If the symbol type is more than one word in length, it must be enclosed in double quotes.

- label_name is the symbol label.

Symbol Types

You can click on **Help**, and then **View Legend** to list the symbol types that are available.

Figure 212 shows the rs60002 submap after the file system symbol was added.

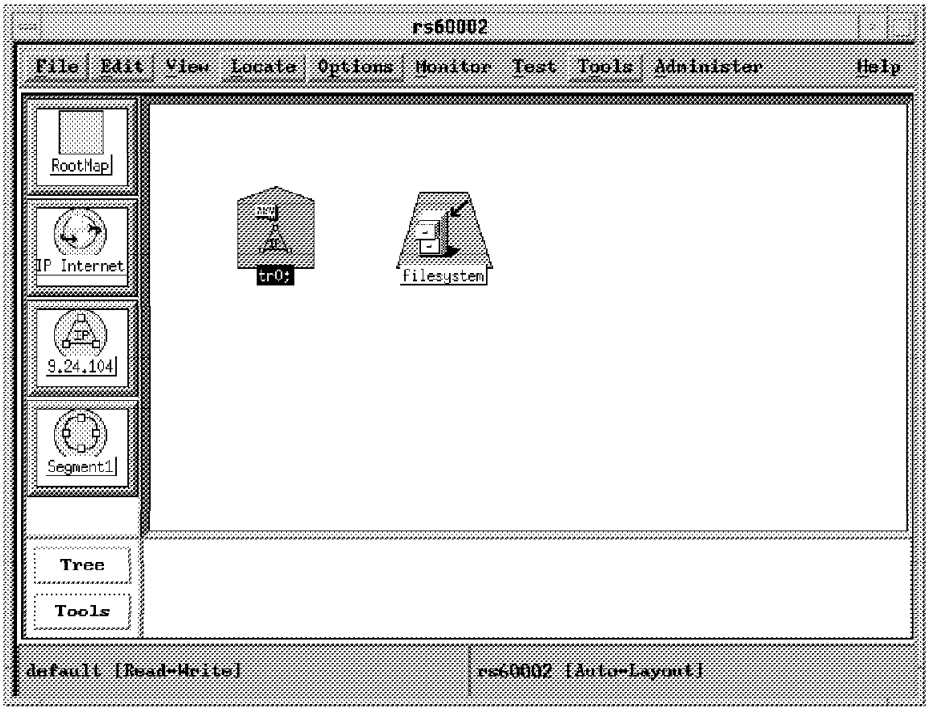


Figure 212. rs60002 Submap with File System Symbol

ipmap, the NetView for AIX daemon responsible for displaying and updating the IP views, will not be able to determine the status of this new symbol because it is not an IP addressable device. Therefore it will consider the icon to be in an unknown state and set it initially blue.

We can use the wtdriver6 command to set the status of the rs60002.file system object, and therefore change the color of the icon, as follows:

```
wtdriver6 -f rs60002 set rs60002.filesystem up
```

This command will change the status color of the filesystem symbol from blue to green. The status options are: critical, marginal, unknown, up, down, user1, user2 and unmanaged.

In order to change the color of the icon when the file system utilization exceeds a specified threshold on node rs60002, we need to configure the appropriate NetView for AIX events. The event that is generated when the file system utilization exceeds 95% has a specific trap number of 1004 and is a Systems Monitor Threshold event. We therefore need to configure this event, as shown in Figure 213 on page 237.

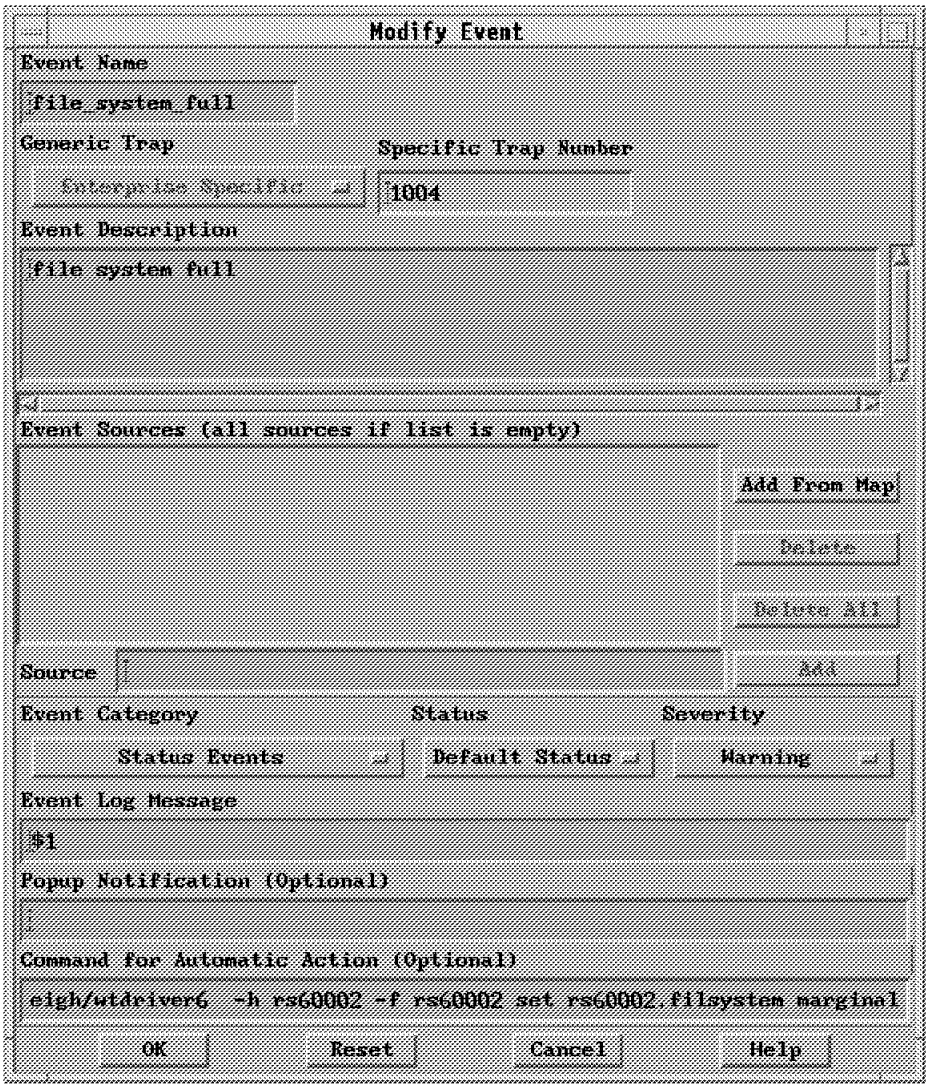


Figure 213. Event Configuration to Change File System Symbol Yellow

It is best to specify the full path name for the wtdriver6 command, when adding it to the Command For Automatic Action field. This configuration will cause the file systems symbol in the rs60002 submap to change yellow, whenever trap 1004 is sent to NetView for AIX from Systems Monitor for AIX, as shown in Figure 214 on page 238.

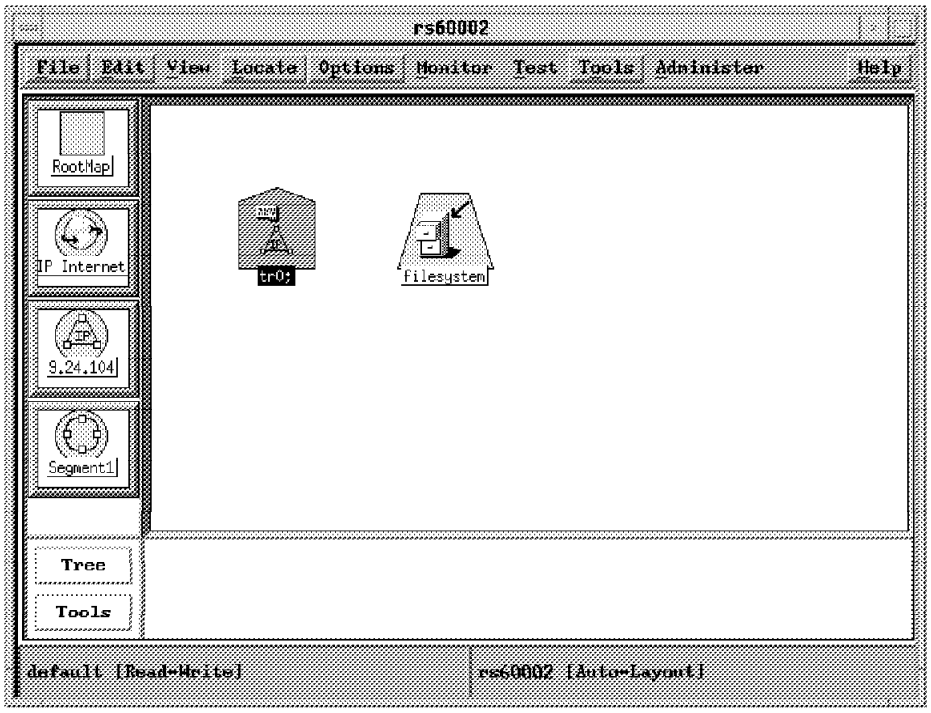


Figure 214. rs60002 Submap Showing Color Change of File System Symbol. OK...so this is book is not in color! The lighter shade of gray of the symbol on the right shows that it is yellow, instead of green.

Similarly, we could define a further file system monitor to check for utilization in excess of 99%. The event configuration for this would be the same as for trap 1004, shown in Figure 213 on page 237 except with the following command being used in the Command for Automatic Execution field:

```
/usr/OV/raleigh/wtdriver6 -f rs60002 set rs60002.filesystem critical
```

These color changes will be seen on the rs60002 submap (that is, the submap that is opened as a result of double-clicking on the node symbol for rs60002). Ideally we want these color changes to be propagated to the higher map levels, so that status changes are represented at the segment and network levels. To allow this, you have to modify the NetView for AIX map configuration so that the status of nodes is not measured *only* on IP. You do this by selecting **Edit**, then **Modify/Describe**, and then **Map** from the NetView for AIX menu bar to bring up the Map Description panel shown in Figure 215 on page 239.

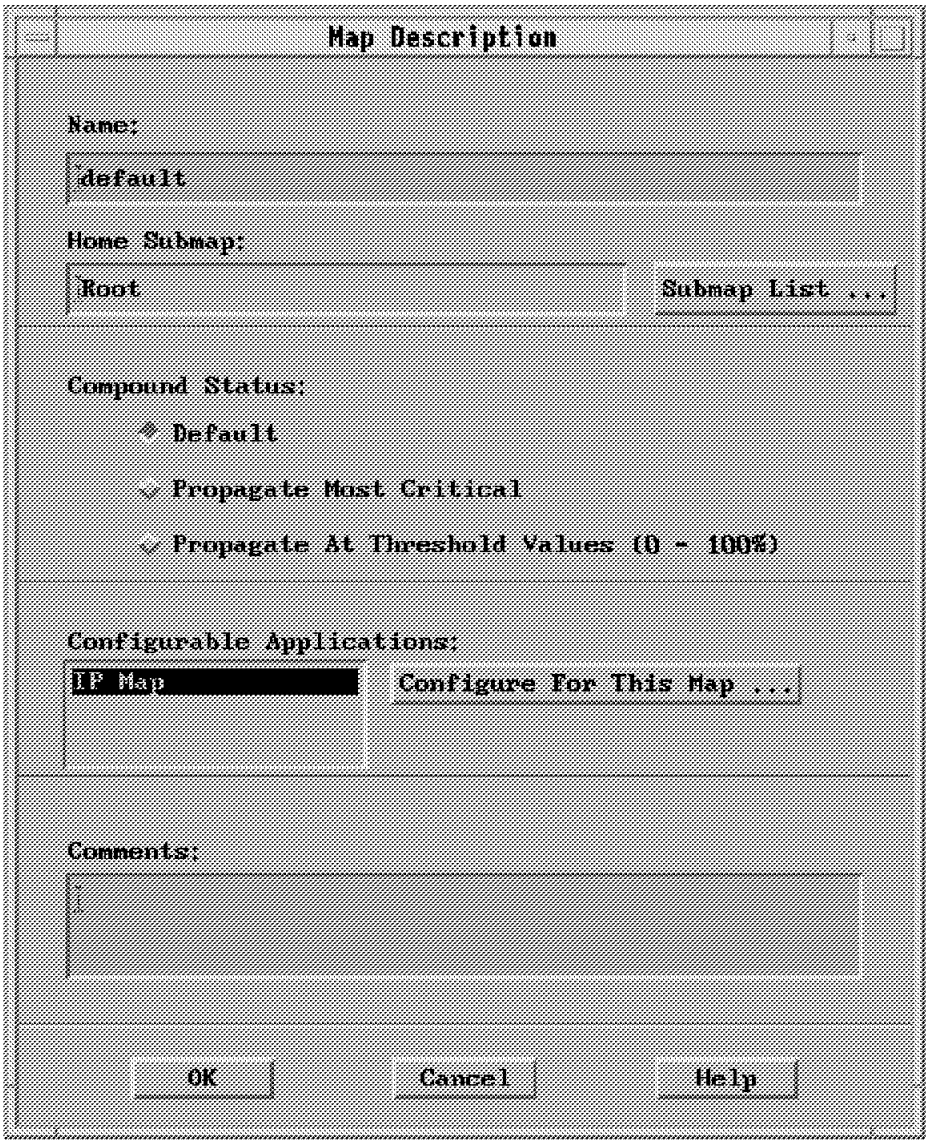


Figure 215. NetView for AIX Map Description Panel

Selecting **IP Map** and clicking on button **Configure For This Map...** will take you to the map configuration panel shown in Figure 216 on page 240.

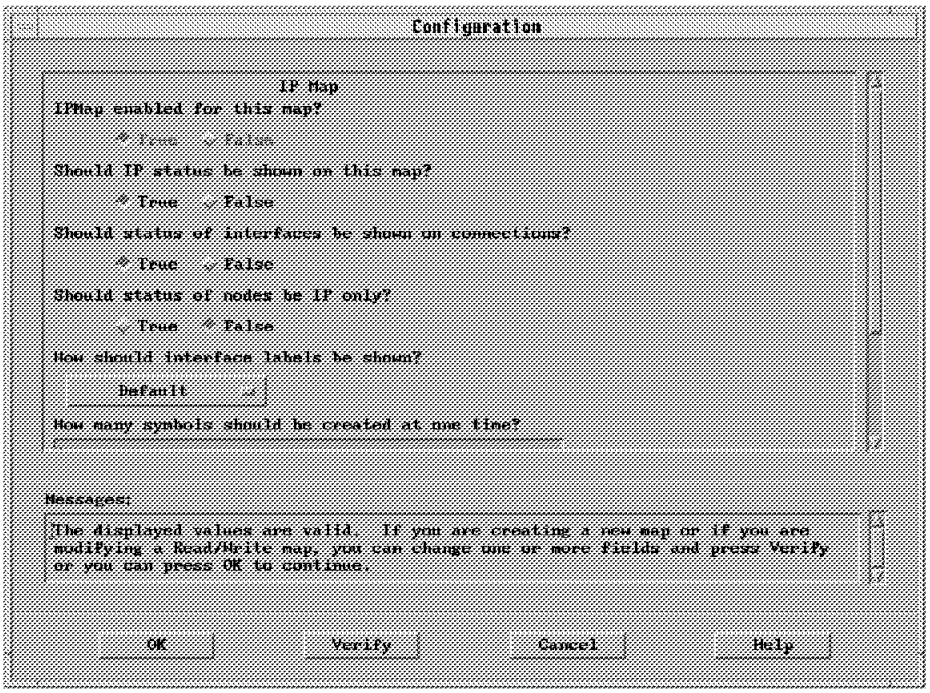


Figure 216. NetView for AIX Map Configuration Panel

Set Should status of nodes be IP only? to False and then select **Verify**.

7.3.5.1 Making the Symbol Executable

It is also possible to make the new symbol that we have created executable. This means that the operator may double-click on the icon to execute one of the NetView for AIX menu options automatically. In this case we decided that we wanted a double-click to display the file system summary for the node.

We achieved this by selecting the symbol and then pressing the right-hand mouse button and selecting **Edit**, then **Modify/Describe**, and then **Symbol** from the pop-up menu. The window that is displayed is shown in Figure 217 on page 241.

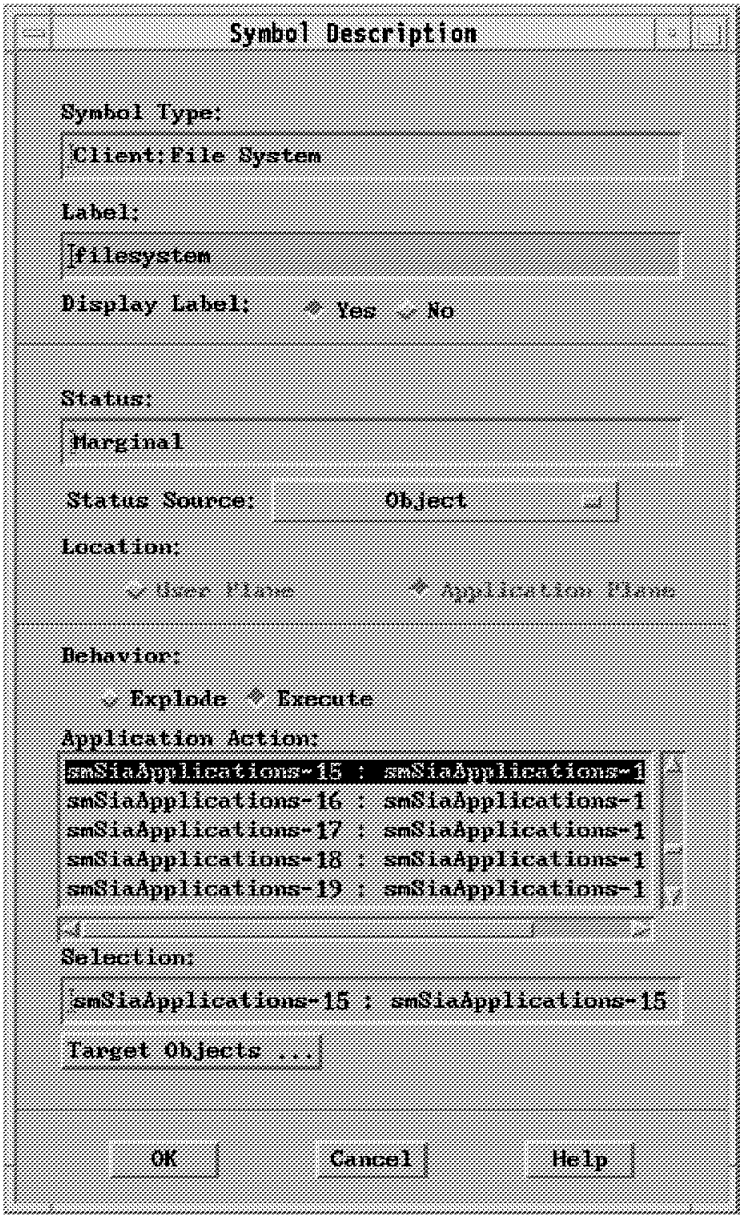


Figure 217. Symbol Description Window

We then followed the following procedure:

1. Set Behavior to Execute.
2. Select an application to execute from the application actions window.

   The Application Action window has a list of applications that can be executed when the symbol is executed. These are all of the options that are found within the NetView for AIX menus. All such menu entries are defined in files under the /usr/OV/registration/C/ directory. In 7.1.1, “Monitoring Printer Status” on page 162 we showed an example of adding a menu entry in this way.

   The application we selected, called smSiaApplications-15, lists file system utilization.

3. Click on target objects and enter the name of the object on which you want the application to execute. In this case it is rs60002 since this is where the SIA is running, to which we will send our file system query command.

7.3.6 Monitoring Excessive CPU Utilization for Processes

In 7.3.2, “Monitoring Processor Utilization” on page 225 we showed an example of setting thresholds based on the total utilization of a system. In that case we were monitoring a 6611, but we could equally well monitor the RISC System/6000 utilization figures provided in the Systems Information Agent MIB. However, often the *total* CPU utilization is not as interesting as the utilization of particular processes within the system. In this example, we use the Mid-Level Manager to monitor a remote node for processes exceeding 10% CPU. The Threshold Table entry we defined for this is shown in Figure 218 on page 243.

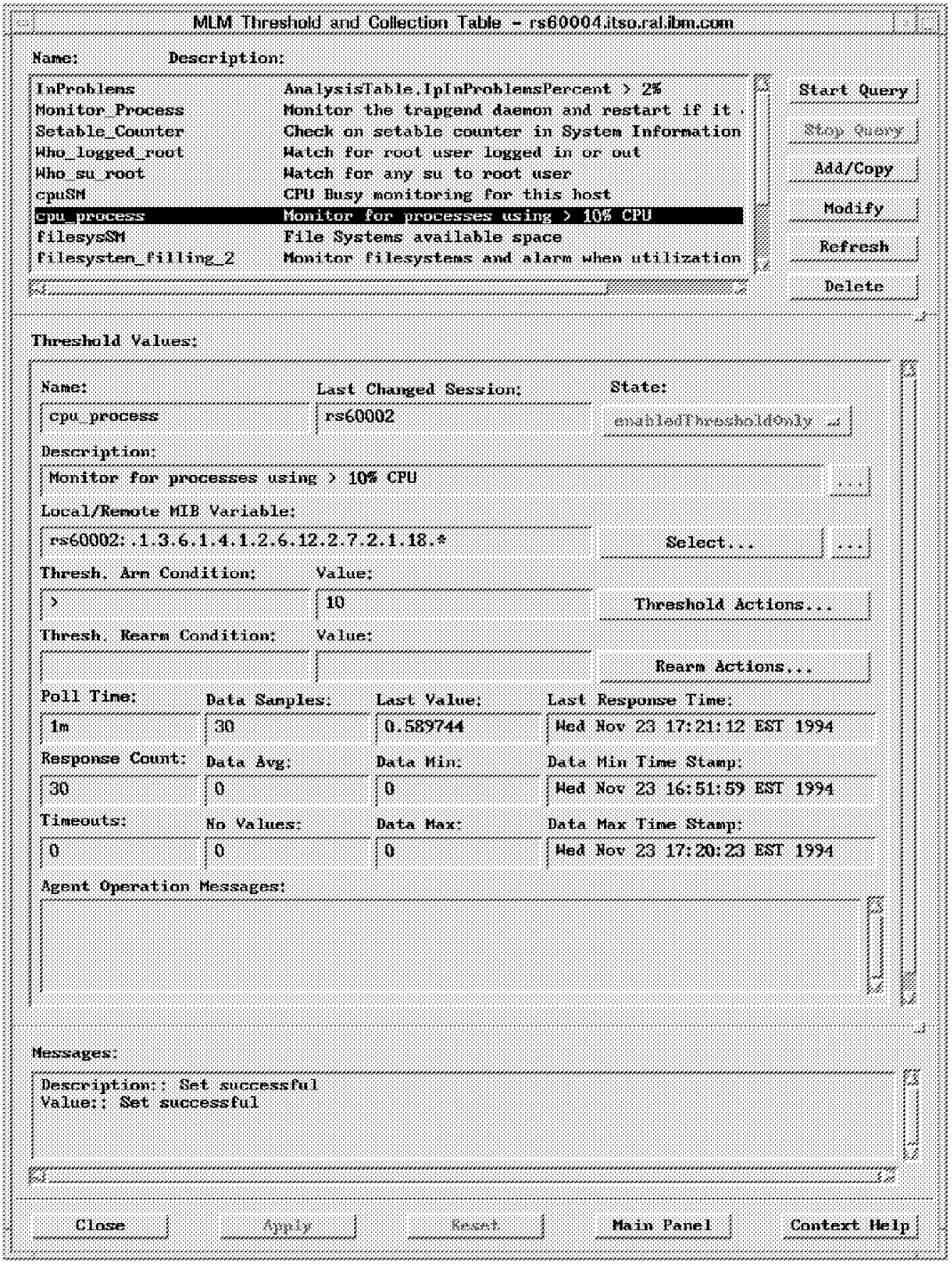


Figure 218. Threshold Table Entry for CPU Monitoring per Process

The target node, rs60002, has the SIA running and the MIB object that we are polling is .1.3.6.1.4.1.2.6.12.2.7.2.1.18 which is smSiaSystemProcessCPU from the SIA MIB. We have specified the instance ID as "*", which means "all instances". See page 167 for further explanation of how to specify the Local/Remote MIB Variable field.

Defining the Threshold Actions for this monitor gives us the same problem that we described in 7.3.3.2, "Enhancing File System Threshold Monitoring" on page 230. That is, we want to report the name of the process that is using

excessive CPU in the trap that we send to NetView for AIX, but we only have it in dotted decimal, as the MIB instance ID. The solution is to create a shell script to convert the instance ID into a text string, in exactly the same way as before. The script we used, send_ps_trap, is exactly the same as send_fs_trap (see Figure 281 on page 316) except for the MIB object ID from which the instance ID is extracted. The Threshold Actions panel for this example is shown in Figure 219.

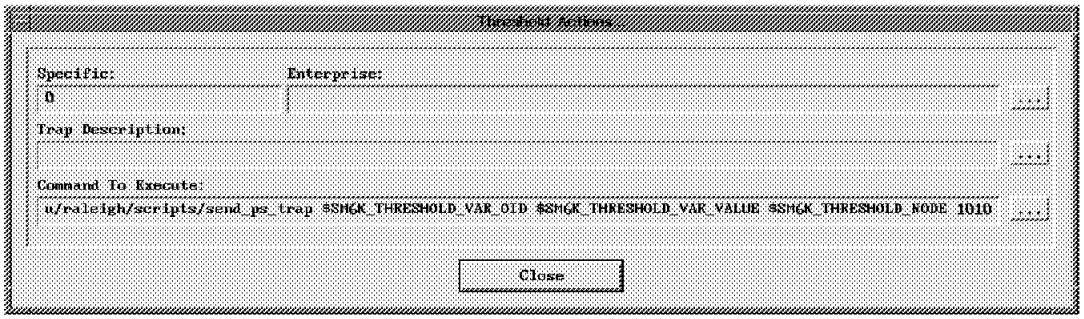


Figure 219. Defining Threshold Actions for CPU per Process Monitor

The NetView for AIX events display generated by this monitor is shown in Figure 220.

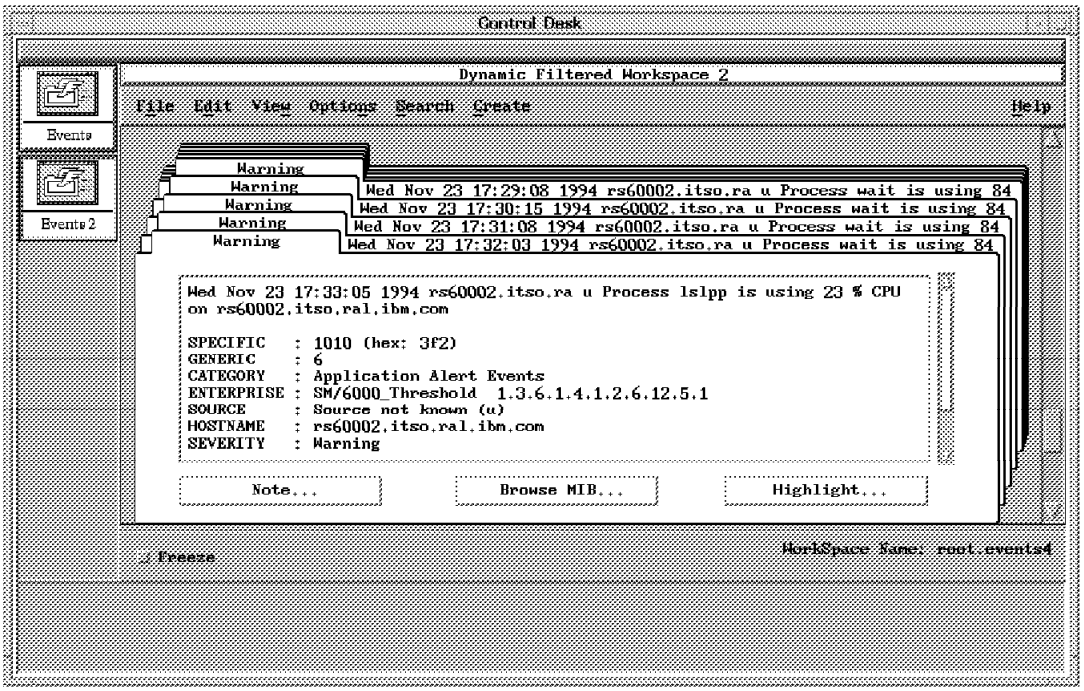


Figure 220. Events Created from CPU Monitoring per Process Threshold

If you examine the event cards in Figure 220, you will see that the monitor is not working exactly as we would like. It has correctly highlighted that process lslpp is using a lot of CPU in the latest polling interval. However, in previous intervals when no process was too busy the monitor shows a process called ″wait″ using 84% CPU. This is an internal function of AIX, which uses spare CPU cycles to perform system housekeeping. High utilization by wait is an indication of a healthy, under-loaded system. We certainly do not want a warning that it is happening.

To circumvent the false reading for the wait task, we decided to filter it out. There are several places we could do this:

1. In the shell script that generates the trap
2. Using the MLM Filter Table
3. Using NetView for AIX event configuration

We chose to add an MLM filter (see Figure 221).

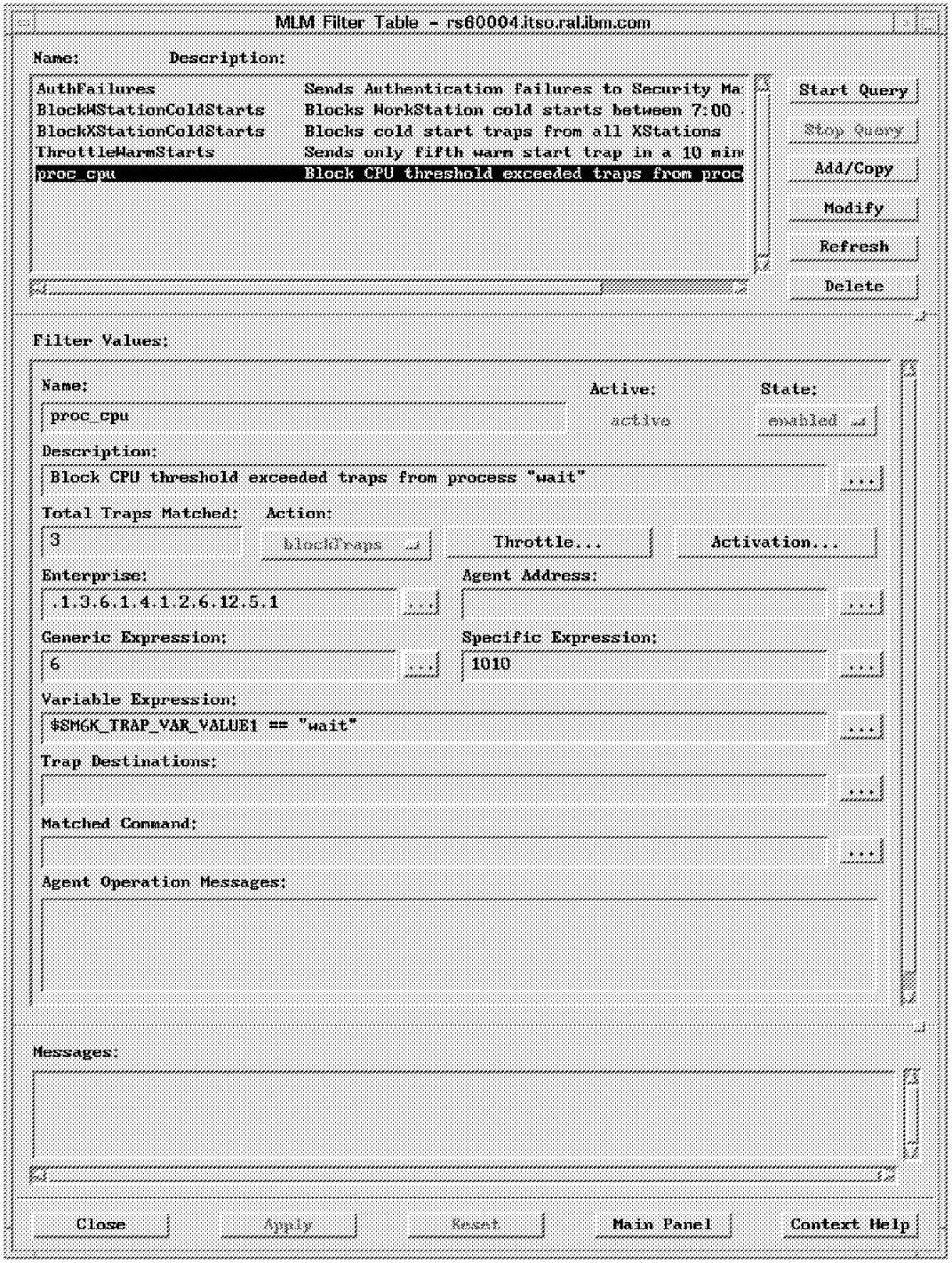


Figure 221. Filter Table to Remove Unwanted CPU Threshold Events

This Filter Table entry works because the send_ps_trap shell script has localhost defined as its trap target. This means that the traps will be handled by the MLM, filtered by the Filter Table and passed on to any active managers defined in the Trap Destination Table. In the entry we are filtering on the Enterprise/Generic/Specific trap combination for trap 1010, as created by send_ps_trap. However we have further refined the filter to check for a value of wait in the first variable field within the trap. The action that we have specified if these criteria are met is blockTraps, so the trap will be discarded.

Considerations if Using an SLM in Place of the MLM: This example would operate in the same way if an SLM was used instead of an MLM, except that the MIB variable to be monitored would not be prefixed with the host name (rs60002), because the SLM would have to be running on rs60002 in order to be able to monitor it.

7.3.7 Monitoring the Percentage of IP Datagrams in Error

In this example we show the operation of the MLM Analysis Table, to combine MIB values together in a single MIB variable. The value we want to know is the percentage of IP datagrams passing through the system that are in error.

The Analysis Table, like the Command Table, can be used to extend the Systems Monitor MIB. It is used to perform arithmetic expressions on MIB variables, the result of which can themselves be accessed as MIB variables. A detailed explanation of the various expression operators that are available and how expressions are formed is in *Systems Monitor for AIX User's Guide*, SC31-7150.

The source data for our Analysis Table entry is all from MIB-II:

ipInHdrErrors The number of input datagrams discarded due to errors in their IP headers.

ipInAddrErrors The number of input datagrams discarded because the IP address in their IP header's destination field, was not a valid address.

ipInUnknownProtos The number of datagrams discarded because of an unknown or unsupported protocol.

ipInDiscards The number of datagrams for which no problems were encountered to prevent them from being processed, but were discarded for other reasons, for example lack of buffer space.

Adding all these values together will result in the total number of IP error packets. Figure 222 on page 247 shows the Analysis Table entry to add these variables together.

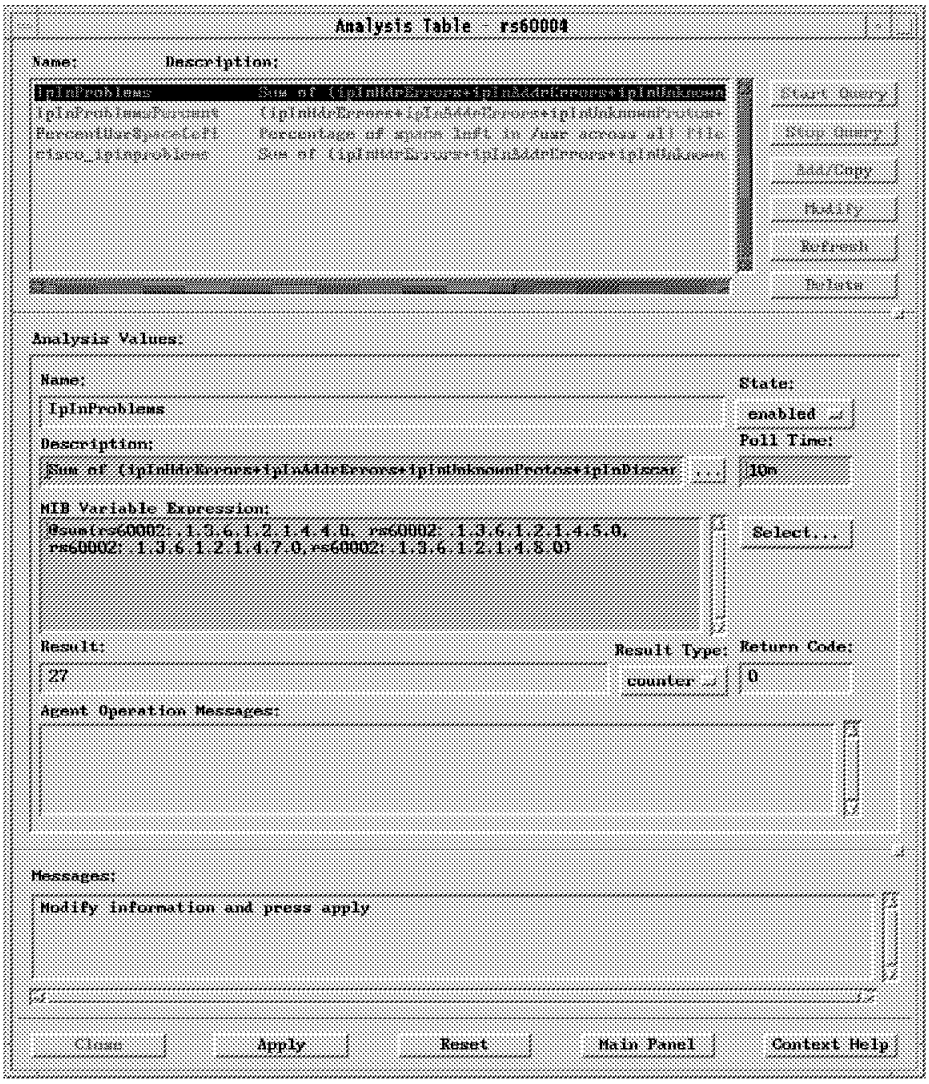


Figure 222. Analysis Table Entry for Measuring Total IP Input Errors

We can then take this a stage further, to calculate the percentage of IP incoming datagrams that have errors. We need to divide the sum of the errors (above) by the total number of input datagrams received (MIB variable `ipInReceives`) as shown in Figure 223 on page 248.

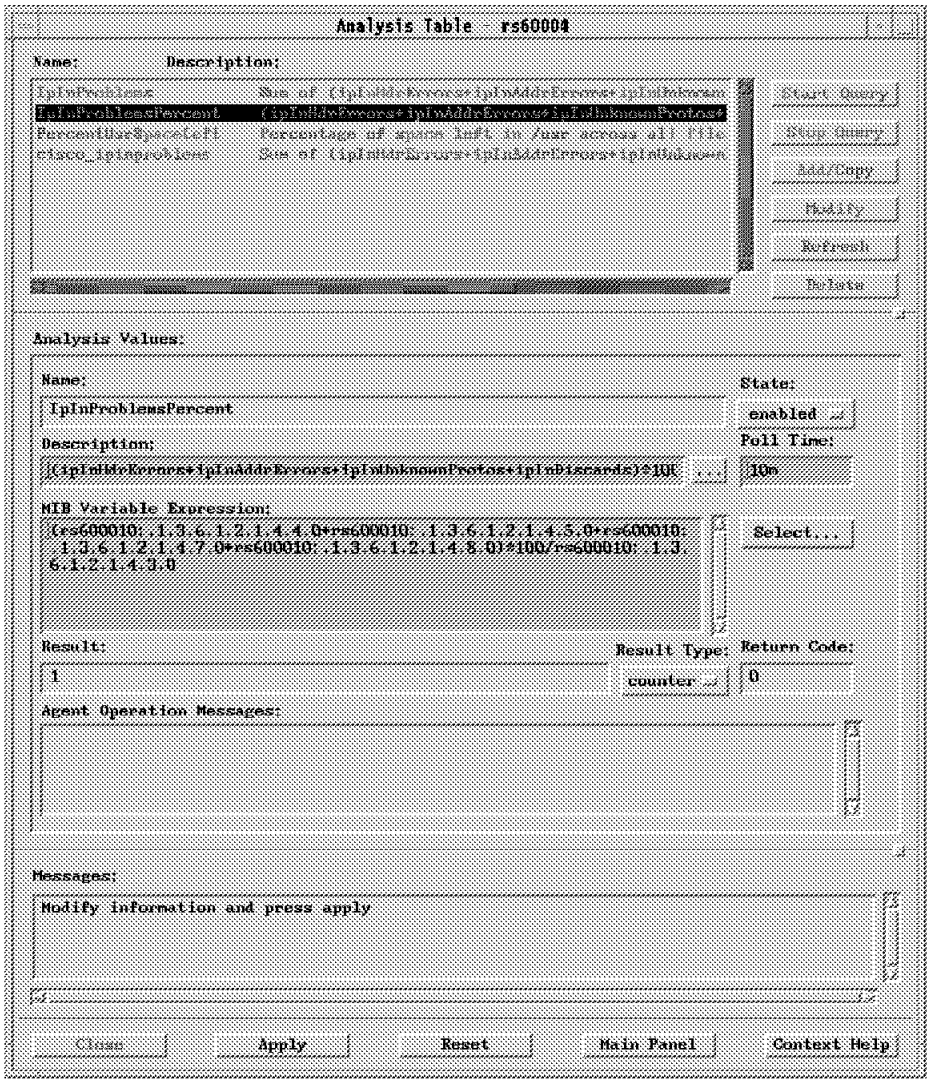


Figure 223. Analysis Table Entry for Measuring Percentage of IP Packets in Errors

The analysis can be performed on an individual host, as shown in the examples above, or on a group of nodes by using an alias.

Note

When an alias is used, the MIB variables will resolve to multiple values. In this situation, *all* resolved values are averaged before use in the expression.

These MIB variables can also be used as input to other tables, for example the Threshold Table, in exactly the same way as any other MIB variable.

The Threshold Table entry shown in Figure 224 on page 249, stores and thresholds the data value for the percentage of IP errors on rs600010. From the Analysis Table entry it can be seen that the value that is output after the arithmetic expression has been applied is a counter. Therefore we want to monitor the MIB variable that corresponds to the counter result of the Analysis Table. The object ID for this variable is:

.1.3.6.1.4.1.2.6.12.6.1.1.9

However, we only want to monitor the counter result for one of the Analysis Table entries. and not all and therefore we must specify the instance ID for the entry we are interested in, which in this case is IpInProblemsPercent. As usual, Systems Monitor uses this entry name as the MIB instance ID, converted into ASCII, so the instance ID may also be expressed as:

73.112.73.110.80.114.111.98.108.101.109.115.80.101.114.99.101.110.116

We have used this latter form to define the MIB variable to monitor, as shown in Figure 224. See page 167 for further explanation of how to specify the Local/Remote MIB Variable field.

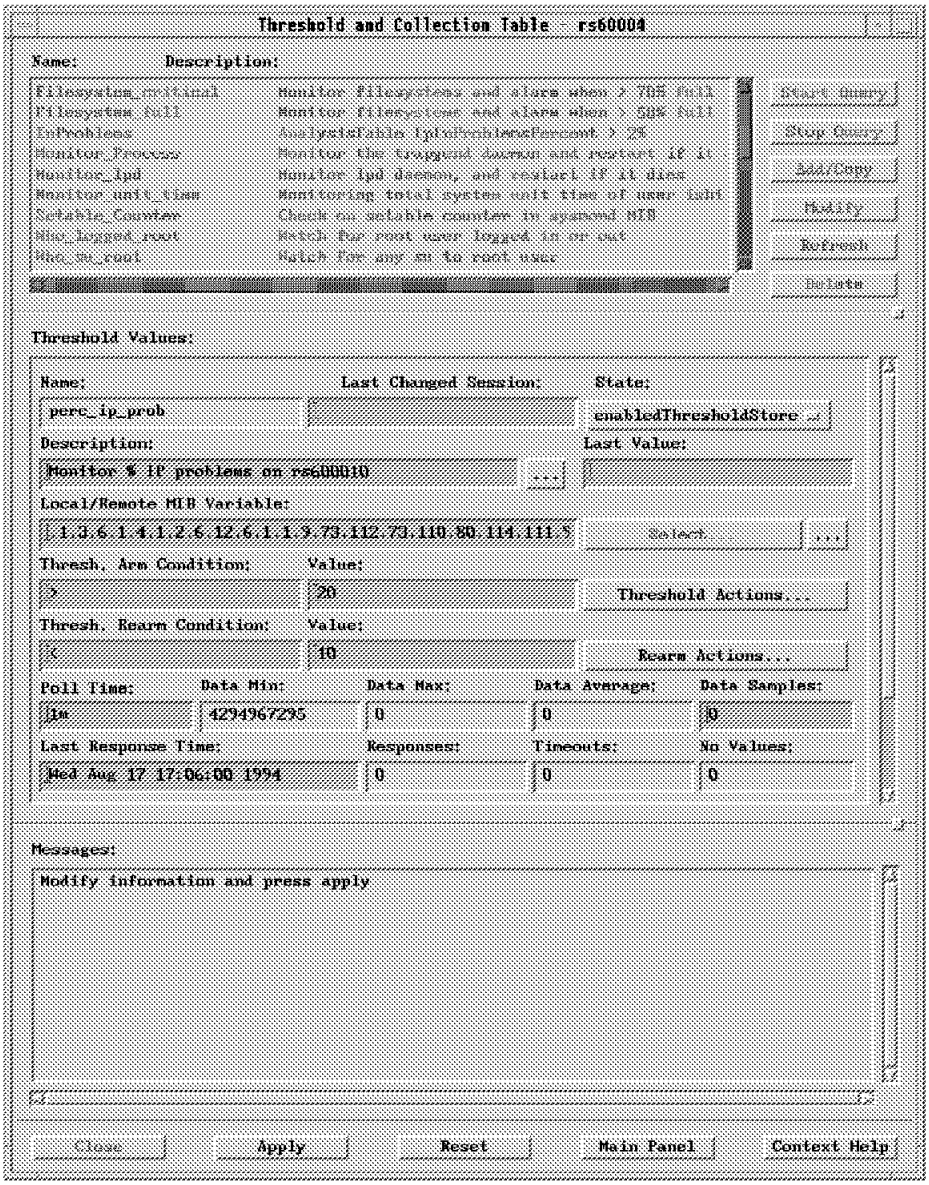


Figure 224. Threshold Table Entry for Monitoring Percentage of IP Errors

In this case, we do not enter the hostname before the MIB variable, as was done previously for thresholding Command Table variables. This is because the Analysis Table is a function of the Mid-Level Manager and therefore its results are stored on the Mid-Level Manager node, that is, the same node where the Threshold Table is active.

Considerations if Using an SLM in Place of the MLM: This example would operate in the same way if an SLM was used instead of an MLM, except that the MIB variable specifications in the Analysis Table would not be prefixed with the node name (rs600010). This is because the SLM can only monitor the node it is on, so it would have to be running on rs600010 to be able to calculate the combined value.

7.3.8 NFS Performance Monitoring

In a client/server environment, NFS is often used to provide distributed file access. NFS is very sensitive to network and server degradation, so it is useful to be able to monitor the health of NFS to be able to preempt any problems.

The most useful command for monitoring NFS status is `nfsstat`. This command reports the number of outstanding NFS (RPC) requests and the status of these requests. The following is a sample of `nfsstat -c` command output:

```
Client rpc:
calls      badcalls   retrans    badxid     timeout    wait       newcred
20447      131        573        5          704        0          0

Client nfs:
calls      badcalls   nclget     nclsleep
20314      0          20314      0
null       getattr    setattr    root       lookup     readlink   read
0  0%      2  0%      0  0%      0  0%      0  0%      0  0%      0  0%
wrcache    write      create     remove     rename     link       symlink
0  0%      0  0%      0  0%      0  0%      0  0%      0  0%      0  0%
mkdir      rmdir      readdir    fsstat
0  0%      0  0%      0  0%      20312 99%
```

Figure 225. Sample Output of nfsstat -c Command

In the above output we should pay particular attention to the retrans field. It counts the number of times that the NFS client retransmits a request to the NFS server. If this value exceeds 5% of the calls field in this output we would expect to see some noticeable performance degradation.

In this sample, we define an SIA Command Table entry with the following command:

```
nfsstat -c | awk 'NR == 4 {print ($3 / $1) * 100}'
```

This takes the fourth line of the output and divides retrans by total calls, multiplying by 100 to get a percentage. The configured Command Table to execute this is shown in Figure 226 on page 251.

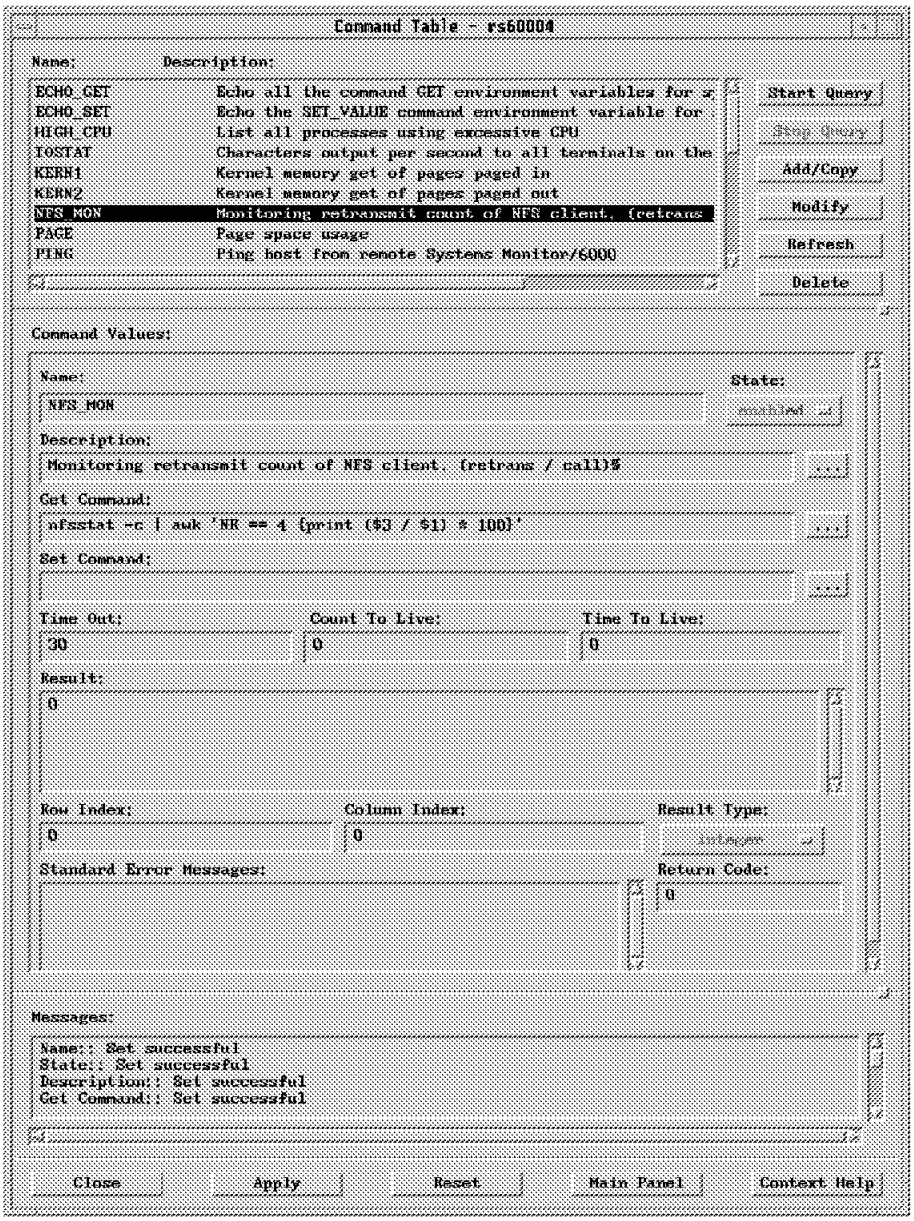


Figure 226. NFS Monitoring Command Table Entry

We would like to know of NFS performance problems before the symptoms become noticeable. Therefore we created a MLM Threshold Table entry to monitor the command as shown in Figure 227 on page 252.

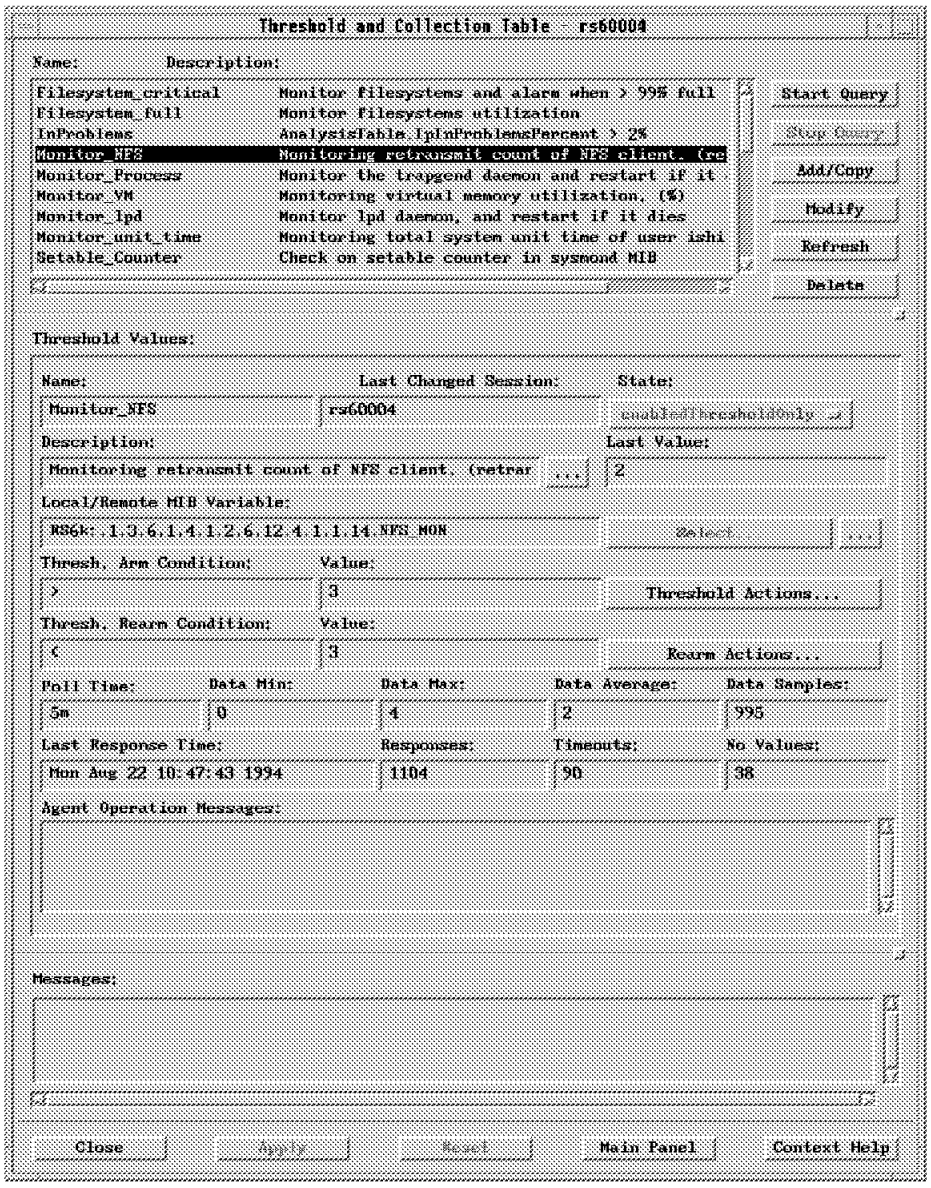


Figure 227. NFS Monitoring Threshold Table Entry

The MIB variable being monitored is the integer output field of the Command Table. See page 167 for further explanation of how to specify the Local/Remote MIB Variable field. The threshold will be triggered if the percentage error value exceeds 3%. Figure 228 on page 253 shows the Threshold Actions panel for this example.

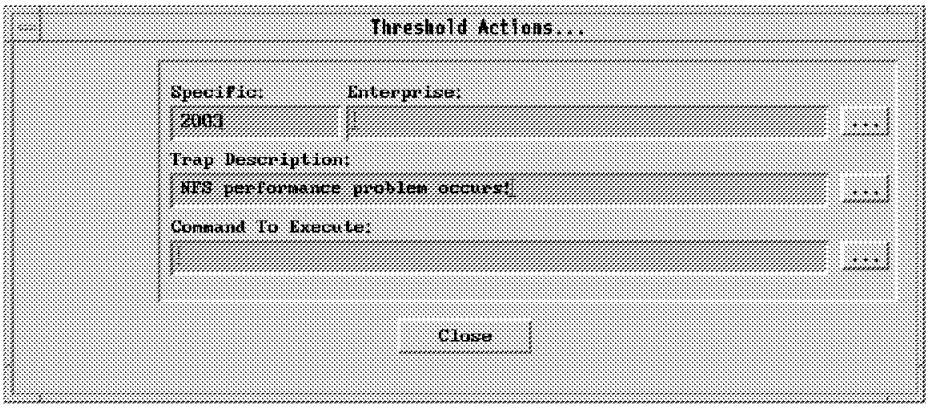


Figure 228. NFS Monitoring Threshold Action

Considerations if Using an SLM in Place of the MLM: This example would operate in the same way if an SLM was used instead of an MLM, except that the MIB variable to be monitored would not be prefixed with the alias name (RS6k), because the SLM would have to be running on each node being monitored (it can only monitor the node it is on).

7.4 Performance Data Collection

Many of the threshold monitoring examples described previously in this chapter could equally well be applied to data collection. In this section we show how data collected by the Mid-Level Manager may be imported into NetView for AIX and then converted into relational database format, using the extended RDBMS support of NetView for AIX Version 3.

The following examples are in this section:

- 7.4.1, “Collecting MIB Data with Mid-Level Manager” shows how to configure the Mid-Level Manager Threshold Table to collect performance data.
- 7.4.2, “Converting MLM Collected Data” on page 254 shows how to convert the data collected by Mid-Level Manager into NetView for AIX format and then to load it into the relational database.
- 7.4.3, “Querying Collected MIB Data Using SQL” on page 255 shows some examples of using NetView for AIX utilities to report the collected data.
- 7.4.4, “Automatically Converting Collected MIB Data Using cron” on page 257 shows an automated approach to data collection and conversion.

7.4.1 Collecting MIB Data with Mid-Level Manager

The MIB object we decided to collect for this example is the system CPU utilization, provided as part of the Systems Information Agent MIB. The MIB object is `smSiaSystemUtilizationCPUUser` (`.1.3.6.1.4.1.2.6.12.2.9.1.3.1.4`). The configuration of the Threshold Table is shown in Figure 229 on page 254.

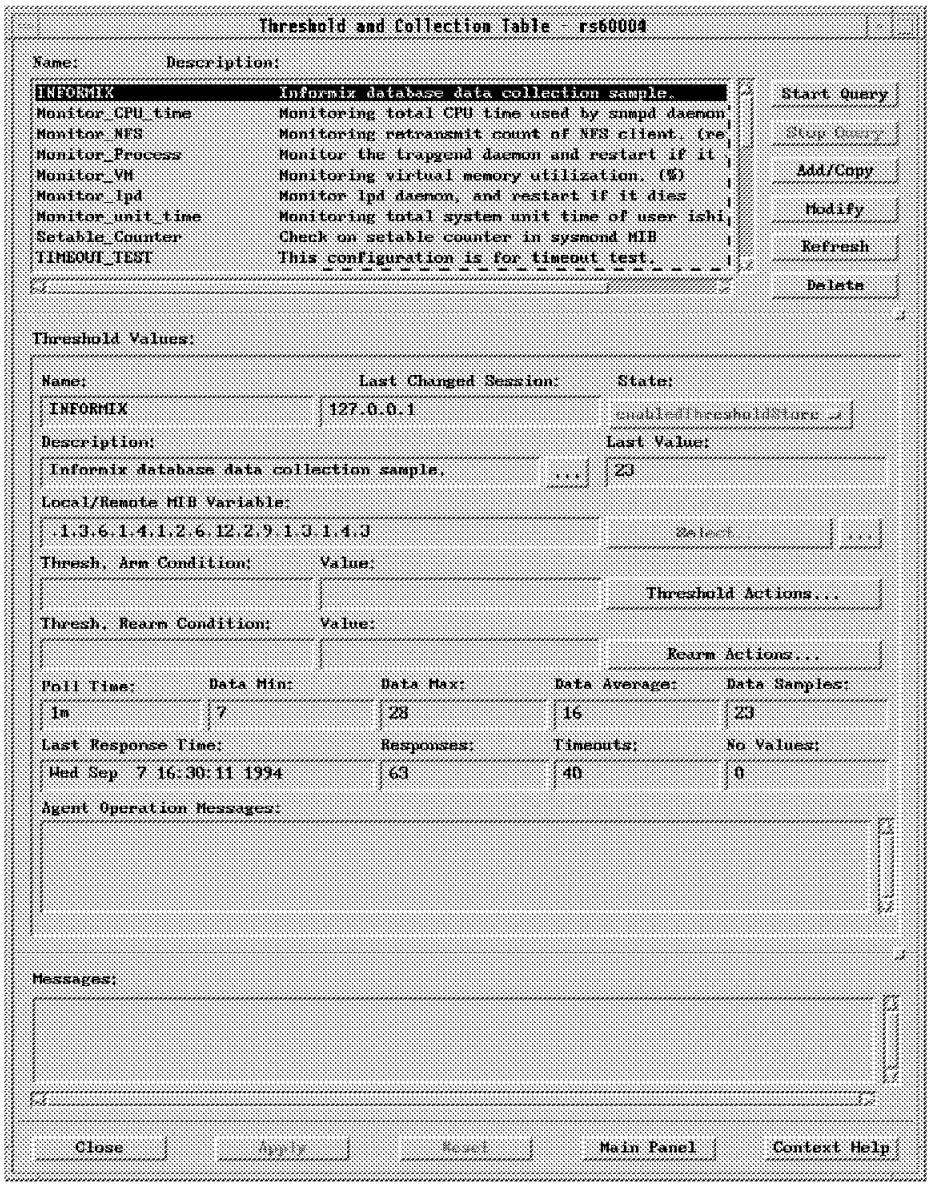


Figure 229. Threshold Table Configuration for Data Collection

You will notice that the collection option is set to enabledThresholdStore which means that the data collected will *both* be compared with a threshold value and stored for later analysis. The polling interval is set at 1 minute, which is what you may expect for threshold polling, but much more frequent that is normal for data collection. If you want to threshold and store data in this way, you may well want to set up two Threshold Table entries, one with a short cycle for exception monitoring, the other with a longer cycle for data collection.

7.4.2 Converting MLM Collected Data

In this step we will take the data collected by the Mid-Level Manager and convert it into the format used by NetView for AIX. Then we will take the converted data and load it into an Informix database.

The MLM collects data in file /usr/adm/smv2/collect/midmand.col. This is a text format file, so our first step is to convert it into the format used by snmpCollect on NetView for AIX. The Mid-Level Manager and NetView for AIX will generally

be on separate machines. The first thing you have to do, therefore, is to send the collected data to the NetView for AIX machine for conversion. snmpCollect uses a binary format for data collection. Systems Monitor for AIX provides a conversion tool, the smconvertv2 command, to move the data from one format to the other. The smconvertv2 command has the following syntax:

```
./smconvertv2 [-d dir] [file]*
```

The first argument is the directory in which we want to store the converted file, the second is the source file. The reason the target has to be a *directory* is that the snmpCollect format keeps the data collected for *each* different MIB instance in two separate files, so one MLM log file can result in many converted files. The two files for each MIB instance are the data file and the ″bang″ file. The data file has a file name describing the data, with an extension of the MIB instance (for example, snmpInPkts.0). The ″bang″ file has the same file name suffixed by a ! (for example, snmpInPkts.0!). It describes the MIB object information for the data file. We recommend that the data is converted into the standard snmpCollect database directory, /usr/OV/databases/snmpCollect, since then it may be manipulated in the same way as ″real″ snmpCollect data.

Our next step is to load the collected data into the relational database. NetView for AIX allows you to use one of five different RDBMS′s for this purpose (Informix, Sybase, Oracle, Ingres or DB2/6000). In our case we used Informix. We will not describe the steps needed to set up the database server here. Please refer to *NetView for AIX Database Guide, Version 3*, SC31-7190 and *Examples Using NetView for AIX*, GG24-4327 for instructions and examples of how to install and use the RDBMS feature.

NetView for AIX provides a command to import snmpCollect collected MIB data into the relational database. The command is nvColToSQL. You have to tell nvColToSQL which file to import, using the -f option:

```
nvColToSQL -f filename_root
```

If there is no message after command execution, the nvColToSQL command has completed successfully and we can start doing SQL queries against collected data.

Notes

The nvColToSQL command by default appends new data to the MIB data already in the database. Therefore you should be careful not to convert data twice. If you wish to clear the database before loading, you can use the nvColToSQL -C command.

7.4.3 Querying Collected MIB Data Using SQL

Once the collected data has been loaded into the relational database, we can write SQL queries to extract, combine, and format it. In this project we did not have time to generate any examples of such SQL reports. However, there are some general purpose queries provided as part of NetView for AIX and we show examples of them below.

For these examples we had two types of data in the database:

1. Paging space percentage used (MIB object ID .1.3.6.1.4.1.2.6.12.2.4.4.2.1.5). This is SIA data collected by the MLM.

2. IP outbound octets (MIB object ID .1.3.6.1.2.1.2.2.1.16). This is MIB-II data collected by snmpCollect.

We will now look briefly at the output from the NetView for AIX basic reports.

7.4.3.1 nvHostSumCol Command

This command prints a summary of the hosts for which data is in database and the number of records of each variable.

```
# nvHostSumCol

Variable Summary:

              Host                           VarID                                Instance      Count
================================== ======================== ================ ======
rs60002.itso.ral.ibm.com           .1.3.6.1.4.1.2.6.12.2.4.4.2.1.5.104.100.54               1    185
rs600010.itso.ral.ibm.com          .1.3.6.1.2.1.2.2.1.16                                    1     49

Expression Summary:

              Host                           ExpName             Instance      Count
================================== ======================== ================ ======
No records found for MIB Expressions


nvHostSumCol completed successfully
#
```

Figure 230. nvHostSumCol Command Output

7.4.3.2 nvDataSumCol Command

The nvDataSumCol command prints a summary of the data types (MIB variables and expressions) stored in the database and the hosts to which they apply.

```
# nvDataSumCol -V

Variable Summary:

        VarID:            .1.3.6.1.4.1.2.6.12.2.4.4.2.1.5.104.100.54
        VarName:
 .iso.org.dod.internet.private.enterprises.ibm.ibmProd.systemsMonitor600.
smSiaSystemInformation.smSiaSystemPagingI
nformation.smSiaSystemPagingSpace.smSiaSystemPagingSpaceTable.
smSiaSystemPagingSpaceEntry.smSiaSystemPagingSpacePercentUsed.104.100.
54
        VarType:          UNITS
        VarUnits:         GAUGE
        Instance:         1
        Host:             rs60002.itso.ral.ibm.com
        Count:            185


        VarID:            .1.3.6.1.2.1.2.2.1.16
        VarName:          .iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifOutOctets
        VarType:          units/sec
        VarUnits:         COUNTER
        Instance:         1
        Host:             rs600010.itso.ral.ibm.com
        Count:            49

nvDataSumCol completed successfully
#
```

Figure 231. nvDataSumCol Command Output

7.4.3.3 nvQColData Command

The nvQColData command prints rows from the colData table. We can specify arguments to restrict the data extracted by data, host name or data type. In this case we restrict it to the paging space information collected by the MLM.

```
# nvQColData -V .1.3.6.1.4.1.2.6.12.2.4.4.2.1.5.104.100.54

        VarID:            .1.3.6.1.4.1.2.6.12.2.4.4.2.1.5.104.100.54
        Instance:         1
        CollectTime:      Wed Aug 31 11:56:33 1994
        StartTicks:       0
        StopTicks:        778348593
        HostName:         rs60002.itso.ral.ibm.com
        IpAddr:           9.24.104.28
        StringValue:      <null>
        FloatValue:       77


        VarID:            .1.3.6.1.4.1.2.6.12.2.4.4.2.1.5.104.100.54
        Instance:         1
        CollectTime:      Wed Aug 31 11:57:25 1994
        StartTicks:       0
        StopTicks:        778348645
        HostName:         rs60002.itso.ral.ibm.com
        IpAddr:           9.24.104.28
        StringValue:      <null>
        FloatValue:       77

        ......................................................
                (181 similar records removed from here)
        ......................................................


        VarID:            .1.3.6.1.4.1.2.6.12.2.4.4.2.1.5.104.100.54
        Instance:         1
        CollectTime:      Wed Aug 31 15:11:36 1994
        StartTicks:       0
        StopTicks:        778360296
        HostName:         rs60002.itso.ral.ibm.com
        IpAddr:           9.24.104.28
        StringValue:      <null>
        FloatValue:       78


        VarID:            .1.3.6.1.4.1.2.6.12.2.4.4.2.1.5.104.100.54
        Instance:         1
        CollectTime:      Wed Aug 31 15:12:36 1994
        StartTicks:       0
        StopTicks:        778360356
        HostName:         rs60002.itso.ral.ibm.com
        IpAddr:           9.24.104.28
        StringValue:      <null>
        FloatValue:       78

nvQColData completed successfully
#
```

Figure 232. nvQColData Command Output

7.4.4 Automatically Converting Collected MIB Data Using cron

In this example we try to build an automated procedure to store the data collected by the Mid-Level Manager into the relational database. The system environment for this example is shown in Figure 233 on page 258.

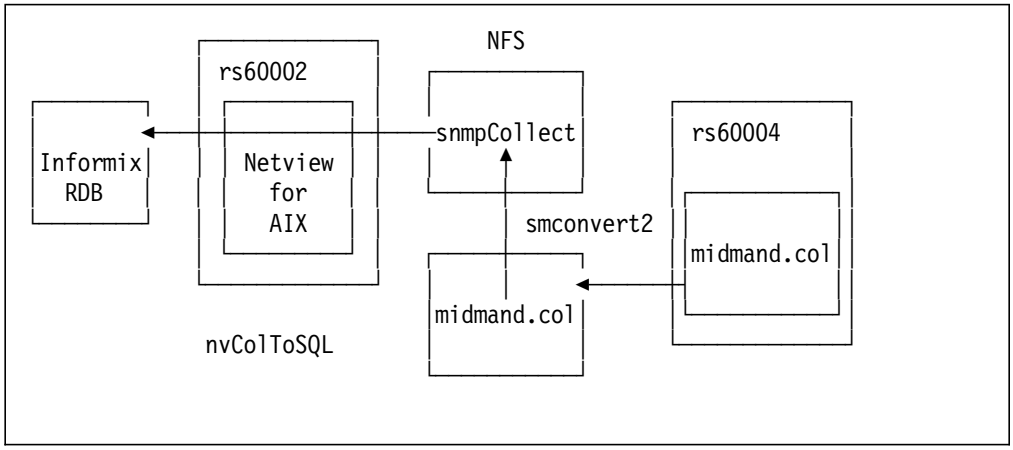


Figure 233. System Environment for Automated Data Collection

In Figure 233 the Mid-Level Manager is on node rs60004. It collects data into the midmand.col file. Our automated procedure will, every hour, take the data collected here and convert it into snmpCollect format, passing it across to the NetView for AIX node, rs60002, in the process. This is done using NFS to mount the snmpCollect database directory, /usr/OV/databases/snmpCollect on rs60004 in read/write mode.

We used the following shell script to perform the conversion:

```
#!/bin/ksh

cp /usr/adm/smv2/collect/midmand.col /usr/adm/smv2/collect/midmand.tmp

cat /dev/null > /usr/adm/smv2/collect/midmand.col

/usr/lpp/smmlm/original/smconvertv2 /usr/adm/smv2/collect/midmand.tmp

rsh rs60002 /u/ishii/SHELL/INFO

rm /usr/OV/databases/snmpCollect/3.*

rm /usr/adm/smv2/collect/midmand.tmp
```

Figure 234. Shell Script for Automatic Data Conversion

This script takes a copy of the current collected data and clears the file so that no duplicate data is sent. Then it invokes smconvertv2, which creates files called 3. and 3.! in the snmpCollect database. Next, the shell script remotely executes another shell script on the NetView for AIX (rs60002) node, using rsh. This script (/u/ishii/SHELL/INFO) is for converting MIB data from the flat file to the relational database format using the nvColToSQL command. It is shown in Figure 235 on page 259.

```
#!/bin/ksh

export INFORMIXDIR=/usr/informix
export SQLEXEC=$INFORMIXDIR/lib/sqlturbo
export PATH=$PATH:$INFORMIXDIR/bin

/usr/OV/bin/nvColToSQL -f 3.
```

Figure 235. Shell Script for Converting MIB Data

We used cron to execute these shell scripts every hour. The crontab entry to do this is as follows:

```
0 * * * * /u/ishii/SHELL/db_auto 1>/dev/null 2>/dev/null
```

In the above crontab table, /u/ishii/SHELL/db_auto is the shell script in Figure 234 on page 258.

Chapter 8. Introduction to Agent Policy Manager

The examples described in Chapter 7, “Systems Monitor Examples” on page 161 showed how to exploit some of the capabilities of the Systems Monitor for AIX agents. However, they were each focused on a single node. In reality, you are more likely to want to deal with *groups* of nodes. The new collection facility of NetView for AIX Version 4 gives us a powerful way to handle dynamic groups of nodes.

The Agent Policy Manager is another new NetView for AIX Version 4 feature, which uses collections to simplify the administration of multiple System Monitor Agents.

Think of APM as having two main pieces:

- The APM daemon
- The APM interface

8.1.1 APM Daemon

The APM daemon, /usr/OV/bin/C5d, is the central piece of software that connects AIX Systems Monitor/6000, NetView for AIX, and the Collection Facility. This daemon performs the following several functions:

- It receives definitions from the APM Configuration Interface and makes appropriate changes to Systems Monitor configuration.
- It updates the APM interface to report changes in status.
- It handles incoming Systems Monitor threshold and file monitoring traps and updates NetView for AIX submaps and nvevents accordingly.
- It catches update information from the Collection Facility and makes changes to threshold and file monitor definitions on nodes that were added to or deleted from collections.
- It retries failed definitions, both when the daemon is started and periodically thereafter.

When the daemon is started, it searches the object database, looking for MLMs. It creates domains for all MLMs and assigns one to be the default domain. If this is not the first time the Agent Policy Manager has been started, it will look for new MLMs and create domains for them, and it will move domain responsibilities for any deleted MLMs to another MLM that is still active.

At startup, the daemon also retries any outstanding distributions, including any partially distributed or failed definitions and deletions, and any distributions or deletions that were in progress when the daemon stopped.

8.1.1.1 Configuring APM (C5d) Daemon

If you want to use the new NetView for AIX features like APM in conjunction with AIX Systems Monitor/6000, you need to activate the C5d daemon. This daemon is not registered during the installation of NetView for AIX Version 4. NetView for AIX uses this daemon to coordinate the threshold monitor and file monitor definitions for AIX Systems Monitor/6000 MLMs and SIAs. C5d depends on another daemon, nvcold, which manages collections. Make sure nvcold is registered and running before you activate C5d.

```
rs600010:/u/peterg > ovstatus nvcold
 object manage name: nvcold
 behaviour:          OVs_WELL_BEHAVED
 state:              RUNNING
 PID:                43449
 last message:       Initialization complete
 exit status:        -
```

Figure 236. Checking nvcold Status

If you migrated from NetView for AIX Version 3 to the current version of NetView for AIX, you probably decided to use your existing topology database. C5d will examine this database when the daemon starts for the first time. This examination may take from minutes to a couple of hours. Make sure, C5d has a chance to finish its database examination before you proceed.

You must register the C5d daemon using the NetView for AIX registration utility.

```
rs600010:/u/peterg > ovaddobj /usr/OV/lrf/C5d.lrf
ovaddobj - Static registration Utility
Successful Completion
rs600010:/u/peterg > ovstart C5d
rs600010:/u/peterg >
```

Figure 237. Registration and Start of C5d

As with other NetView for AIX related daemons, you register the daemon using the static registration utility ovaddobj as shown in Figure 237.

As an alternative you can use to review and eventually adjust the daemon configuration to your needs.

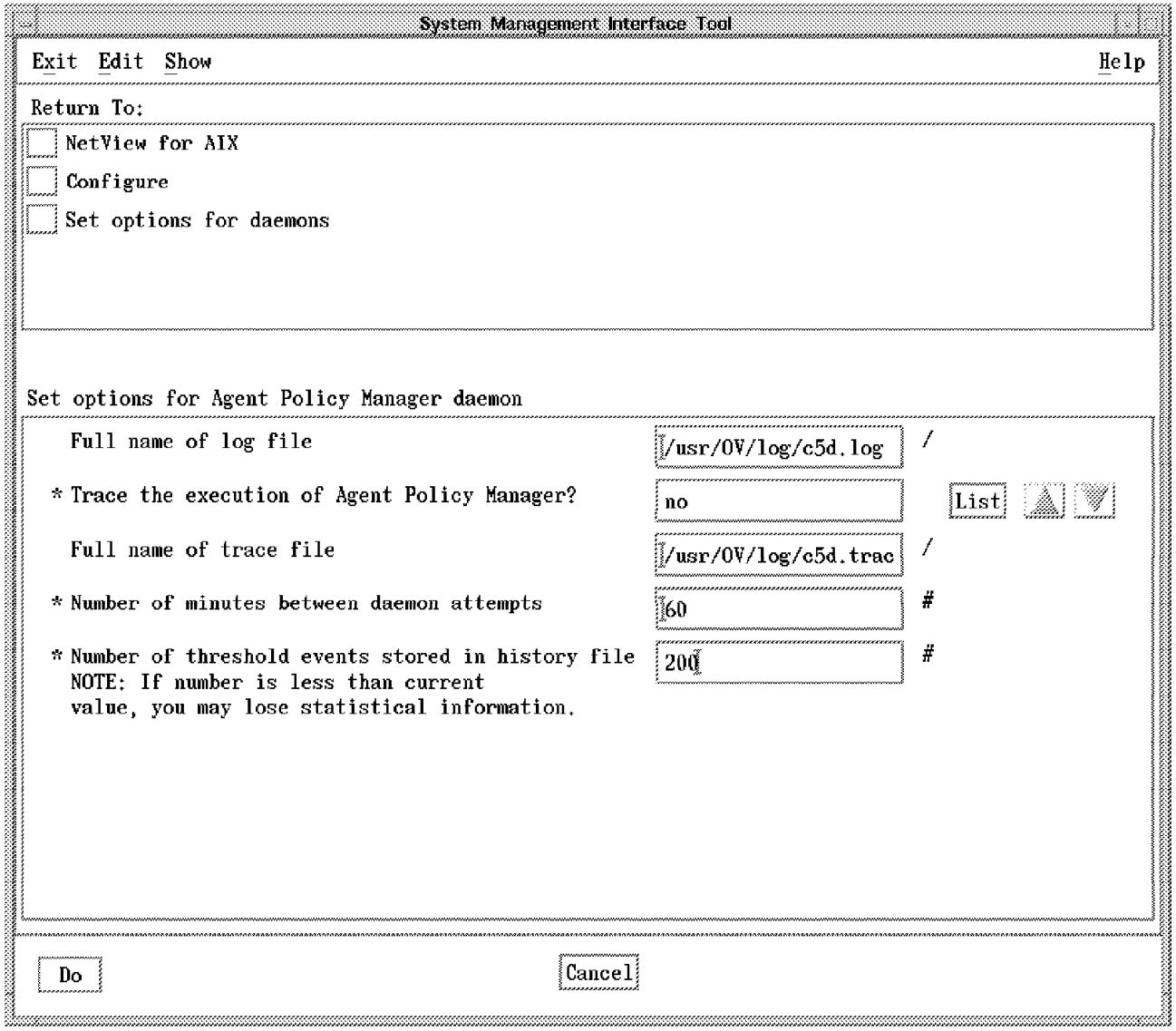


Figure 238. APM Configuration SMIT Dialog

Use the smit nv6000 fast path to access the NetView for AIX related SMIT menus. From there select the following:

Configure
-- Set Options for Daemons
---- Set Options for Agent Policy Manager Daemon

which brings you to the APM related Dialog. Using the SMIT Dialog will also register the C5d daemon for you, so you don't have to do it explicitly. The Dialog itself provides you with the following options:

Full name of log file
Specifies the full path to the daemons' log file. The default path is /usr/OV/log/c5d.log.

Trace the execution of Agent Policy Manager?
Toggles tracing of the C5d daemon. The default is No. Setting this option to Yes produces a ASCII file containing trace information.

Full name of trace file

Specifies the full path name to the C5d trace file. The default path is set to /usr/OV/log/c5d.trace.

Number of minutes between daemon attempts

This value determines how often C5d will attempt to distribute threshold and file monitoring definitions that it could not distribute previously (for example, if a node was down). The value is expressed in minutes, the default is set to 60 Minutes.

Number of threshold events stored in history file

Changes the number of threshold events stored in the history file. You can use these events as plot points by the xnmgrapher application. Also, You can graph a threshold's history through the PDA Application. By default, C5d stores the 200 most recent events in the history file.

Note

If the number of history events is less than the default value, you may lose statistical information.

Select **Do** in the Dialog to accept the changes and register the daemon. After you have registered and started the daemon, it will be started automatically when you start NetView for AIX.

The C5d daemon accepts the following two command line parameters you can issue every time C5d is active:

-L Toggles logging to the log file specified during configuration or startup.

-T Toggles tracing to the trace file specified during configuration or startup.

Once started, C5d immediately builds collections containing nodes running MLM, SIA and SLM agents. In addition, the daemon builds collections containing all the nodes managed by the discovered MLMs. You may check the NetView for AIX root map for the existence of these submaps.

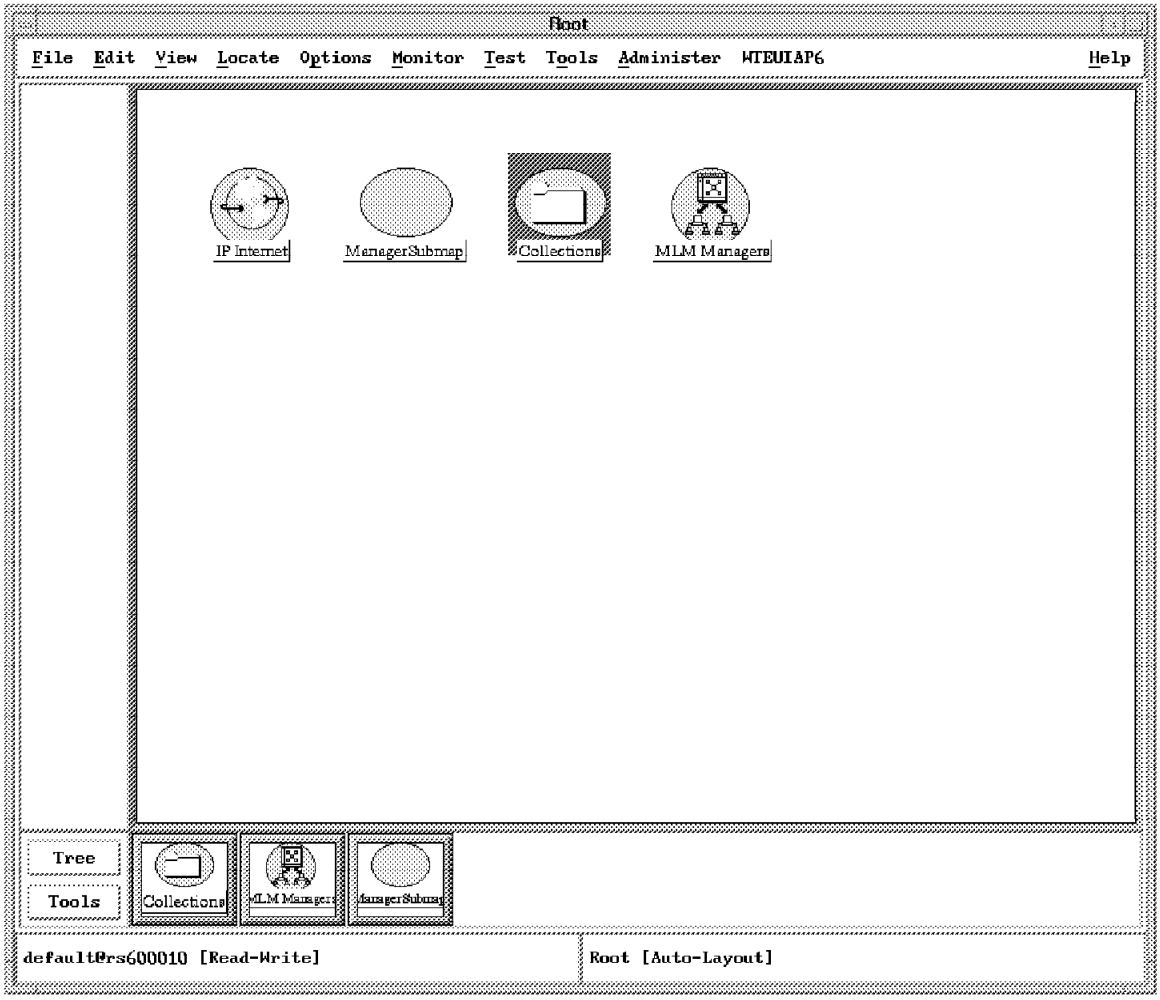


Figure 239. Root Map After Activating C5d

Please note that those submaps are not activated right now. Their color appears blue (which is not easy to recognize in the greyscale Figure 240 on page 266). NetView for AIX will activate the collections as soon as you select them for the first time.

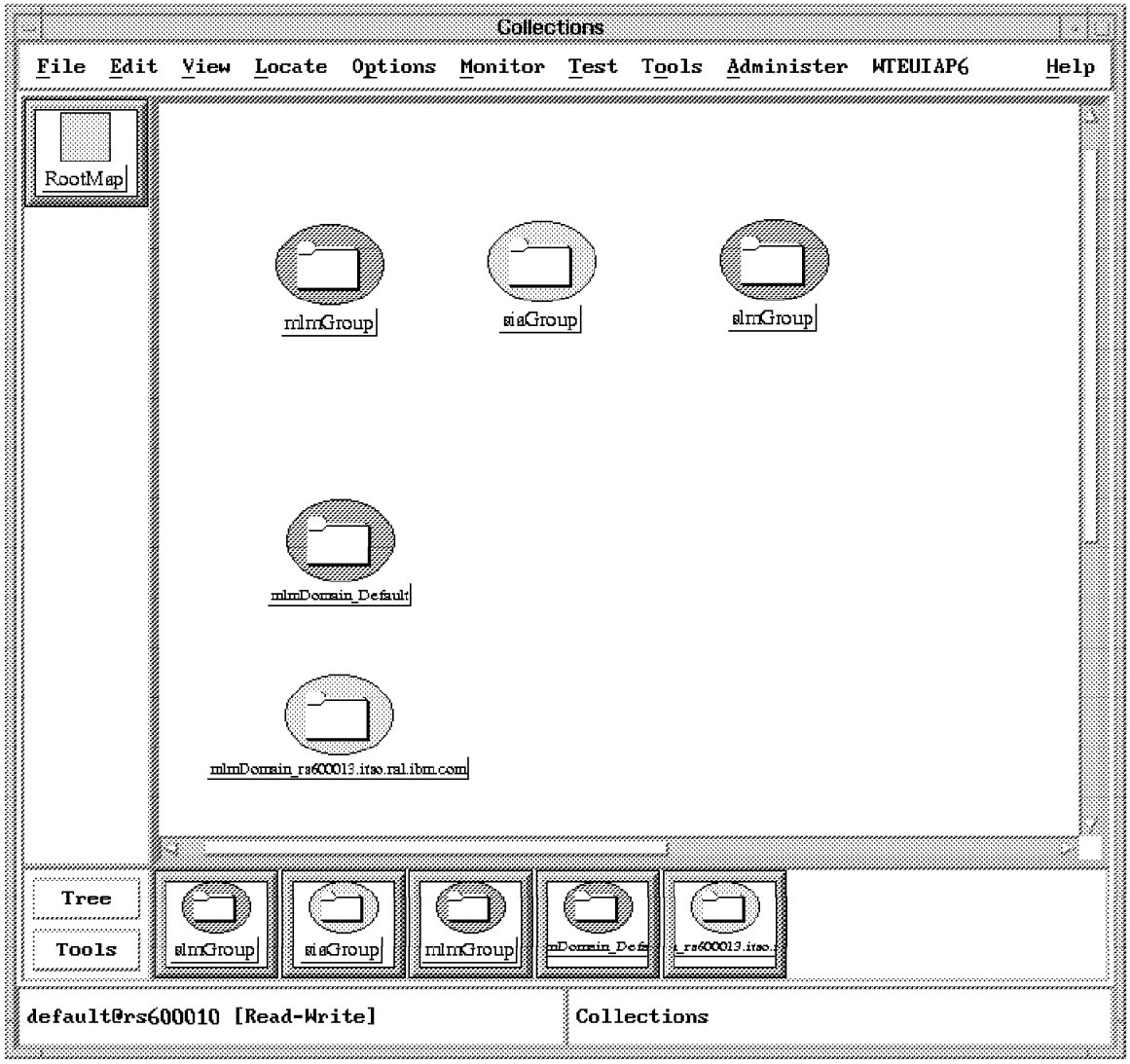


Figure 240. Collections Automatically Built by C5d

8.1.2 APM Configuration Interface

The APM interface is your tool for defining and manipulating thresholding and file monitoring definitions. Much like the Systems Monitor Configuration Application, the APM configuration interface gives you fields for defining information about a threshold or file monitoring condition. Through the APM configuration interface, you can get a list of condition that have been defined, add or delete definitions, and distribute definitions. You specify collections to which a definition will be applied through the EUI.

The EUI Dialog maintains the current status of all the definitions. For a complete status list refer to Appendix D, “APM Distribution Status Indicators” on page 311.

8.1.2.1 Starting the Configuration Interface

The APM Interface is integrated into the NetView for AIX map interface. You can start it from two places:

- Select **Tools...APM Configuration** from the NetView for AIX menu bar.
- Double-click on the **APM** icon on the Tools Window or drag the icon to a free space on the desktop.

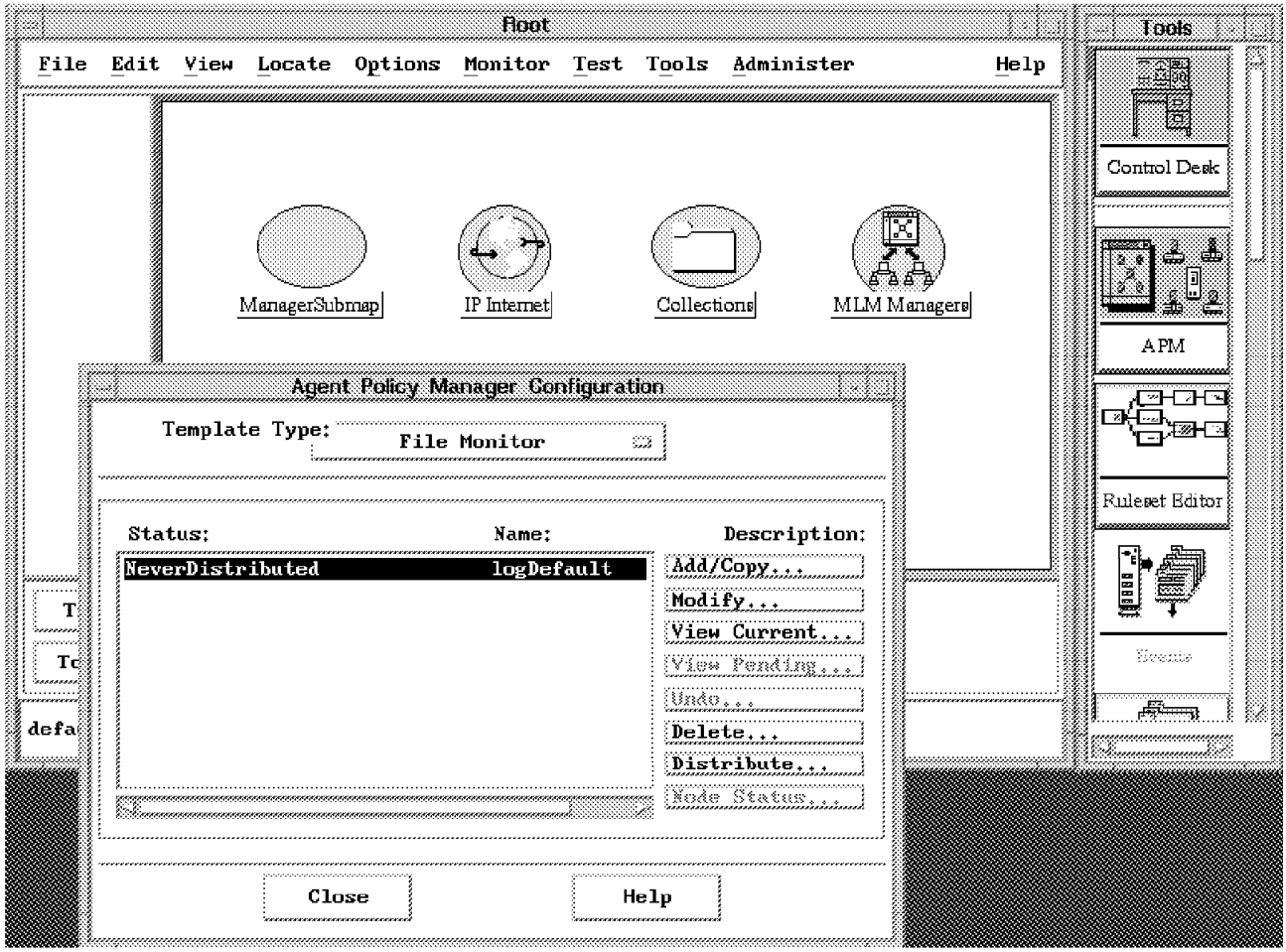


Figure 241. Initial APM Configuration Dialog

From the APM main dialog, you can look at a list of current log and threshold definitions. To switch from threshold to log definitions, use the Template type toggle button. You can also add, delete, or modify a file monitor or threshold definition, and distribute the definition to other nodes in the network.

Note

The first time you start the APM EUI, it might take some time to appear. The EUI must set status on all nodes in the associated collections. Agent Policy Manager uses and creates submaps for the collections. After this initial synchronizing has been done, you will not see a delay in the EUI startup.

8.2 APM MLM Domains

For thresholding, APM defines domains for the MLMs in the network. Basically, it determines where the MLMs are and assigns the managed objects.

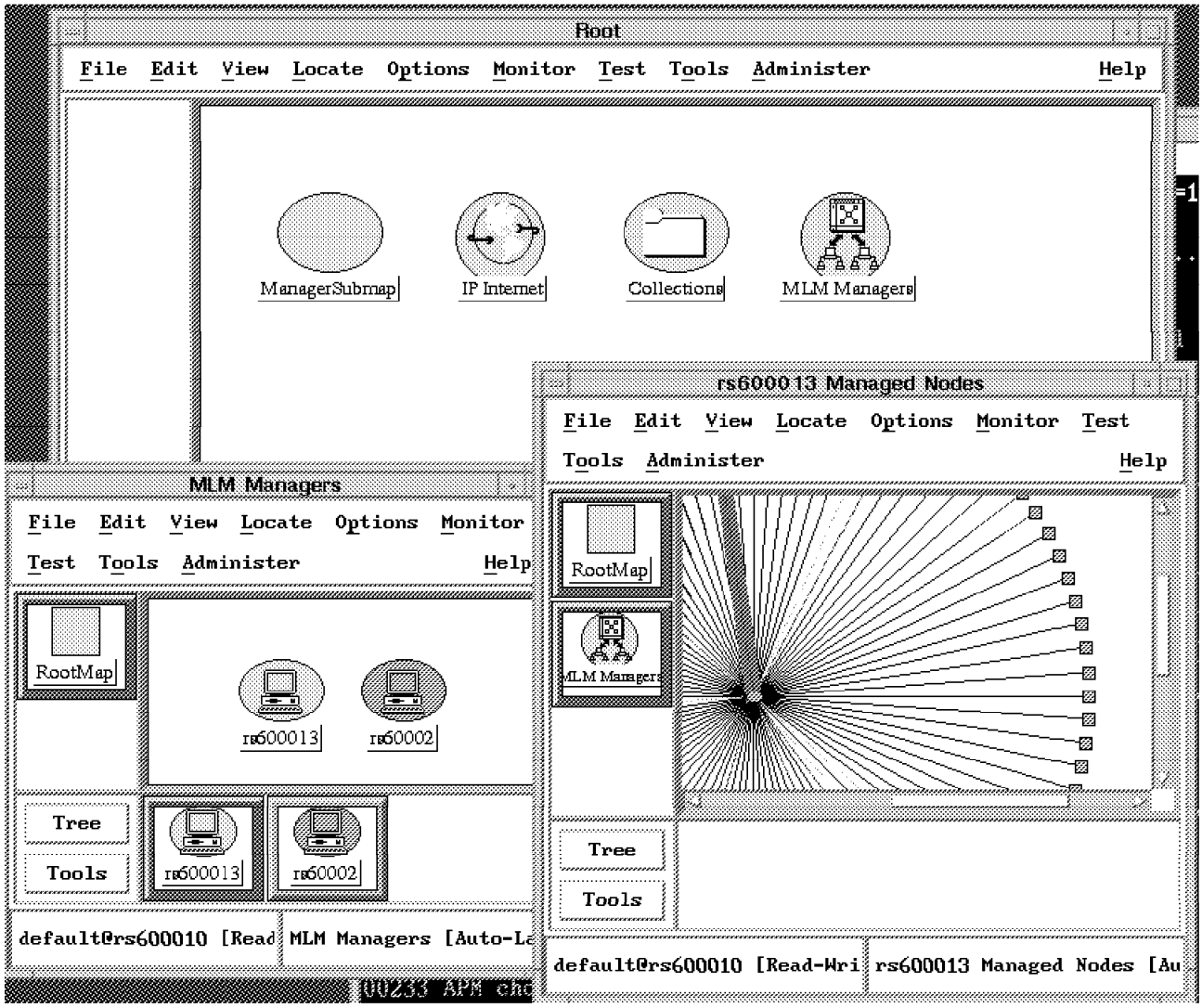


Figure 242. MLM Domains Automatically Discovered by NetView for AIX

Initially, MLM domains are defined based on subnet IDs. Nodes that are in the same subnetwork as an MLM are assigned to that MLM. Nodes that do not have an MLM in their subnet are assigned to a default domain. There is no overlap in these initial definitions; each node is managed by one MLM only.

You can see these domains from the NetView for AIX root map. The MLM domains are placed under the icon MLM Managers.

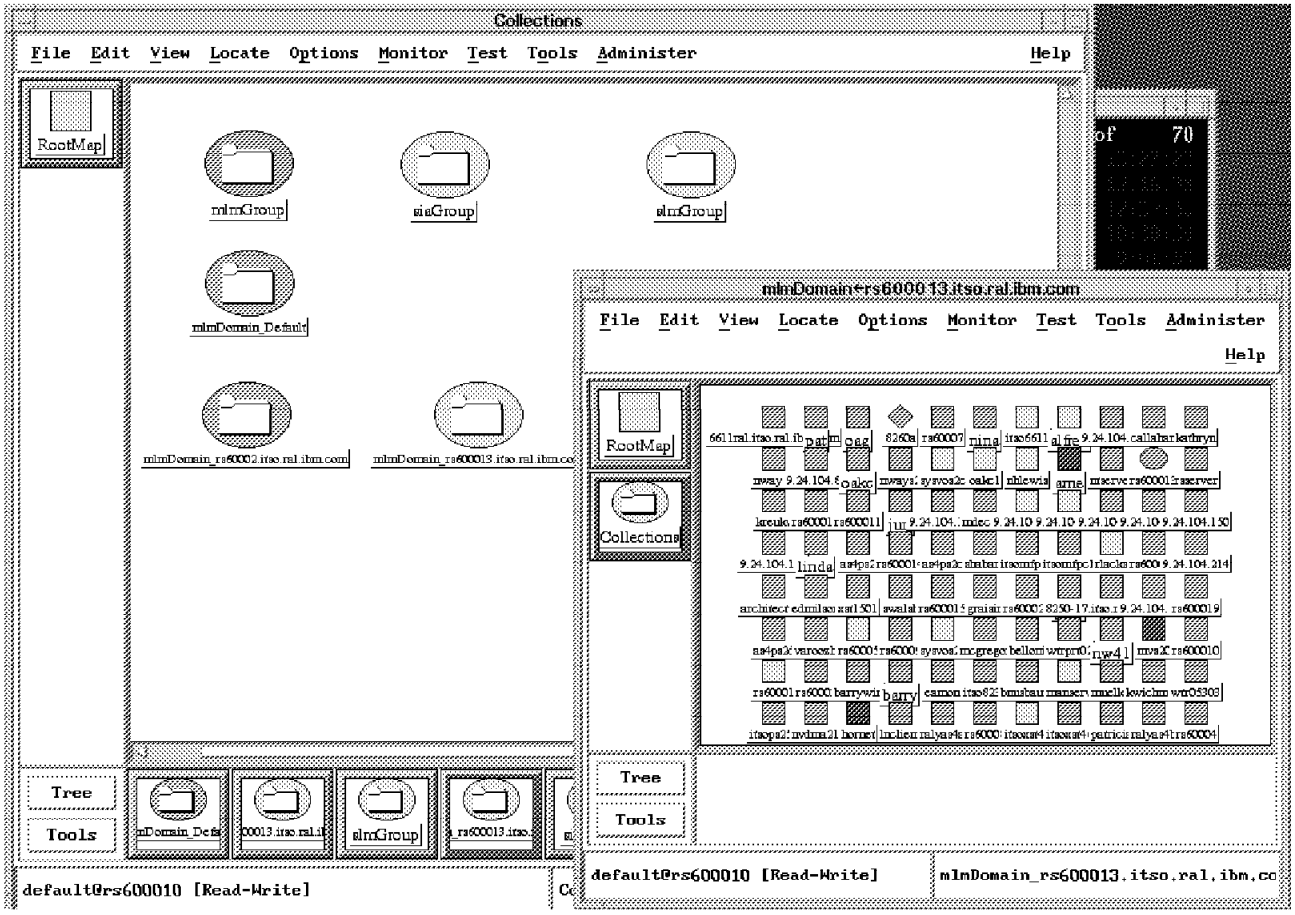


Figure 243. Collection View of MLM Domains

Also, if you start the Collection Editor, or select the collection icon from the root map, you will see the MLM domains (as well as the mlmDomain_Default domain) listed as already existing collections. APM chooses the MLM that will serve as the default domain depending on how your network is set up:

1. If there is an MLM on the NetView for AIX manager, it will be used.
2. If the MLM is not installed on the NetView for AIX manager, you must assign the default domain collection to one of the MLMs in your network using the Collection Editor.

If there is only one MLM in your network, that MLM is assigned all nodes, and a default domain might not be created.

8.3 Working with MLM Domains

Since an MLM domain is just a collection, you can use the same rule logic that you use for defining collections to set thresholds against. You do not need to reassign any nodes that were in the old domain but are not in the new domain; APM takes care of assigning new domain responsibilities for the nodes.

If you have MLMs that are off-loading discovery and status monitoring from NetView for AIX and are very busy as a result, you might want to lighten their workload by moving nodes out of their APM domains.

If you add a new MLM to your network, Agent Policy Manager will recognize it as an MLM and assign a domain collection, but the collection will be empty. If you edit this collection and add nodes to it, the Agent Policy Manager moves the nodes accordingly.

You might want to alter the distribution of nodes to MLMs to facilitate your management of the network.

8.3.1 Example Using MLM Domains

In this example, we will use an additional MLM to perform status checking for a group of specialized nodes.

You might want an MLM to manage all the routers or to manage all objects in a physical location.

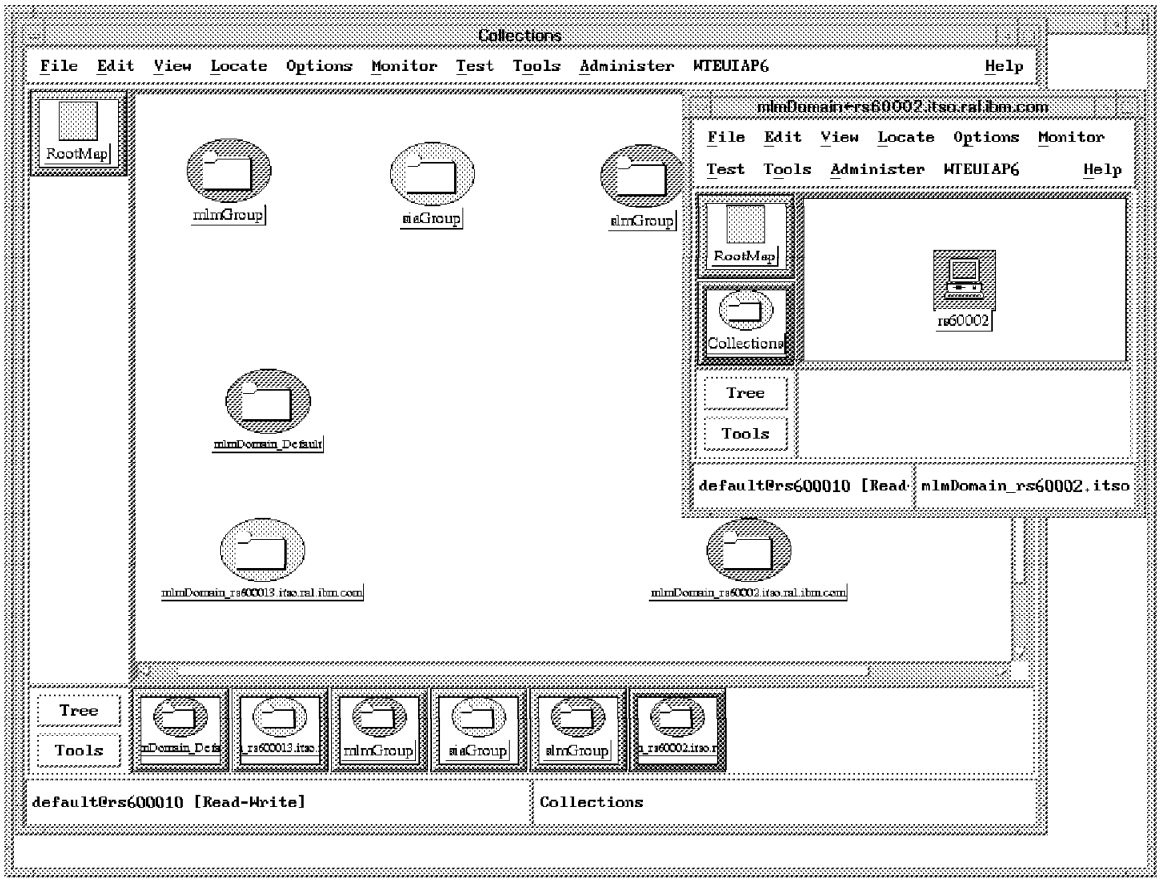


Figure 244. The Newly Installed MLM with No Duties

To test this, we set up another MLM in our subnetwork. As you may remember from Figure 91 on page 109, the nodes rs60002 and rs600013 both have an MLM active and reside in the same subnet. At the time we started the APM interface, rs600013 was first discovered in the network and therefore was assigned all the nodes. Later on, rs60002 was discovered and APM generated a collection for rs60002 containing just rs60002 itself as the only member of the collection.

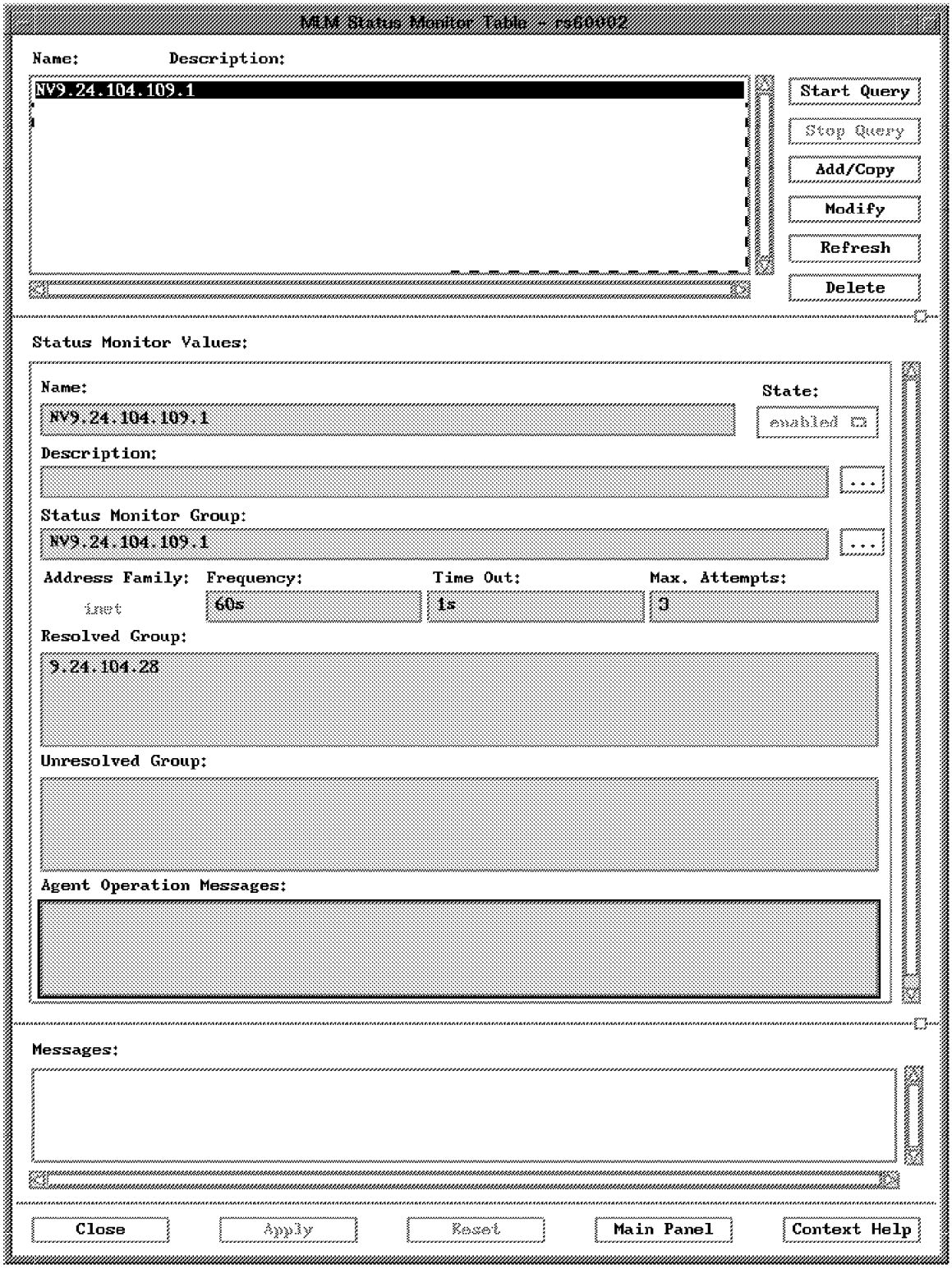


Figure 245. Status Monitor Table of the MLM

Examining the Status Monitor table of the new installed MLM shows the same information: the MLM is not enabled to control other nodes.

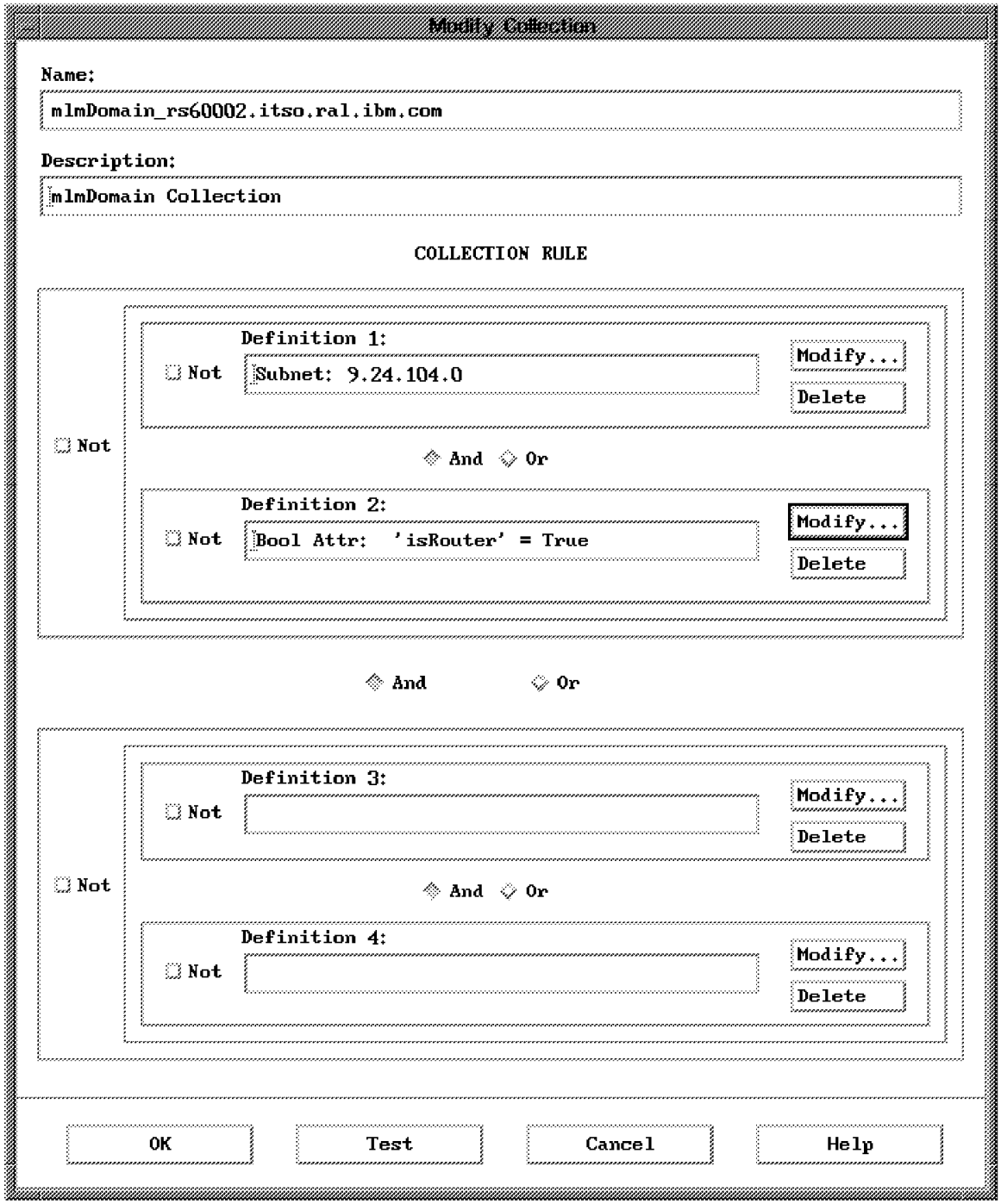


Figure 246. Collection Selecting Specific Nodes in a Subnet

We used the Collection Editor enabling this MLM to control all routers,hubs and bridges in its subnet. To accomplish our task, we used a combined rule:

1. A subnet clause to limit the duties of the MLM to its own subnet. This will be done by coding a IN_SUBNET = 9.24.104 type.
2. Three attribute clauses of type isRouter, isHub and isBridge to select all the specific nodes.

Figure 246 shows the dialog to build this Collection rule.

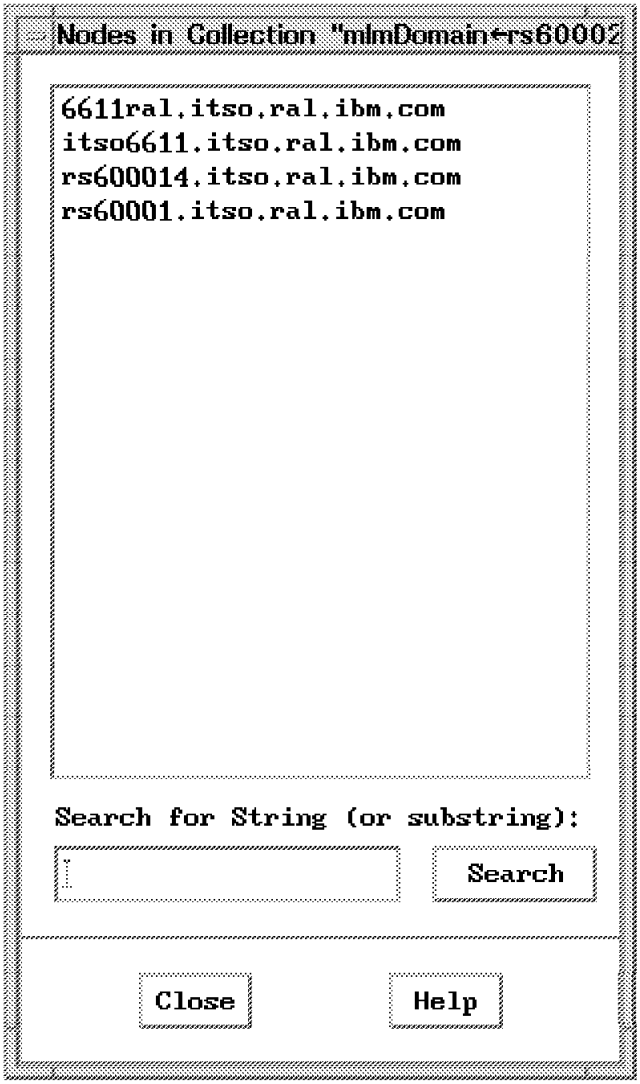


Figure 247. Resultant Collection

Before we finally build the collection, the Collection Editor offers to test the collection rule. As you can see in Figure 247, six nodes will be inserted into our collection. We used Ok to activate the collection.

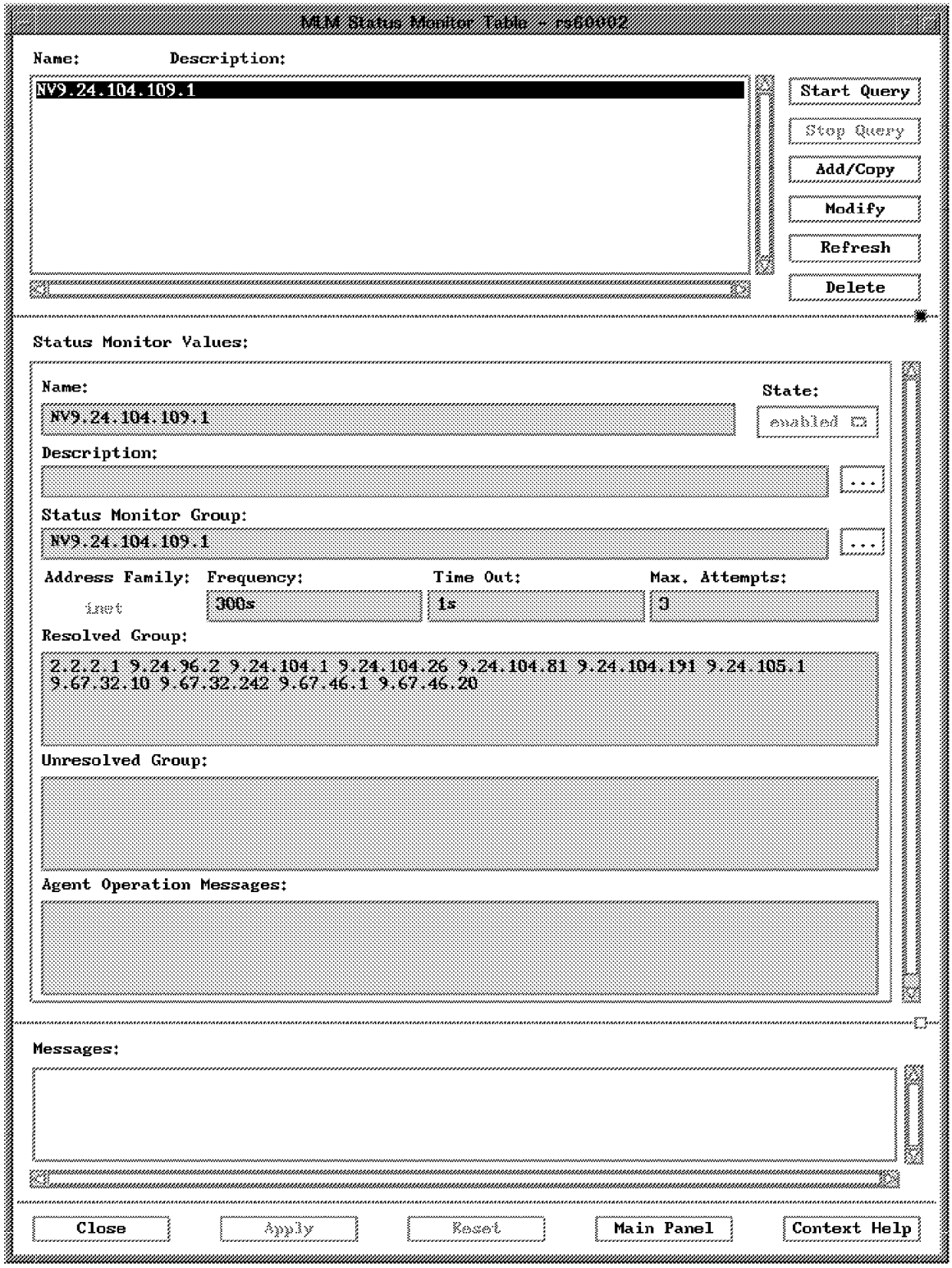


Figure 248. MLM Status Monitor Table Updated by APM

To check for correct distribution of the new collection, we examine the Status Monitor Table of the targeted Mid-Level Manager. It should reflect the recent changes. Note that not only the selected nodes *and* their interfaces into other subnetworks have been included into the Status Monitoring Table. This is how status monitoring works.

Note

The current version of APM in NetView for AIX Version 4 does not update the MLM tables immediately after changes have been made. While the maps will be updated as soon as a topology change is detected, the required update of MLM tables occurs a variable amount later, so, be patient.

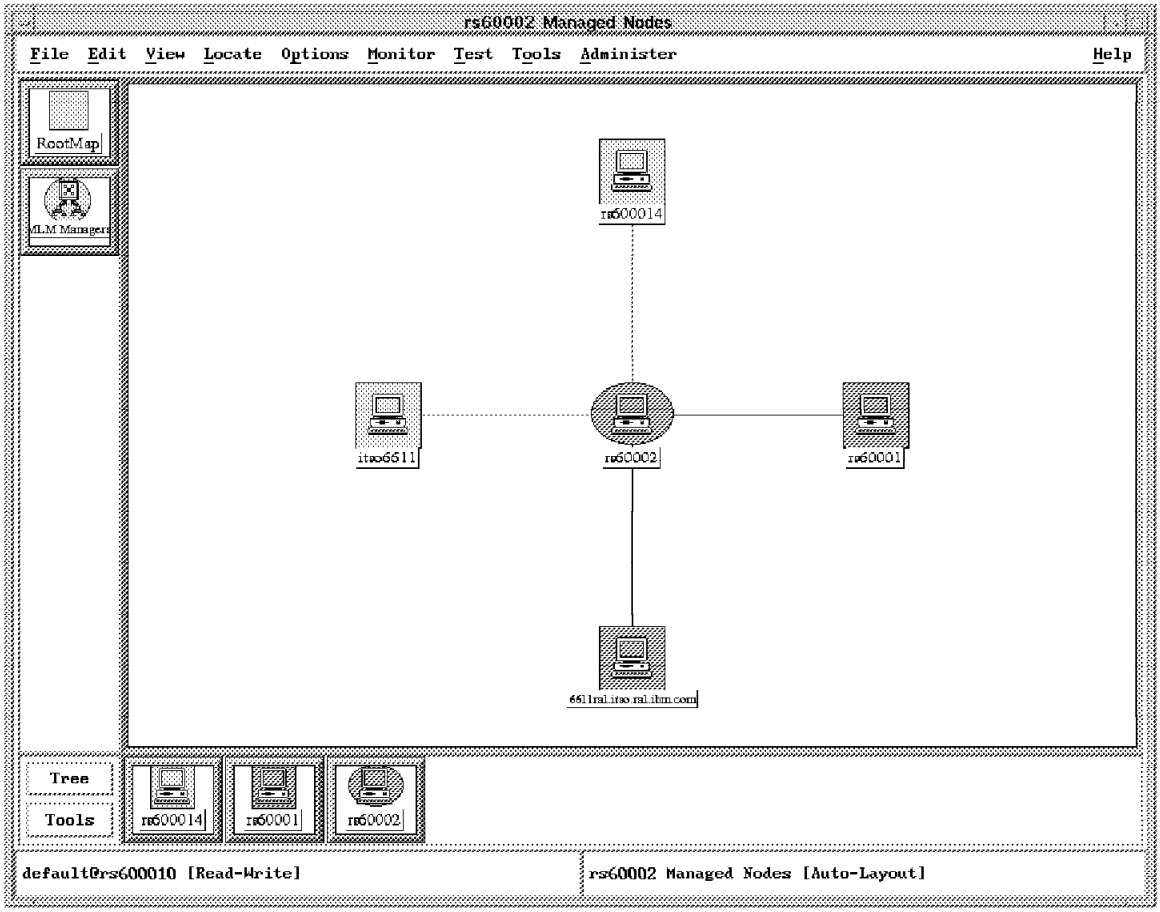


Figure 249. MLM Manager View of the Collection

For a last verification of the correct implementation of this rule, we use the MLM Manager View. Selecting the rs60002 MLM from the MLM Manager submap shows the six nodes being managed by rs60002 as in Figure 250 on page 276.

8.3.1.1 A Common Pitfall

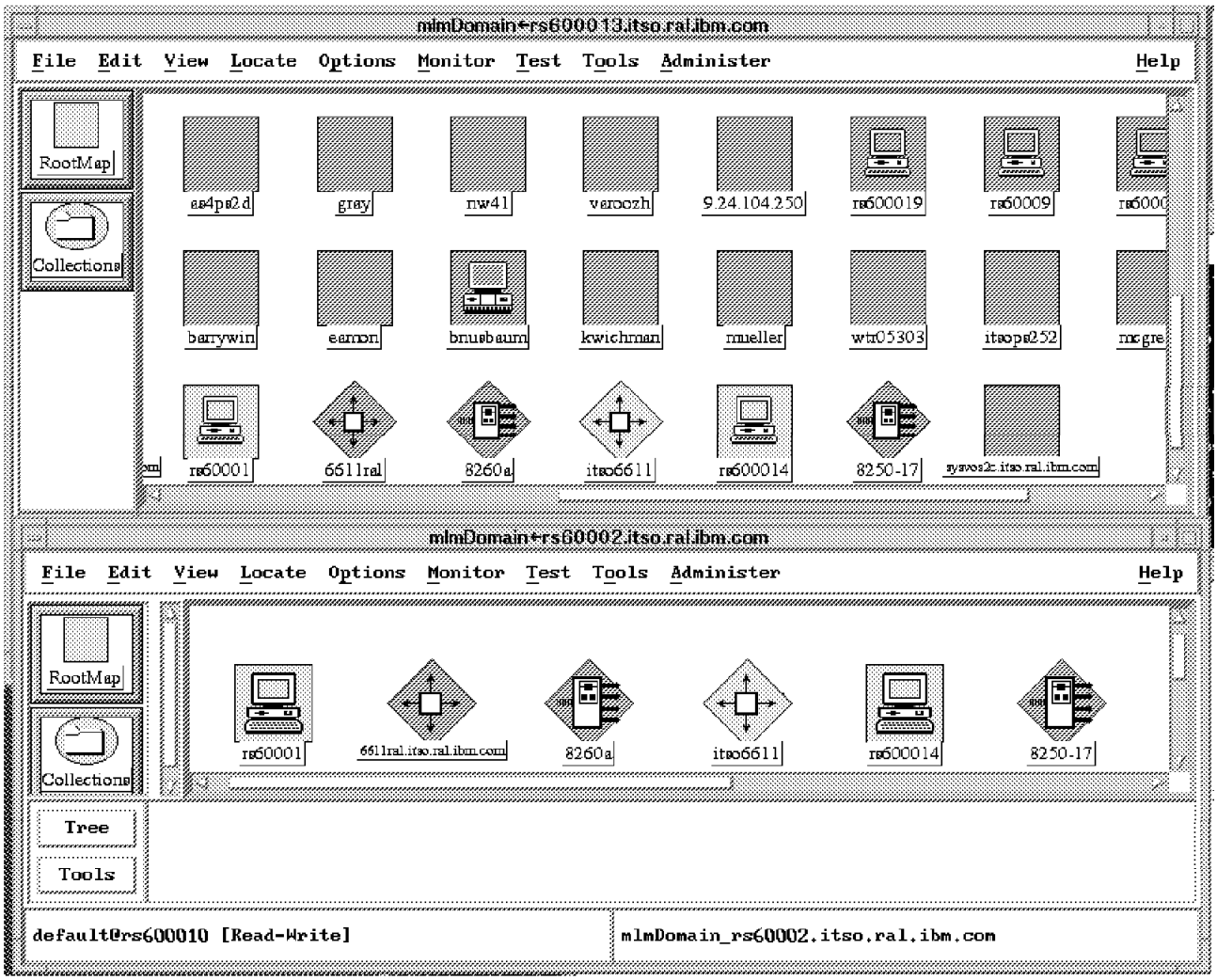


Figure 250. Duplicate Nodes

After modification of the MLM collection as described in 8.3.1, “Example Using MLM Domains” on page 270 the collection seems to be working. A closer look at both collections, as in Figure 250, shows that the selected nodes appear in both collections. Now, what can happen in such a constellation?

To get some more information, we modified the rs60002 collection to include just the OS/2 node 9.24.104.185. Then we simulated a node down condition by issuing an `ifconfig lan0 down` command. The result of that operation was examined in the /usr/OV/log/trapd.log file.

```
rs600013 n Systems Monitor detected interface 9.24.104.185 down.
9.24.104.185              N Interface 802.5 down.
9.24.104.185              N Node Down.
rs60002  n Systems Monitor detected interface 9.24.104.185 down.
itso6611.itso.ral.ibm.com N Interface te1 up.
```

Figure 251. Multiple Traps for the Same Event

As expected, when we saw the duplicate nodes both MLMs were sending an interface down trap to the NetView for AIX manager. As Figure 251 shows, only

the first arriving trap from rs600013 caused NetView for AIX to generate a status event. The second trap from the other MLM is ignored.

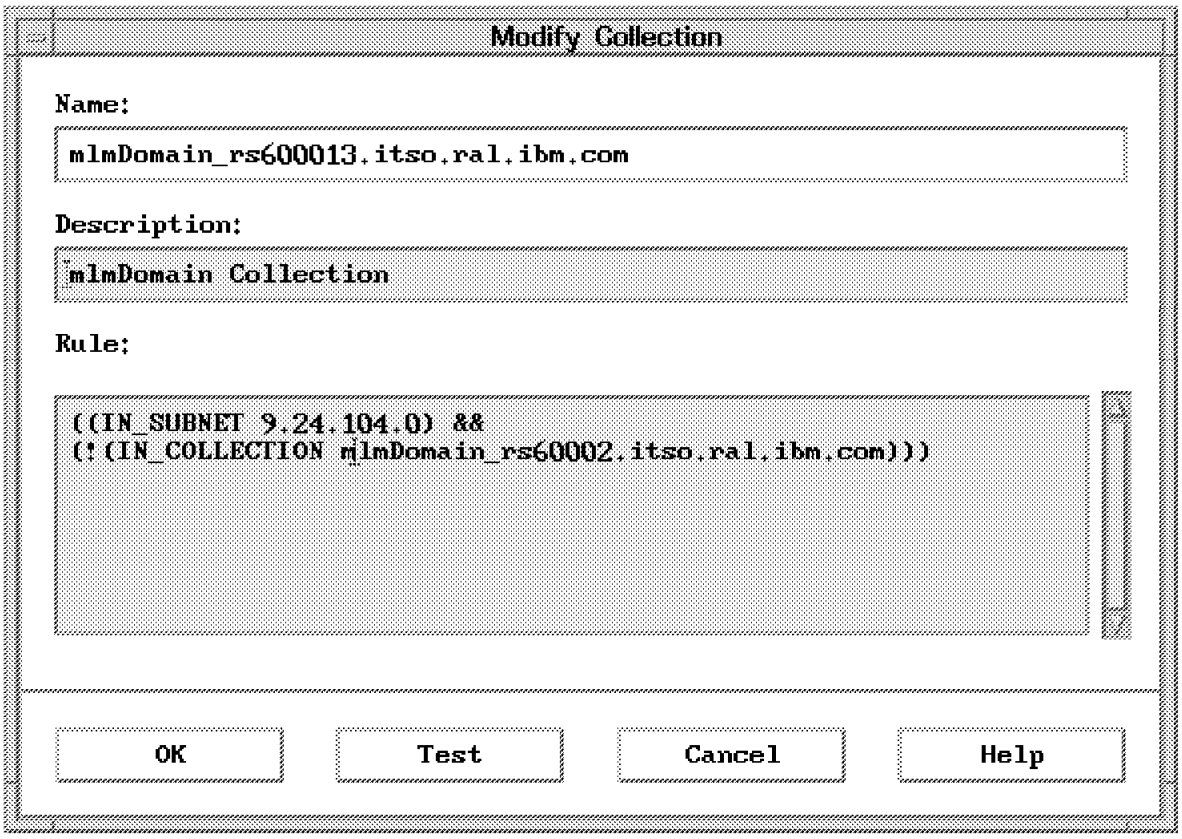


Figure 252. Corrected Collection Rule

In other words, duplicate nodes produce additional network traffic but won't confuse NetView for AIX. To resolve this situation, we need to modify the rule of the rs600013 MLM collection by a simple statement: exclude all those nodes already included in the collection rule for rs60002. The complete working collection rule can be declared as in Figure 252.

8.3.2 Rearranging MLM Domains Automatically

If an MLM disappears from the network, the Agent Policy Manager will automatically redistribute that MLM's workload. A new collection is created called `delDomain_mlmname`, where mlmname is the name of the MLM. The nodes for which the MLM was responsible are redistributed to other MLMs in the network. They are assigned according to their subnets.

If the MLM later reappears in the network, the domain is recreated and the nodes are reassigned to the original MLM.

Chapter 9. Examples Using APM for Threshold and File Monitoring Tasks

This chapter explains how to fill in the fields on the Threshold and File Monitor dialogs in the APM EUI. The dialogs are very similar to the Threshold Table and File Monitor Table dialogs in the Systems Monitor configuration application. APM just adds the ability to assign collections to the definitions and allows you to distribute the definitions to all nodes in a collection.

In this chapter you will find a number of examples, both for thresholding and for file monitoring. Some of the examples are derived from examples in Chapter 7, “Systems Monitor Examples” on page 161 and adapted to work through APM.

9.1 APM Aliases and Names

The AIX Systems Monitor/6000 MLM MIBs include an Alias Table that is used for many MLM functions. If you configure the MLMs in your network to take over local discovery and status monitoring duties from NetView for AIX, the MLM assigns aliases for groups of nodes to facilitate management of the nodes. The APM also use the Alias Table to keep track of groups of nodes. It sets aliases for collections that have thresholds set against them, and for the thresholds themselves.

After you define a threshold and distribute it to a collection, an alias is defined for this collection. Aliases for collections have names in the format {NVip-address_collname}, where ip-address is the IP address of the NetView for AIX manager, and collname is the name of the collection. For example, a collection called fileservers, assigned to the MLM by the NetView for AIX manager on address 9.24.104.109, would be assigned an alias of {NV9.24.104.109_Fileservers}.

Similarly, there is an alias for each threshold you define. Thresholds have aliases in the format {NVip-address_thresholdname}, where ip-address is the IP address of the NetView for AIX manager, and thresholdname is the name of the threshold.

File Monitor definitions don′t need an MLM and are sent directly to the particular SIA. APM creates an entry in the SIA′s File Monitor in the format {NVip-address_Monitorname}, where ip-address is the address of the APM Manager node and Monitorname is the name of the APM definition.

9.2 Managing Aliases

The alias definitions are APM′s mechanism for managing changes to collections and thresholds. Collections are dynamically updated to reflect changes in a network. When a collection is changed, the change is made to the alias definition as well. You do not have to redistribute a threshold definition if you use the Collection Editor to change the members of a collection; the APM automatically redistributes, based on the changes to the alias.

9.3 Preparing for the Examples

The following sections will show a number of examples using the Agent Policy Manager. Because all of the Agent Policy Manager operations use collections, we prepared collections to be used for the examples. To show and verify how Agent Policy Manager even distributes threshold definitions across multiple Mid-Level Managers, we assigned a few nodes from our subnet to be controlled by a dedicated Mid-Level Manager.

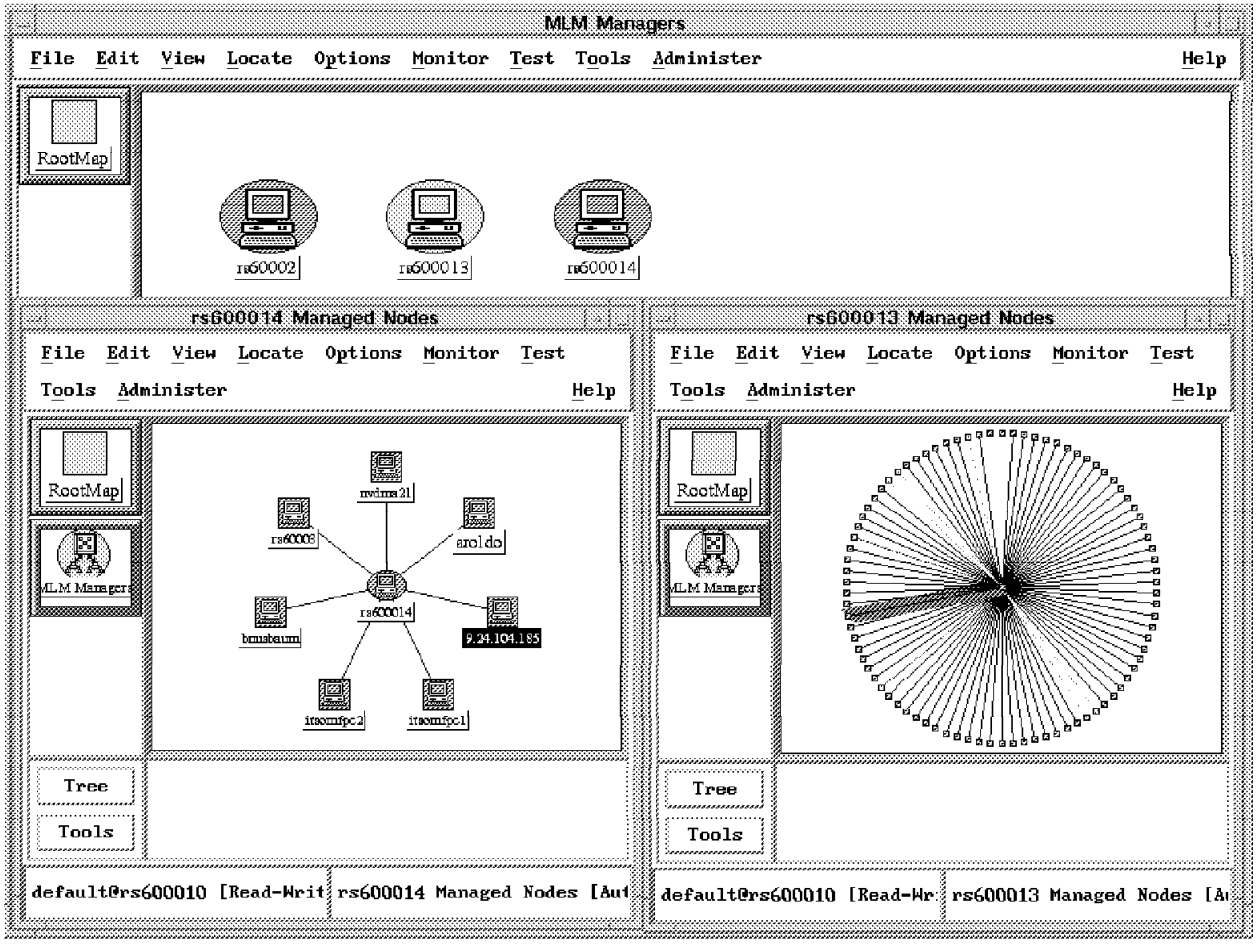


Figure 253. MLM Collections Used for the Examples

As you can see from Figure 253, rs600013 controls the majority of the nodes in subnet 9.24.104.0., rs600014 controls just the nodes of type PC and, to make the examples somewhat easier to implement, rs60003 has one dedicated AIX node. To review the collection rules, have a look at Figure 254 on page 281.

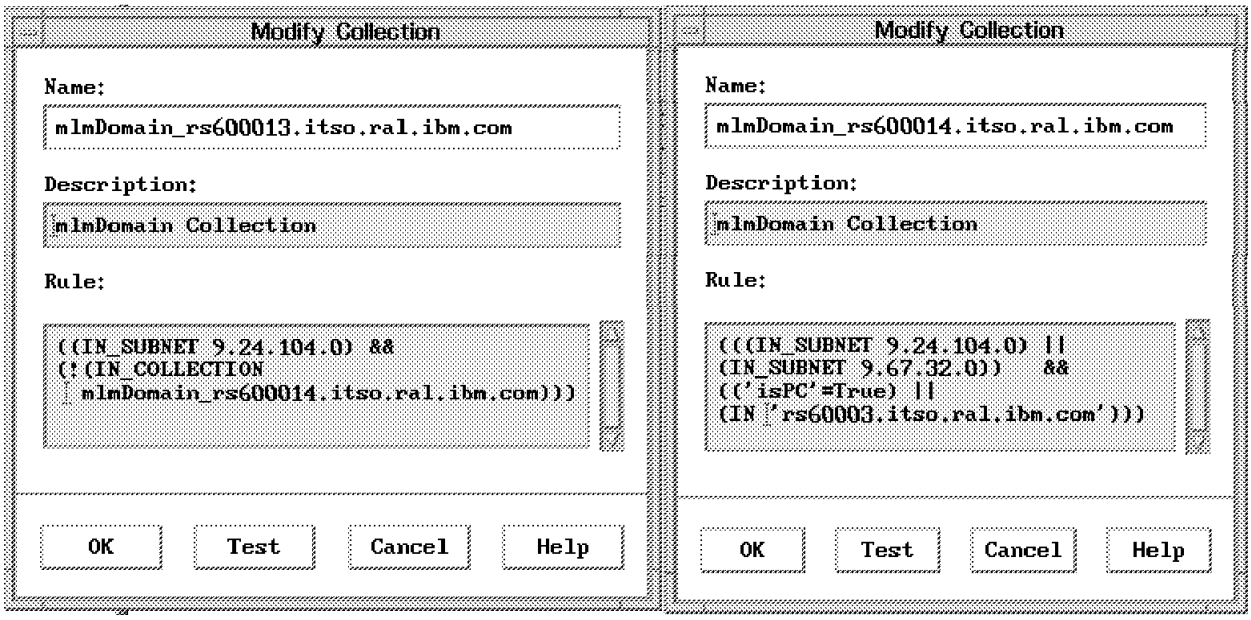


Figure 254. Rules for the MLM Collections Used in the Examples

9.4 Using APM for File Monitoring tasks

APM allows you to set up file and log monitoring on nodes having a Systems Information Agent or a System Monitor Version 1 installed. When you define a file monitor condition using the APM EUI, APM uses SNMP SET commands against the Systems Information Agents executing on the nodes defined in the collection you chose. That means, the community name for each node you want to do file monitoring must match a community name defined in the management nodes ovsnmp.conf file.

Files of any type can be monitored. Different conditions and combinations of conditions can be monitored. Each File Monitor condition returns a different trap:

Table 11. Traps Returned by File Monitors

Condition	Trap Returned
Existence of a text string	Specific trap 21 - String found Specific trap 22 - File data modified
Changes to characteristics of the file, such as owner, group, or permissions	Specific trap 23 - File status changed
Existence of a file	Specific trap 24 - File does not exist Specific trap 25 - File exists

NetView for AIX provides you with a predefined filter to display these traps, the APM Console. Refer to 1.3.3, “APM Console” on page 19 for more information.

You can select a monitor type being used against a given file. Currently available monitor types are:

string The SIA watches the specified file for the given string. The string or pattern can be any limited regular expression in the style of egrep command. If you use an anchor symbol (like $),

the affected line must be correctly terminated by a new line character (\n). If a string in the file matches the condition the specified action is performed. String monitoring is not supported for Version 1 Systems Monitor agents.

dataChange The SIA watches the file for any change to the contents of the file, such as an increase or decrease of the file size, added or deleted characters.

statusChange The SIA watches the file for any change to the status of the file, including the file owner, file group, and file permissions. You define the permissions which will be used for comparison in the File Monitor Dialog. If any of the permissions change, the specified actions are done.

strDataStatus The SIA watches for any of the previous three types of changes (data changes, the appearance of a string, or file status changes). If any of these changes occur, the specified actions are done.

notExist The SIA performs the defined action if the file under monitor disappears.

exist The SIA starts to execute the specified actions if the file appears.

all The SIA performs all types of monitoring and will do the specified action on any change to the file at all.

9.4.1 Example Using String Expressions

In this example, we show how to watch a file for a given string. A trap will be sent as soon as a string is written to the file. To show a sample you may use in your normal everyday work, we watch for SU logins to root. AIX logs each superuser login attempt in the file /var/adm/sulog. For each login, there is an entry in the form

```
SU mm/dd hh:mm [+|-] <terminal> user-suuser
```

where mm/dd and hh:mm define the date and the time, + stands for a successful login, and - for an unsuccessful login. The user-suuser lists the user who attempted the SU login and to which user he wanted to SU. In our example, we want to trap all the unsuccessful logins to root.

To start APM, select **Tools**, and then **APM Configuration** from the NetView for AIX main menu. As mentioned, it may take a while for APM to become ready to accept user interaction.

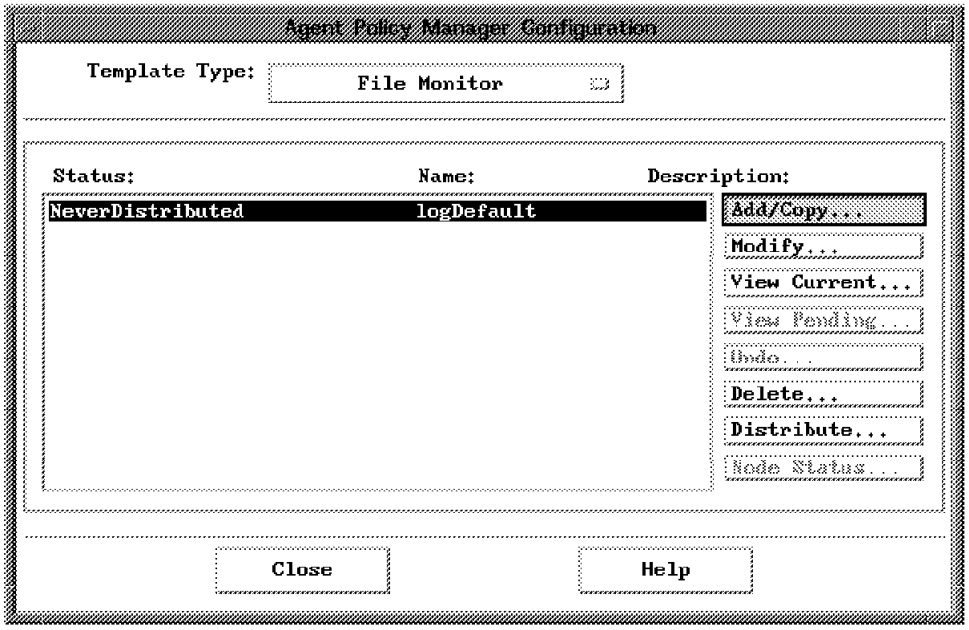


Figure 255. Initial APM Dialog

The APM main dialog allows you to select from either a File Monitor or a Threshold/Data Collection template. The dialog comes up with the File Monitor template selected by default. One default entry called logDefault already exists. You should use this entry and copy it instead of building a new entry from scratch.

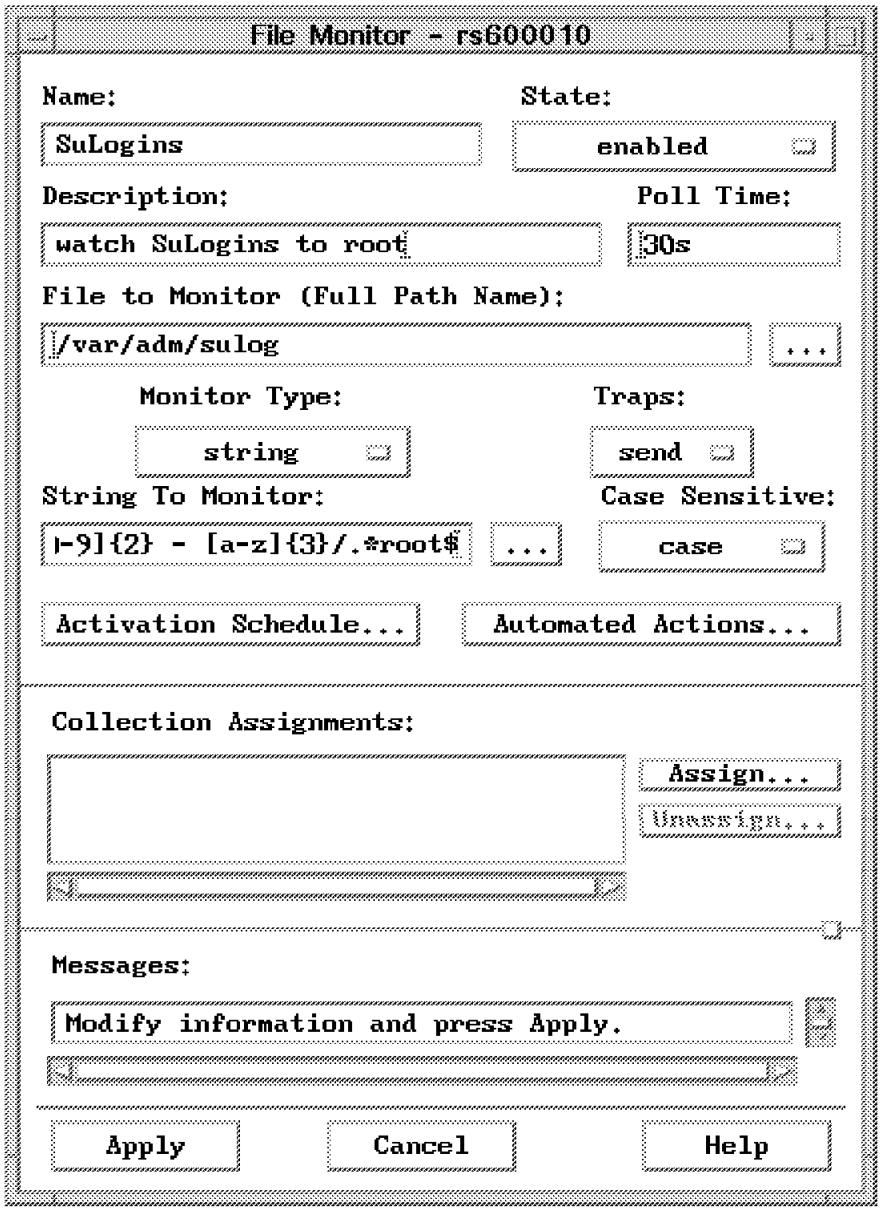


Figure 256. File Monitor Dialog

The APM File Monitor Dialog is similar to the SIA File Monitor Table. First you enter a name for this APM File Monitor. This name must be unique across all APM definitions. In other words unique for both, File Monitor *and* Threshold distributions. Then you need to provide an optional description and the poll time. The poll time defaults to 10 seconds if you leave the field empty. Otherwise you may enter the poll time in the normal AIX Systems Monitor/6000 format. In our example we define the poll interval to be 30 seconds.

Then we need to define the monitor type. Because the sulog is a flat ASCII file and each SU login attempt is written as a one line log entry into this file, we specify string as the monitor type.

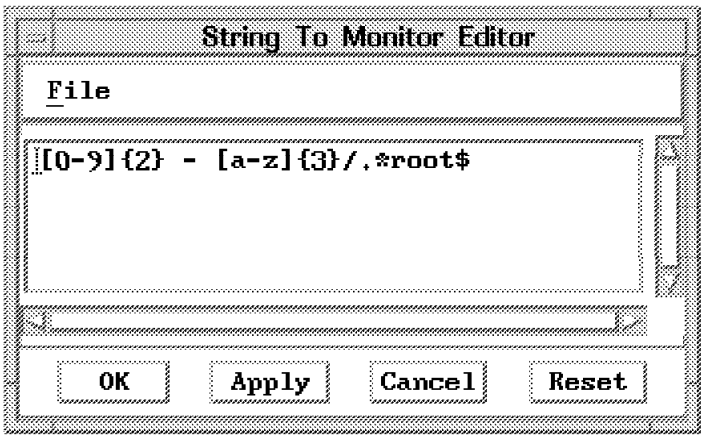


Figure 257. Regular Expression

The string can be any string or a regular expression. We want to catch just unsuccessful superuser logins to root, so we use a regular expression which filters the minus ″-″ for an unsuccessful attempt and the word ″root″. The word root can occur as the from and the to user in the sulog file. The dollar ($) character at the end of the expression limits the word root to be the last word in the line. You can enter the complete expression in the String to Monitor field. However, if an expressions gets too complicated, you may select the button marked with dots (...). This gives you a separate Editor Dialog as in Figure 257 to enter long strings in a text editor fashion.

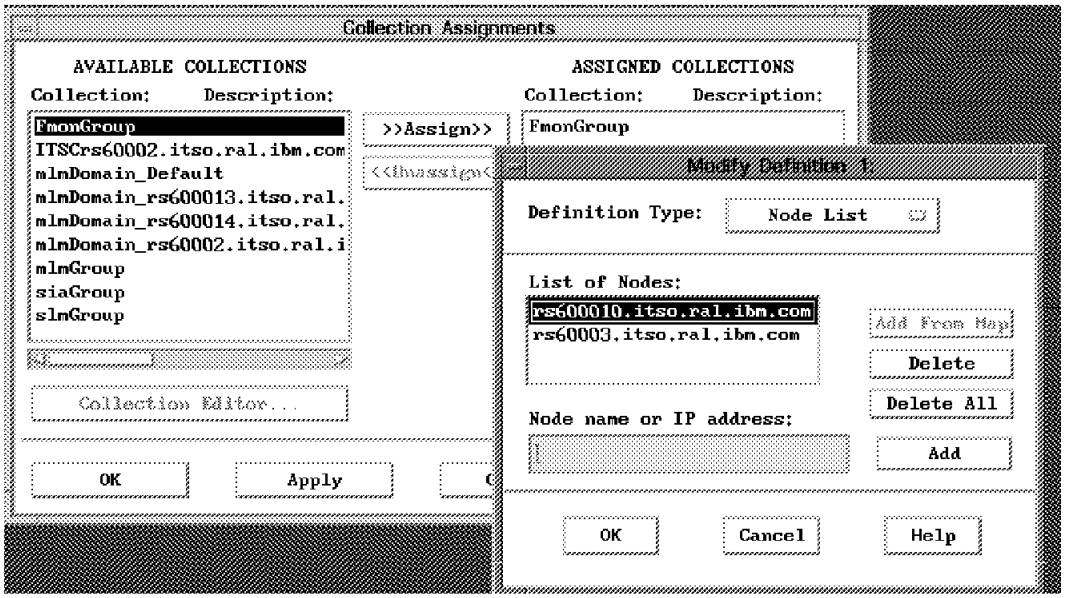


Figure 258. Assigning the Collection

Up to this point, defining a File Monitor through APM is pretty much the same as defining it with the SIA File Monitor Table Dialog. The difference with APM is the way you assign the nodes where the File Monitor will be executed. APM uses collections. For our example, we built a collection named FmonGroup containing nodes rs60003 and rs600010. Because the collection contains only dedicated nodes, we used the Node list type to define the collection. If you don′t have a collection defined which can be assigned to the APM definition, the Assign

Dialog offers a button to call the Collection Editor. You can define your collection and assign it.

This concludes the definition of an APM File Monitor. On the File Monitor Dialog, select the **Apply** button to move the definition to the APM Configuration. In this dialog you should now be able to select the new definition and distribute it.

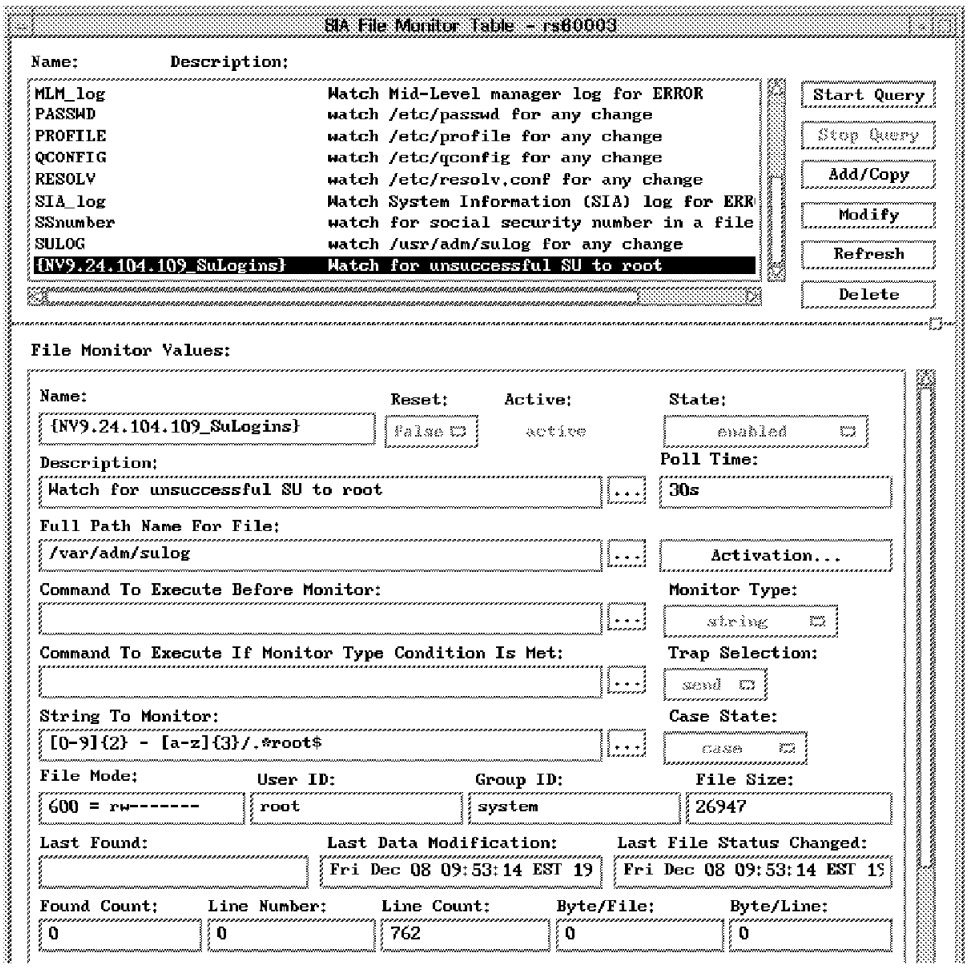


Figure 259. File Monitor Table of Target Node rs60003

The distribution process accesses all nodes defined in the collection and creates the appropriate File Monitor Table on the target node once the process ends. The APM Configuration Dialog should have the status field for our SuLogin definition set to Distributed. To verify the correct distribution, we examined the File Monitor Table of rs60003. As expected, it shows a valid entry named {NV9.24.104.109_SuLogins}.

9.4.2 Verifying the Example

A successful distributed APM File Monitor definition adds a new icon to the Interface submap of each affected node. In case a file condition is met, this icon changes its color to red in order to signal a file monitor event.

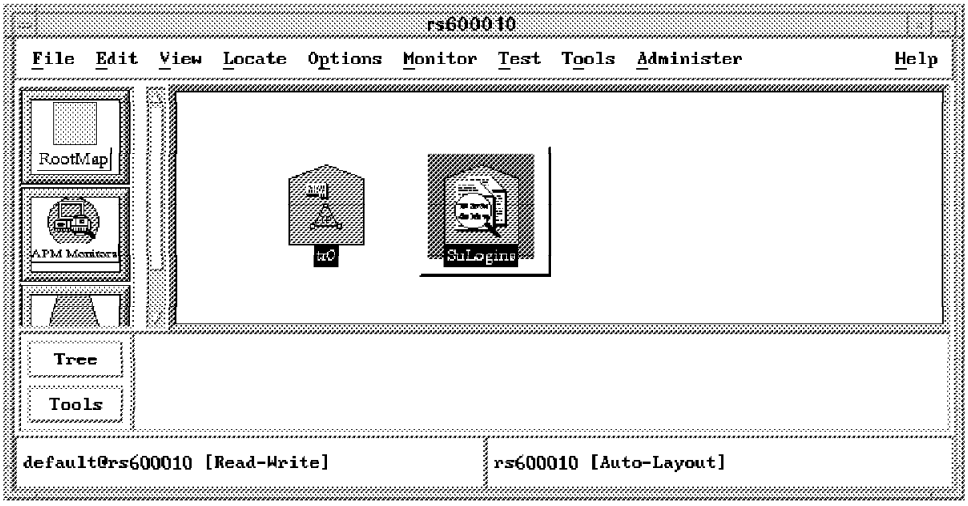


Figure 260. File Monitor Icon on the Interface Submap

Figure 260 shows the icon inserted into the submap of rs600010 and Figure 261 shows the event card generated after an unsuccessful su login has been detected.

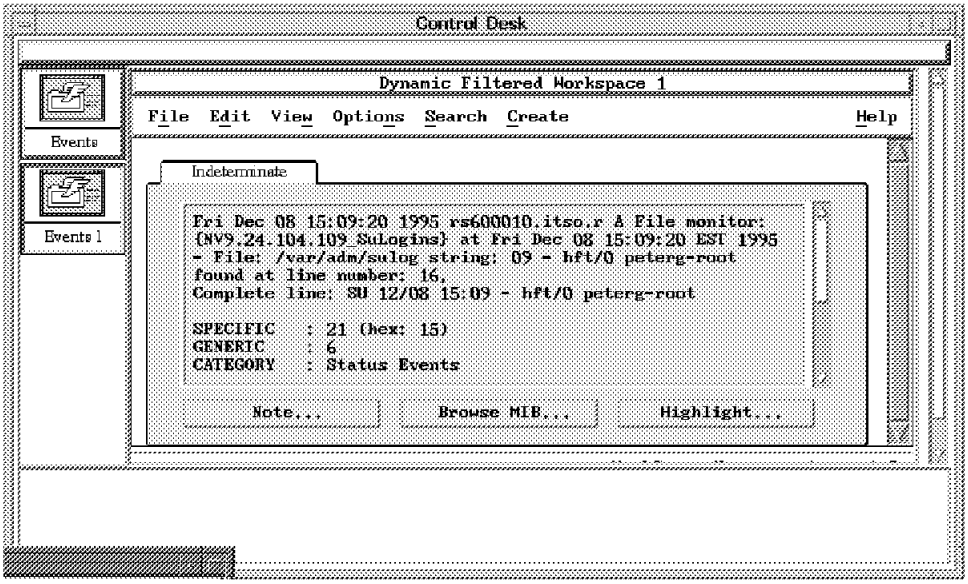


Figure 261. Event Card Generated

To reset the red icon to its original (green) color, you need to execute the connected PDA application. This helpful new tool will be discussed in 9.6, “Problem Determination Assistant” on page 294.

9.5 Using APM for Thresholding Tasks

Through the APM, you can collect important MIB data and set thresholds to send a trap or run a command when the threshold is reached. You can set thresholds against various types of MIB objects:

- MIB objects in the Systems Monitor SIA MIB extensions.
- MIB-I and MIB-II objects.

- MIB objects in private MIB extensions as long as you are able to retrieve them using SNMP get requests.

With APM, you are setting a threshold against a collection of objects. That is, you define a collection of nodes. Then APM searches each object in that collection and looks for the appropriate MLM in the network that is responsible for that object and then determines which MLM in the network has monitoring responsibility for each of the objects. APM adds the correct entry into the Threshold and Collection Table of the MLM. It doesn't matter if one or more nodes of the collection are managed by a particular MLM, APM creates both a Threshold and Collection Table entry and an Alias Table Entry containing the nodes to be monitored.

Note

With Agent Policy Manager, you cannot prepend a node name or an alias to the beginning of the MIB object ID on which you are setting a threshold. Actually, there is no need to prepend a target, since with APM you are setting thresholds against a collection and APM takes care for the distribution to each of the collection members.

9.5.1 Yet Another Filter Console

In 4.1.3.1, "A More Elegant Way to Filter Systems Monitor Traps" on page 112, we discussed a method to filter relevant System Monitor traps and route them into a Dynamic Workspace, which is called a console. Now, discussing thresholds, some more traps come to our attention.

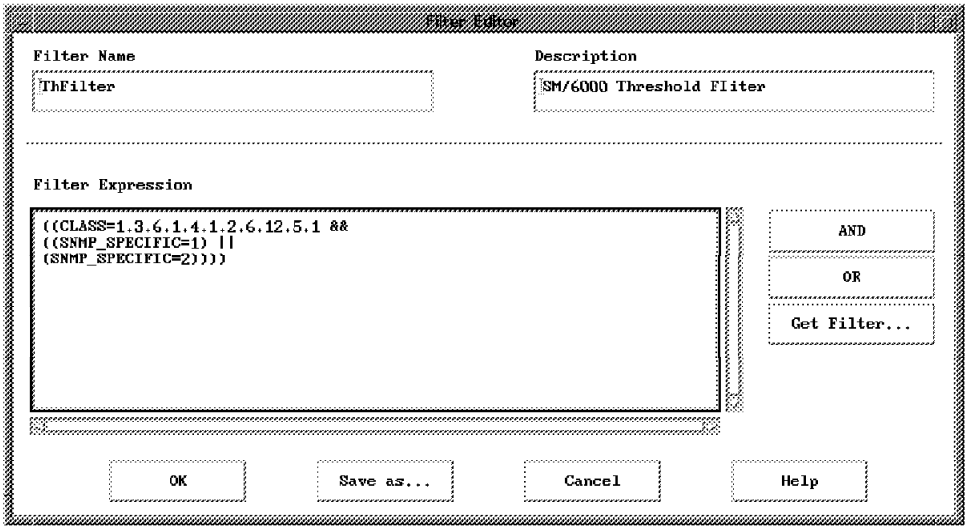


Figure 262. Filter Definition for Threshold Traps

To signal threshold arm and rearm events, AIX Systems Monitor/6000 uses a separate group of traps. They are assigned an object ID (or Class) of 1.3.6.1.4.1.2.6.12.5.1. In this class, only trap numbers 1 (one) for a threshold arm event and 2 (two) for a threshold rearm event are used. If you would like to have these traps in a dedicated dynamic workspace, you should set up the filter shown in Figure 262 using the procedure described in 4.1.3.1, "A More Elegant Way to Filter Systems Monitor Traps" on page 112.

9.5.2 Threshold Setup Example

To demonstrate the capabilites of APM to distribute thresholding tasks to MLMs we use a collection called THGroup. We made sure, the collection contains nodes managed by different MLMs.

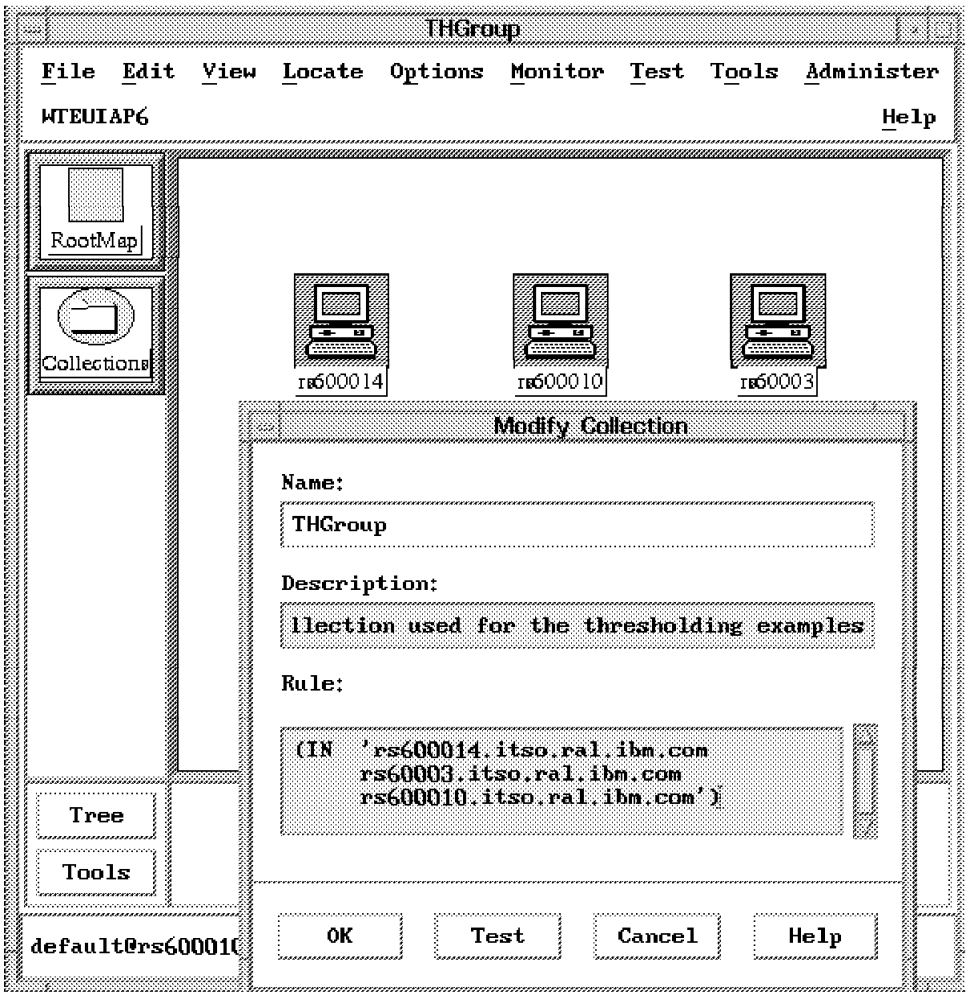


Figure 263. Collection Definition Used for Thresholding

The definition itself is rather simple: the members of the collection are selected by a single Node list type statement as Figure 263 shows.

The threshold example itself is a modification of the example in 7.3.1, “Monitoring Paging Space” on page 220. This example shows how to monitor paging space on a group of nodes. If you would use this ″classic″ way to add a threshold you would have to manipulate three tables on each MLM in the network that manages affected nodes. Using APM, you will be able to define everything in just one dialog.

As with the File Monitor examples, we suggest you start the APM Configuration, select the Threshold/Data Collection template type and use the Add/Copy button. Using the already existing default entry is safer than providing a new entry from scratch. This will bring up the APM Threshold and Data Collection Dialog.

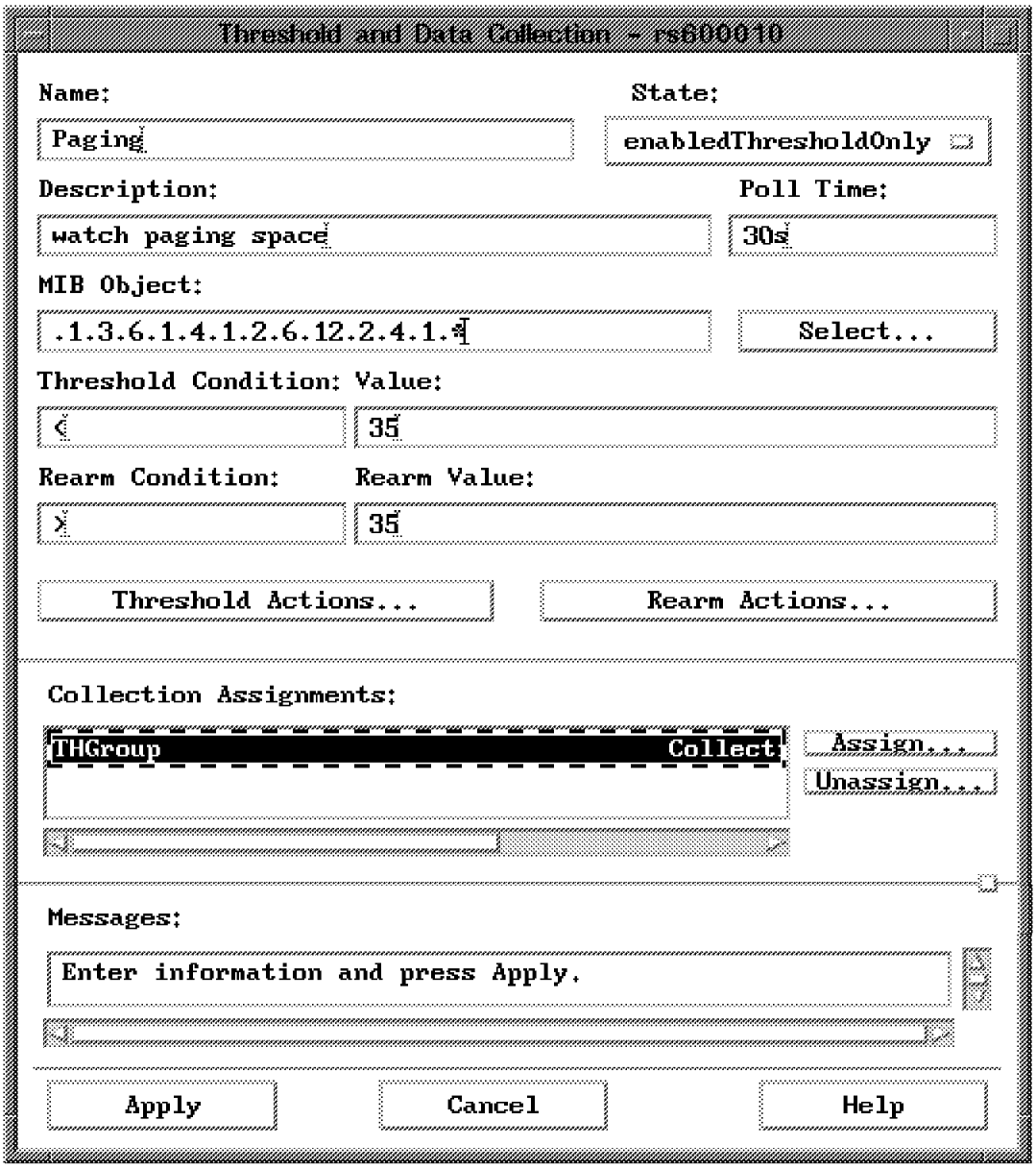


Figure 264. Threshold/Data Collection Dialog

The dialog is very similar to the MLM Threshold and Data Collection Table. However, because you don't use APM to display dynamic data, all the fields that are filled in by a query operation are omitted.

The threshold definition generates an arm trap as soon as the paging space of a node in the collection goes below 35 MB. The rearm trap will be produced if the paging space reaches any value above 35 MB. Figure 264 shows the completed dialog. No actions should be executed at the time a threshold occurs, you only need to assign a collection. We use the previously defined THGroup collection. You then choose **Apply:** to save the definition and make it ready for distribution.

Note

The definition used in 7.3.1, “Monitoring Paging Space” on page 220 uses a MIB variable which is not available across all platforms on which AIX Systems Monitor/6000 is available. For this example we changed the variable to another MIB variable in the smSiaSysteminformation tree. smSiaSystemPagingInformation.smSiaSystemFreePagingSpace returns the overall free paging space in MBytes counting the free space on all data sets dedicated for paging. This variable exists in the MIB database across all platforms.

The APM Configuration Dialog now shows a new template called Paging which has a status of NeverDistributed.

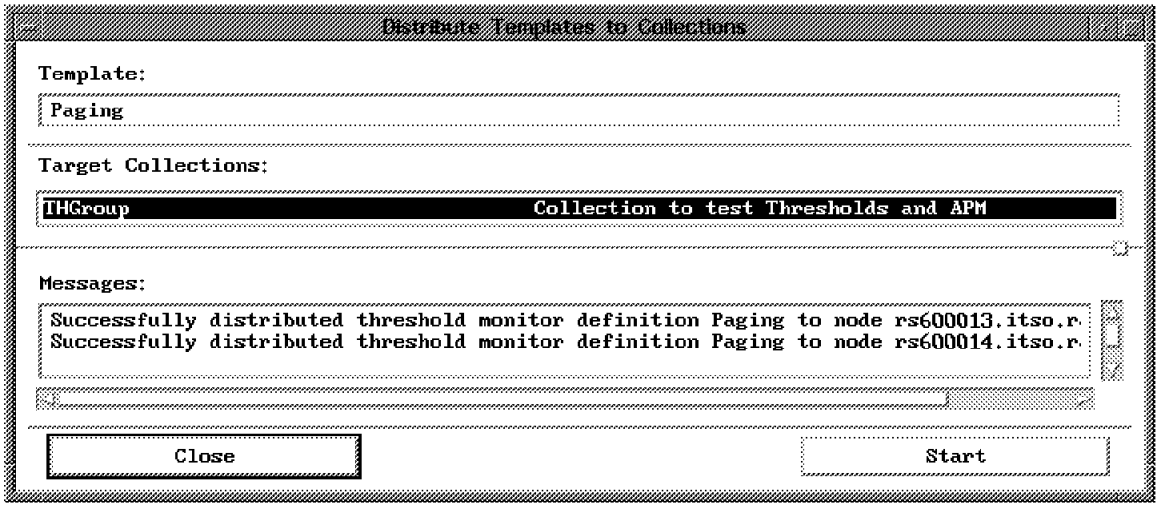


Figure 265. Successful Distribution of a Threshold

To distribute this collection, select the template and click on **Distribute**. APM reports the results of a distribution in a separate dialog similar to Figure 265. The Dialog shows a successful distribution to two nodes. If you have a look at Figure 263 on page 289, we defined three nodes to be part of that collection. Thresholds are handled by Mid-Level Managers. One node in our collection, rs60003 is managed by the Mid-Level Manager executing on rs600013. The other two nodes in the collection are managed by the MLM on rs600014.

Having a look at the tables maintained by the MLM shows the modifications APM made to the MLM to get the threshold working.

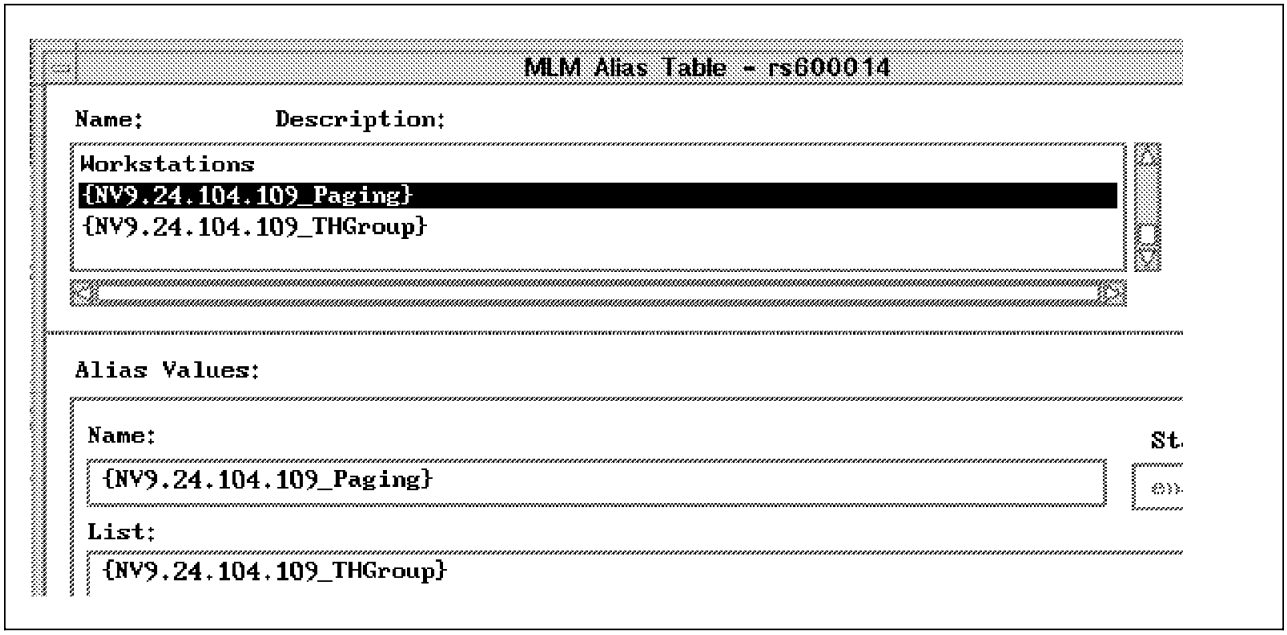


Figure 266. First Alias Table Entry of rs600014

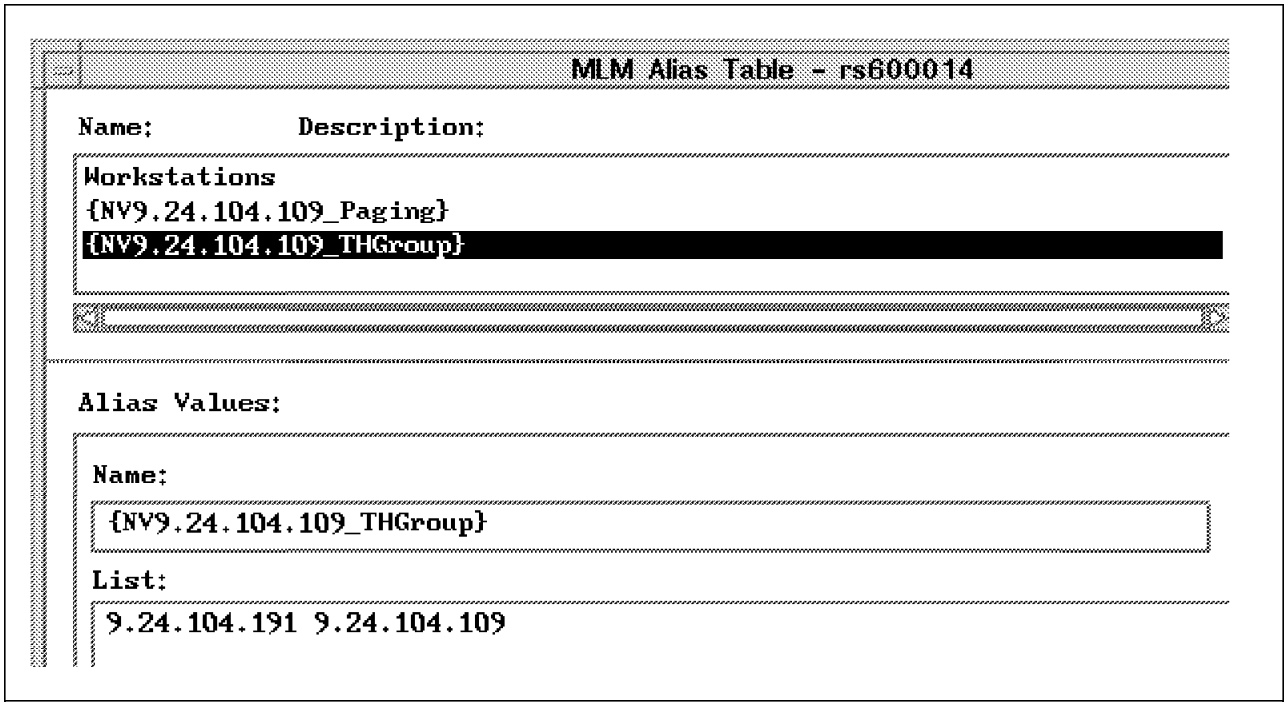


Figure 267. Second Alias Table Entries of rs600014

APM added two entries to the Alias Table. The first one, {NV9.24.104.109_Paging}, just points to the second entry called {NV9.24.104.109_THGroup}. The latter one resolves the SIA nodes that the threshold will be used against.

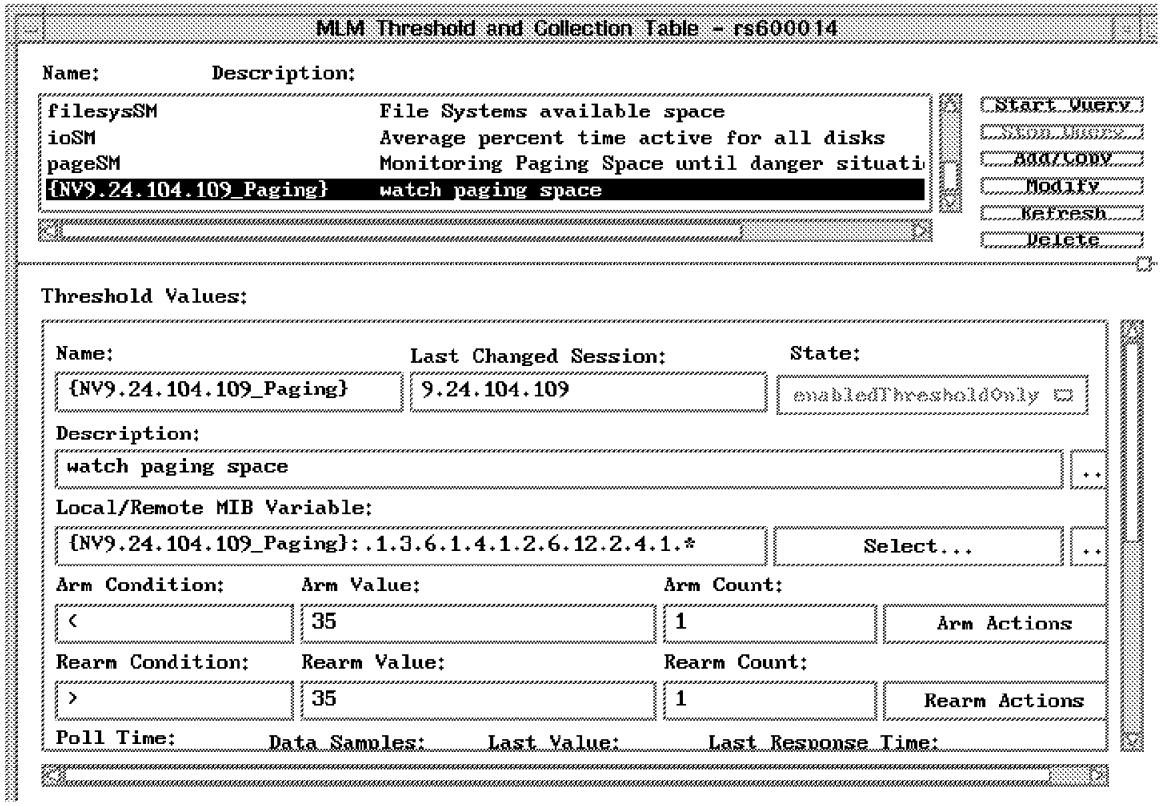


Figure 268. Threshold Table Entry of rs600014

Figure 268 finally shows the Threshold Table entry. It uses all the data you entered in the APM Configuration. Note that the MLM prepends the {NV9.24.104.109_Paging} Alias Table entry to the MIB variable. The use of different aliases for the collection and for the threshold itself makes subsequent changes very easy. If the nodes in the collection change for whatever reason, APM just updates the collection alias across the MLMs.

9.5.2.1 Activating a System Level Manager

You saw in the example how APM distributes threshold definitions across its network and takes care that the tables of the affected Mid-Level Managers reflect the definitions. Now, what happens when a System Level Manager is executing on one of the APM collection members? System Level Manager, always running together with a Systems Information Agent on a single node, is capable in handling thresholds for its local Systems Information Agent.

To verify how APM distributes a threshold definition in a mixed Mid-Level Manager and System Level Manager environment, we activated an SLM on node rs600010.

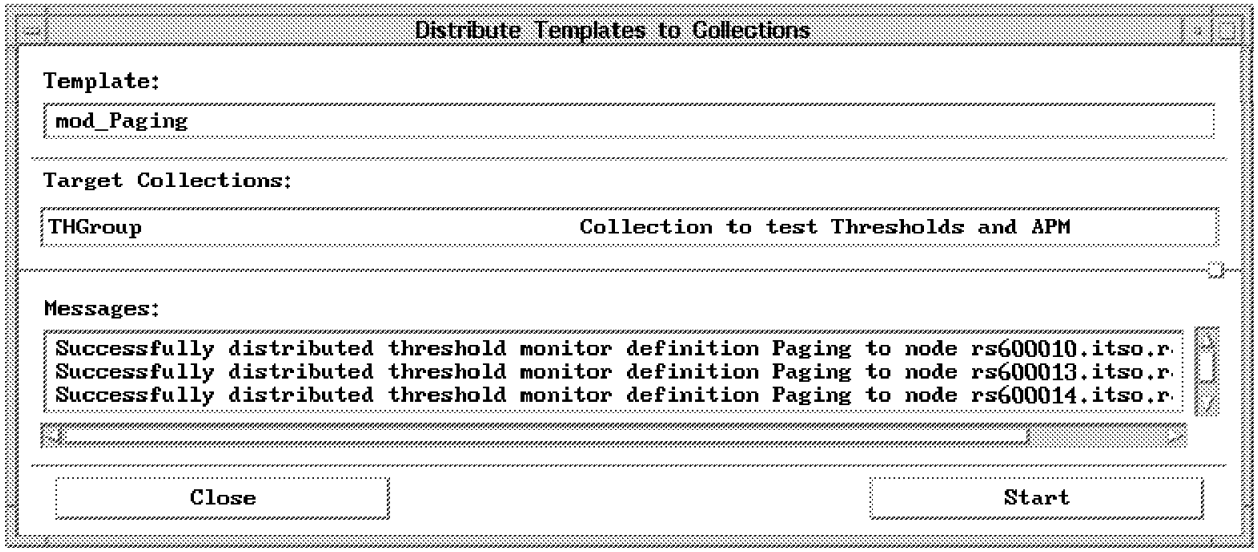


Figure 269. Distribution List After Activating an SLM

To force another distribution of our threshold definition, we modified the Poll Time of our threshold and redistributed the definition. Figure 263 on page 289 shows the result. The System Level Manager was detected by APM and the threshold was distributed correctly to the SLM.

9.6 Problem Determination Assistant

APM helps to distribute File Monitor and threshold definitions to remote nodes in the network. It adds or modifies relevant table entries on both remote SIAs and MLMs. On NetView for AIX, APM adds symbols to the Node submap an APM is issued against. The symbols reflect the current status of the APM definition. They change from green to red when either a File Monitor condition is met or a threshold trap arrives.

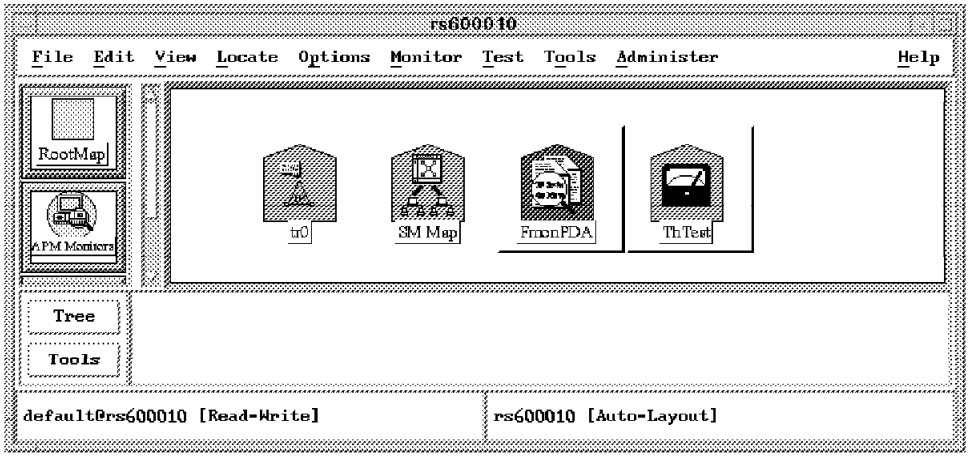


Figure 270. Icons Inserted by APM

In the current version of NetView for AIX, APM uses two different symbols to represent the type of distribution. Unlike the normal card type symbols on that submap, the APM symbols are marked executable. You can identify executable symbols by their button-like style.

When you double-click a normal symbol, ″explodable″ in NetView for AIX terminology, NetView for AIX opens the child submap connected to the symbol or offers to create a submap if no submap is connected. If you double-click on an executable symbol, NetView for AIX starts and executes the application connected to the symbol. You can mark a symbol as executable by using the symbols context menu (use the right mouse button and click on the symbol). In the menu you choose **Edit**, then **Modify/Describe**, and then **Symbol...**. Use the resulting Dialog to configure the symbols.

However, APM already makes its symbols executable and connects an application to the symbols. This application, called PDA or Problem Determination Assistant, is part of the C5eui application and registered to NetView for AIX in the /usr/OV/registration/C/C5eui registration file. If you double-click an APM symbol, the Problem Determination Assistant is started and displays a dialog showing information which depends on the type of APM definition.

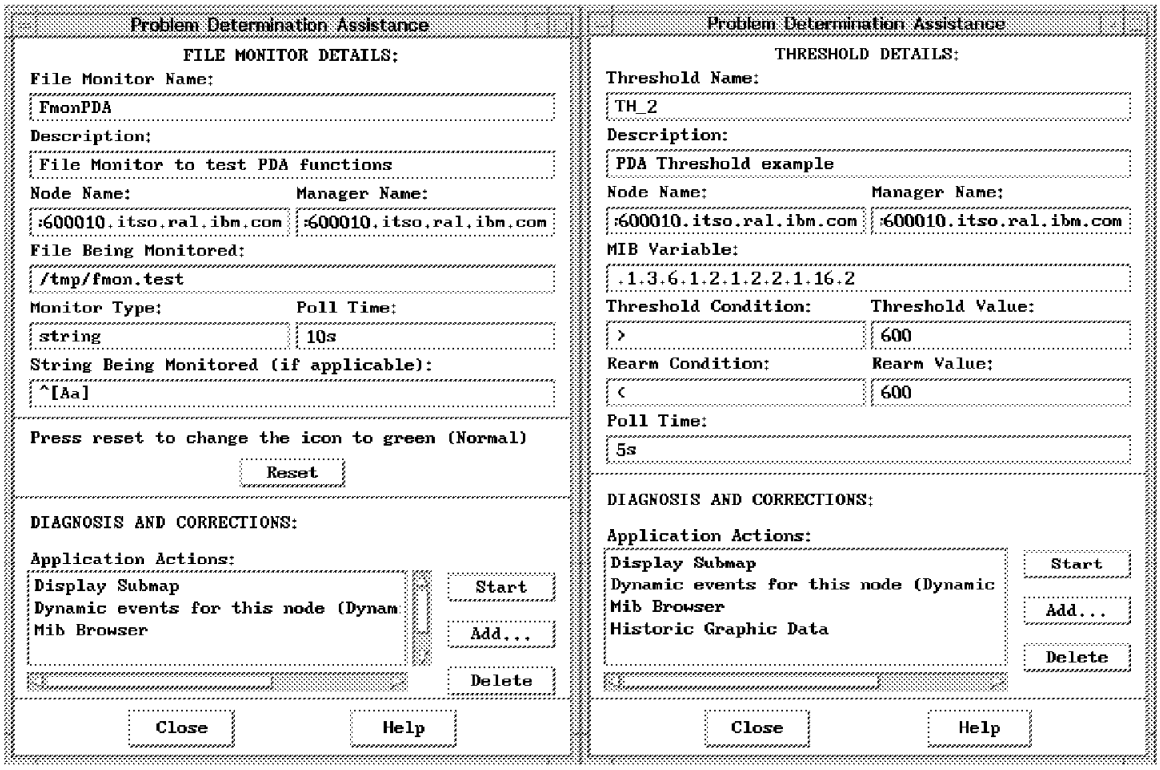


Figure 271. Two PDA Dialogs

9.6.1 Problem Determination Assistant Lab Setup

To demonstrate the capabilities of the Problem Determination Assistant application, we set up a simple lab setup. All the relevant members of the setup NetView for AIX, the Mid-Level Manager and the Systems Information Agent are running on the same node. To see how NetView for AIX and Mid-Level Manager can coexist on the same node, you might review 1.3.1, “TCP Port 165 to Prevent Port Conflicts” on page 16. Further, we provided two APM definitions, one File Monitor and one threshold example. Both definitions use the same collection which contains only the rs600010 node we used for the demonstration.

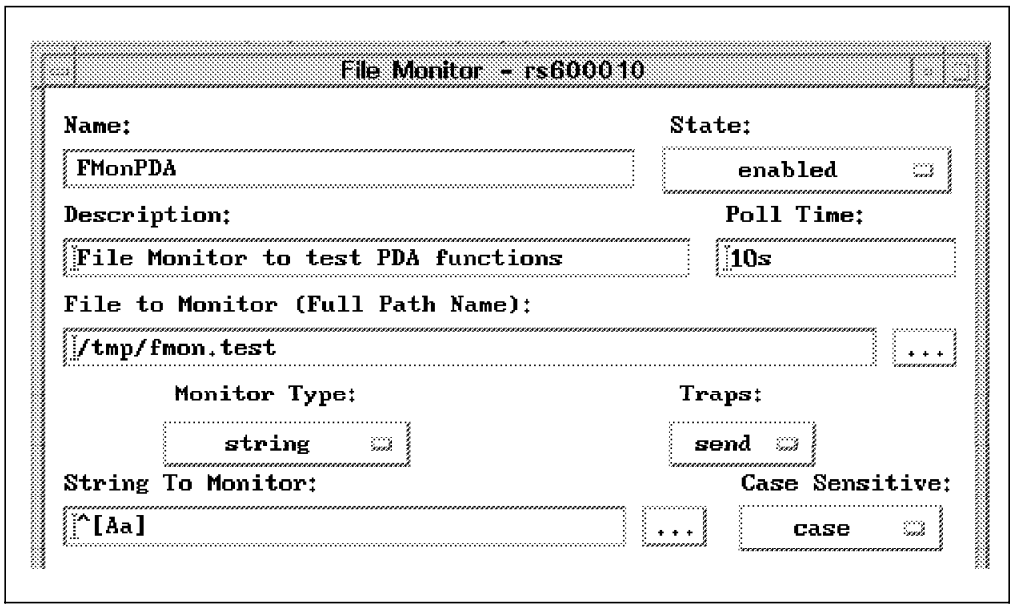


Figure 272. File Monitor Definition

Figure 272 shows the File Monitor definition. This file monito checks a file /tmp/fmon.test every 10 seconds for an upper case or lower case ″A″ at the beginning of a line.

Note

If you look at the File Monitor examples supplied with the Systems Information Agent File Monitor Table, you will find a similar File Monitor called ANCHOR1. Unfortunately, other than stated in the description, ANCHOR1 will only catch upper case ″As″ at the beginning of a line. We use an expression to ensure both cases are recognized.

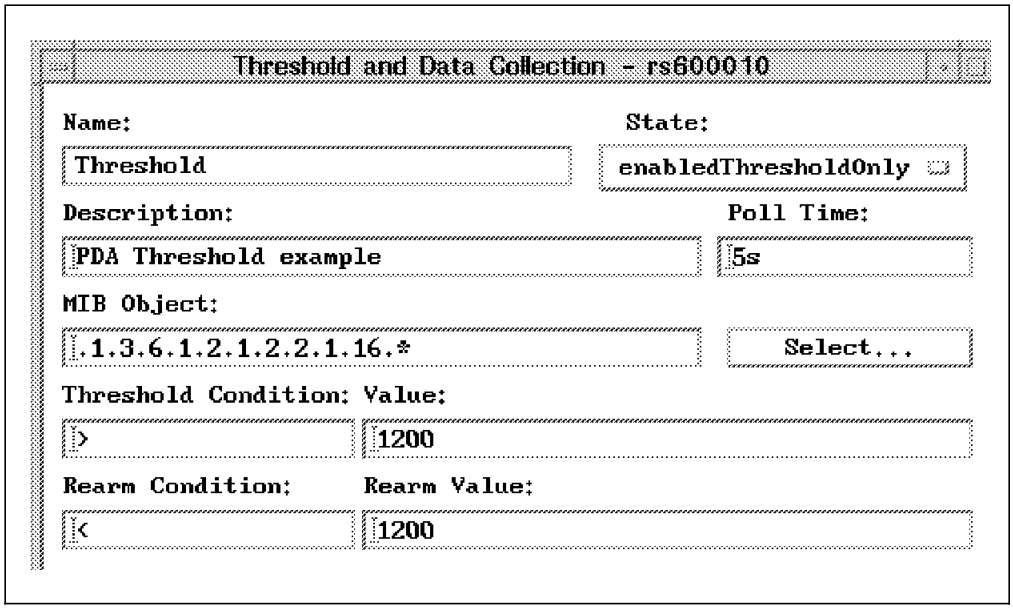


Figure 273. Threshold Definition

The threshold definition in Figure 273 samples a MIB-Variable in the (standard) MIB-II tree every 5 seconds.

The *.mgmt.mib-2.interfaces.ifTable.ifEntry.ifOutOctets* counts the bytes transferred over an interface. We set the arm and rearm threshold to a value which is around the rate of bytes transferred across the interface during two samples. In our demonstration, the value was around 1200 Bytes. This makes sure that the definition produces a reasonable amount of arm and rearm thresholds.

9.6.2 Problem Determination Assistant and File Monitors

If you set up a File Monitor to watch a string in a file with APM, a matching condition will result in a generic 6, specific 21 trap. This trap is of type status events, which causes the status to be distributed to all symbols. In other words, the File Monitor symbol turns red to signal a matching condition.

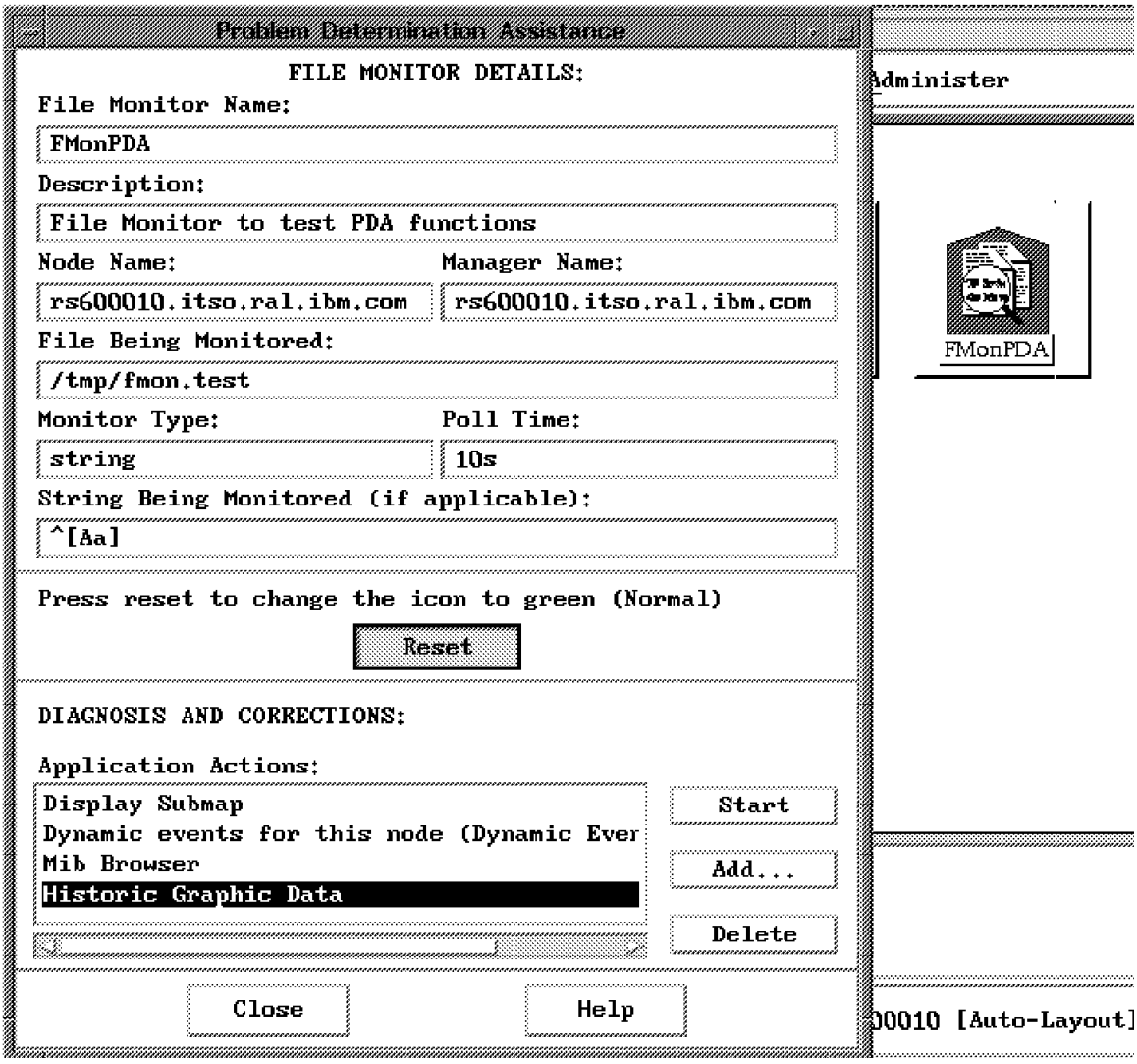


Figure 274. PDA Dialog for the Example File Monitor

Because File Monitor traps are single traps, there is no way to reset the symbol automatically to its initial state. At this point, the Problem Determination Assistant comes to action. Double-click the PDA symbol. This will bring up a dialog already containing your actual File Monitor definitions for easy

identification. Below the displayed definition, you can locate a button called Reset. A single click on **Reset** changes the symbol back to its initial state.

9.6.3 Problem Determination Assistant and Thresholds

Thresholds are done by Mid-Level Managers and System Level Managers. They compare a MIB-Variable against a fixed value for both arm and rearm conditions. For an arm condition, a generic 6 specific 1 trap along with an object ID of SM/6000_Threshold is submitted to all trap destinations. For a rearm condition, a generic 6 specific 2 trap with the same object ID is sent. Having two different traps available for thresholds, allows the dynamic update of a symbol to reflect if it reached the arm condition or if the actual value of the variable is below the rearm value. As with the File Monitor symbol, double-click the symbol to bring up the PDA Dialog.

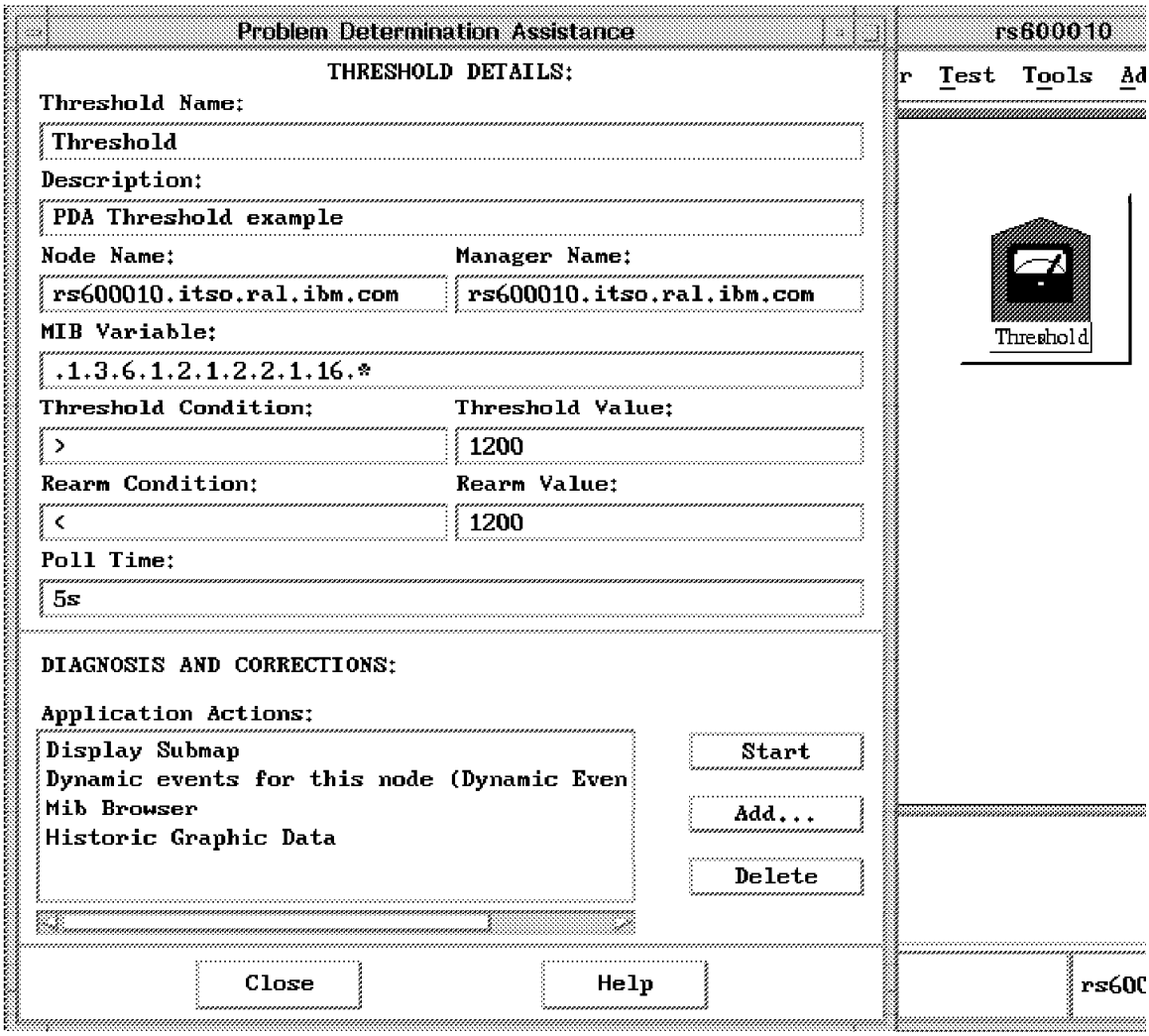


Figure 275. PDA Dialog for the Example Threshold

Figure 275 shows the PDA Dialog for our example threshold. The symbol status is updated depending upon the most recent threshold trap, so you don′t need a reset button. For thresholds, the bottom part of the dialog, DIAGNOSIS AND CORRECTIONS, is much more interesting. It lists a few useful actions which you use to further analyze the event. The next section, 9.6.4, “Problem

Determination Assistant Application Actions" on page 299 shows how this list can be configured. By default, it offers you the following actions:

Display Submap This action brings the actual node submap to the top. This action is sometimes useful to find the submap among those dozens of windows on the screen.

Dynamic events for this node Opens a dynamic workspace for all events coming from the selected node. PDA creates the workspace in your Control Desk if present. Otherwise it creates a separate window for the workspace.

MIB Browser Start the NetView for AIX MIB Browsing facility. The Application Action places the MIB Browser Dialog to the top of the MIB subtree where the thresholded MIB variable is located.

Historic Graphic Data APM stores the most recent 200 threshold events and their values (refer to 8.1.1.1, "Configuring APM (C5d) Daemon" on page 261). If you select this action, PDA will produce a graph of the currently logged threshold values.

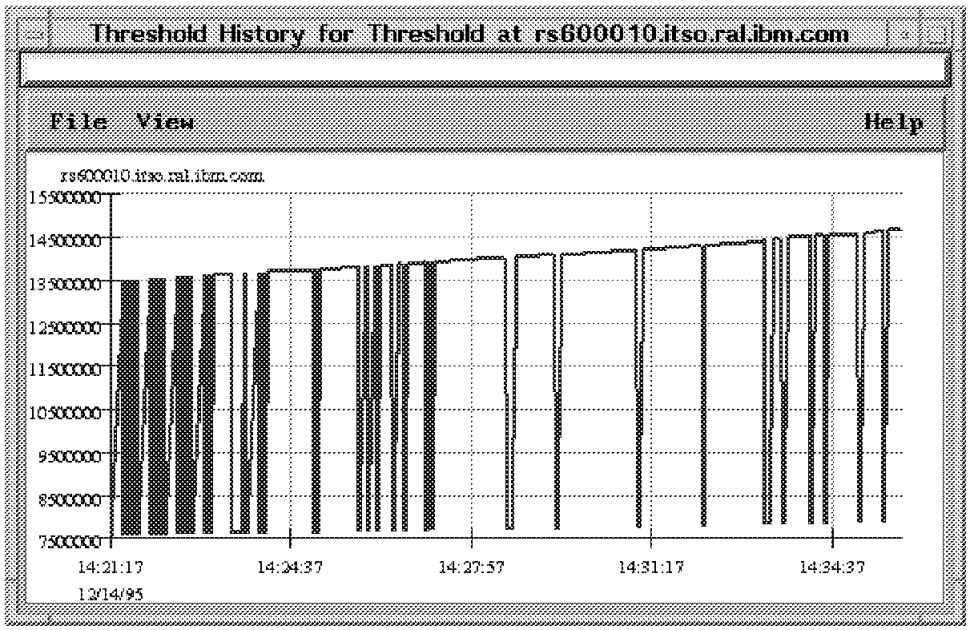


Figure 276. History Graph Showing Actual Threshold Values

Please note, that this graph is static. It does not get updated when new thresholds arrive.

9.6.4 Problem Determination Assistant Application Actions

By default, Problem Determination Assistant provides the four standard application actions discussed in 9.6.3, "Problem Determination Assistant and Thresholds" on page 298. Additionally, you can add or remove application actions from the list of application actions.

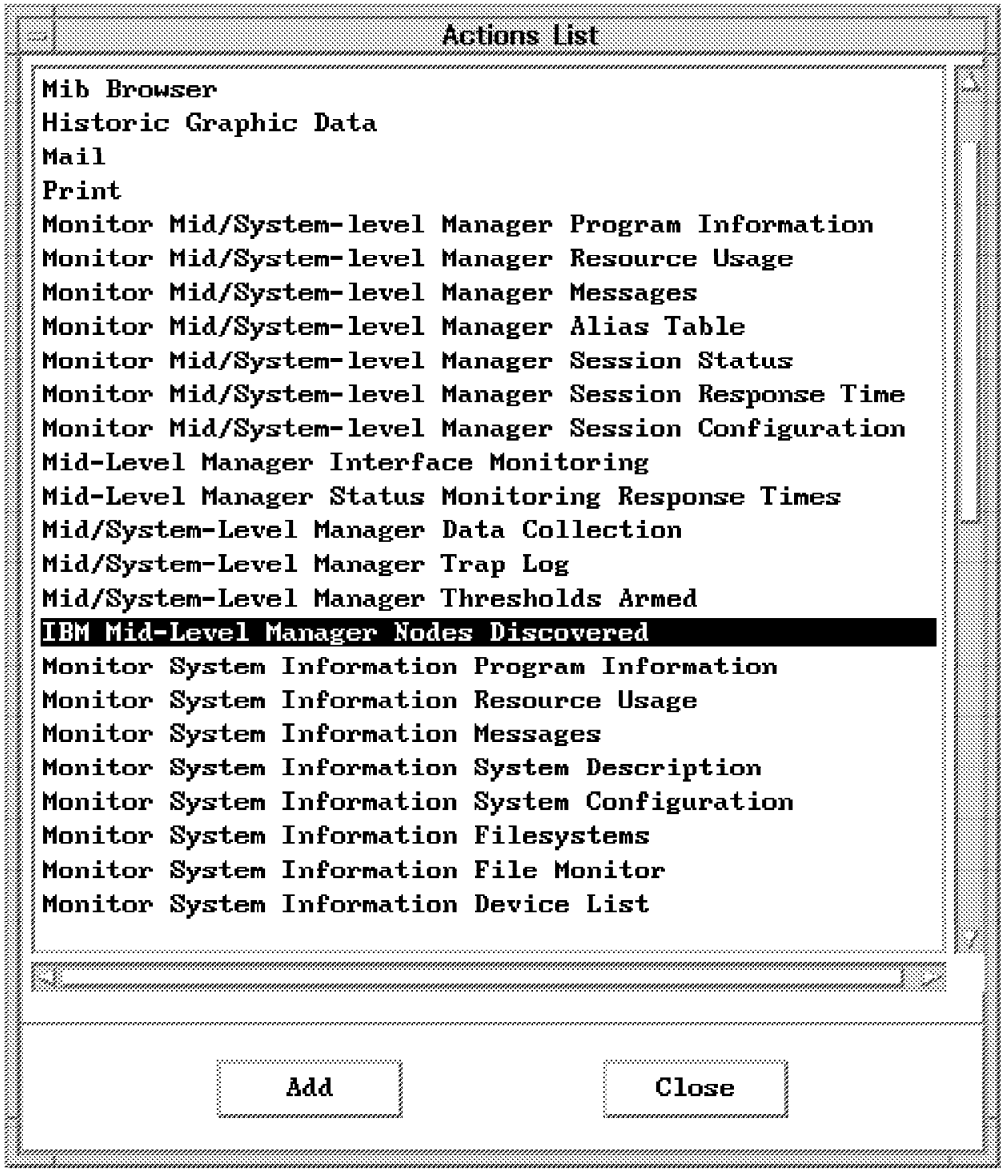


Figure 277. Actions List

Clicking on the **Add** button gives you a selection list similar to Figure 277. You may choose any application out of the list and add it to the PDA Application Actions list. The list of applications resembles those applications you find in the Monitor, Mid/System-Level Manager, Monitor, and System Information menu trees of NetView for AIX. The PDA Application Actions list provides you with a convenient fast path to those applications.

9.6.4.1 Extending the PDA Application Actions List

You can access all the entries in the Problem Determination Assistant Application Actions list via the NetView for AIX menus. NetView for AIX also offers you to build custom MIB applications and integrate them into the menu tree.

The tool NetView for AIX provides for this task is the MIB Application Builder which you can access from the NetView for AIX menus via Tools, and then MIB Application Builder: SNMP. We tried to find out whether applications may be

added to the Problem Determination Assistant Application Actions list with the help of the MIB Application Builder. Here is a way to extend the Problem Determination Assistant Application Actions list. This approach will be investigated further in the future.

Except for the automatically supplied actions, all the entries that show up in the Application Actions selection are added by the Systems Monitor Smconfig application upon installation. The installation process adds a number of registration files into /usr/OV/registration/C/ovmib. All the registration files have a file name in the form of the following where xx is a number:

```
smMlmApplications-xx or
smSiaApplications-xx
```

If you provide a registration file name having this format *and* put it into /usr/OV/registration/C/ovmib, the application will show up in the Problem Determination Assistant Application Action list. If you use the MIB Application Builder, the generated registration file will be stored in the correct subdirectory.

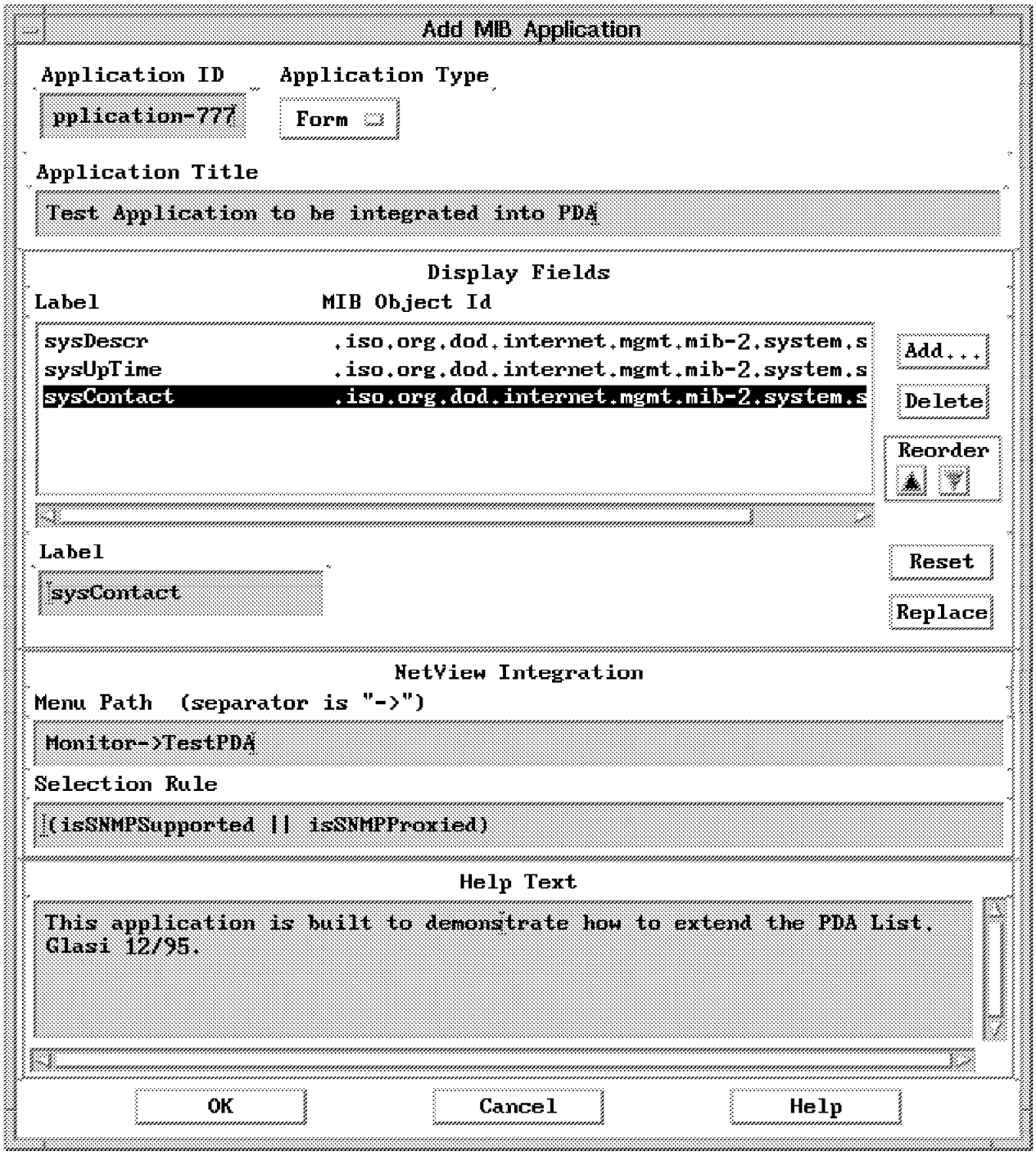


Figure 278. New Application for the PDA Application Actions List

Figure 278 shows a typical MIB Application Builder dialog. The only thing you have to provide is the correct Application ID in the editable field shown in the upper left-hand corner of the dialog. The Application ID you enter will form the file name of the registration file. We gave it the name smMlmApplications-777. The remaining fields of this demonstration only define a form which will list the contents of sysDescr, sysUpTime and sysContact. If the dialog is complete, click on **OK** to submit the MIB Application.

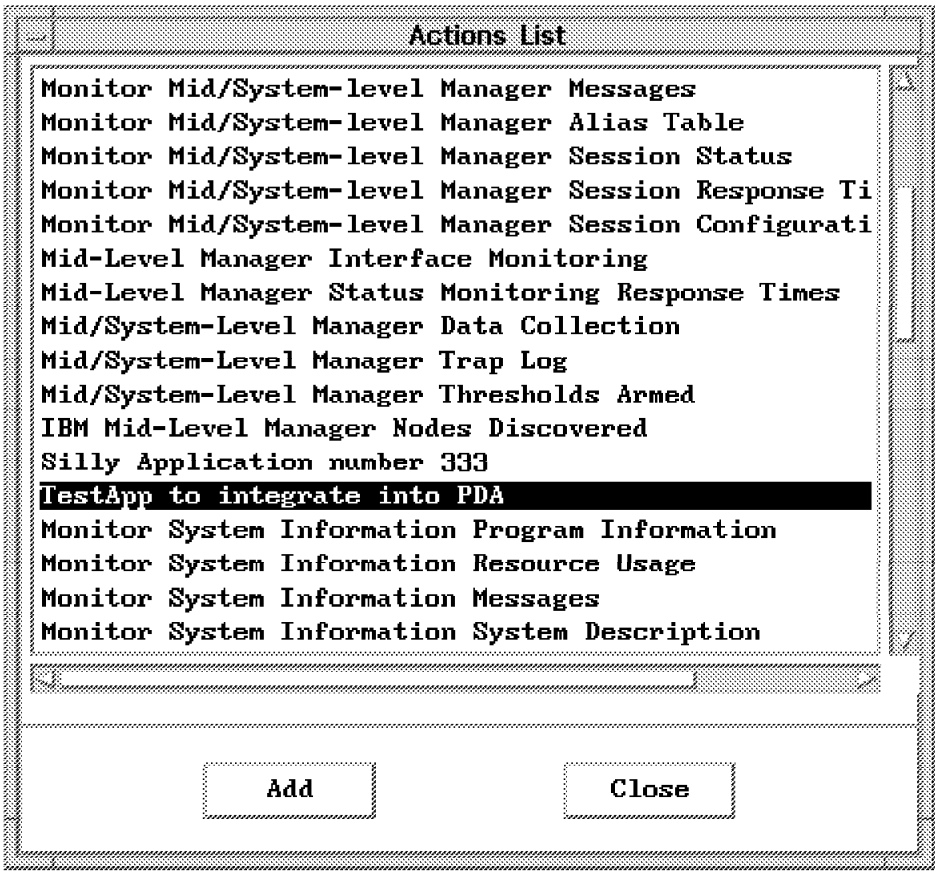


Figure 279. New Application Finally in the List

You should now be able to find the description of the new MIB application when you click on **Add** from the Problem Determination Assistant dialog. You can add the application to the list of application actions and execute it against the affected node.

Appendix A. SIA MIB Field Meanings

The System Information Agent (SIA) has a large extended MIB. For effective system management, we have to select the most useful objects from it. The following table describes the different groups within the SIA MIB and summarizes their functions. The Command column shows the command that could be issued locally to find the information provided by the MIB variable.

Table 12. SIA Extended MIBs Field Meanings

Group		Contents	Example	Command	MIB
System Description		The system information of the SIA machine.	Hostname, OS, Version, CPU_ID etc.	uname etc.	smSiaSystemDescription...
System Configuration		Configuration information of the operating system on the SIA machine.	A number of procs, page size, file table size etc.	lsattr -E -l sys0 etc.	smSiaSystemConfiguration...
System Device	List	Device information about the SIA machine.	Installed device name, desc, VPD, attribute and location etc.	lsdev -C, lscfg -v etc.	smSiaSystemDeviceList...
	Token-Ring	Token-ring device and detailed performance information of the SIA machine.	Token-ring device name, Attribute, VPD, RDTO, MAC addr, Detailed performance data etc.	lsattr -E -l tok0, netstat etc.	smSiaSystemDeviceTokenRing...
	Ethernet	Ethernet device and detailed performance information of the SIA machine.	Ethernet device name, Attribute, VPD, RDTO, MAC addr, Collision, Detail performance data etc.	lsattr -E -l ent0, netstat etc.	smSiaSystemDeviceEthernet...
	X.25	X.25 device, configuration and detailed performance information of the SIA machine.	X.25 configuration information and Performance data etc.	lsattr etc.	smSiaSystemDeviceX25...
System Paging Informa-tion	Free Paging Space	Information about free paging space on the SIA machine.	Free paging space etc.	svmon, vmstat lsps etc.	smSiaSystemFreePagingSpace...
	Paging Space	Attributes of the paging space.	vg, pv name of the paging space etc.	lsps etc.	smSiaSystemPagingSpace...
	Paging Statistics	Various paging statistics about the SIA machine.	Count of pagein, Pageout and Pagefault etc.	vmstat etc.	smSiaSystemPagingStatistics...
System File System		File system information about the SIA machine.	File system name, size, utilization, Inode count and mount point name etc.	lsfs, df, mount, lsvg, lspv etc.	smSiaSystemFileSystem...
System Subsystems		AIX subsystem details of the SIA machine.	Subsystem name, PID, status and subsystem group name	lssrc etc.	smSiaSystemSubSystems...
System Process		Various data about running processes.	Process name, PID, GID, CPU time (user/system), pagefault, priority, status, VM size, memory utilization, CPU utilization and start time etc.	ps etc.	smSiaSystemProcess...
System Users		Information about logged-in users.	User name, login time and PID etc.	who, ps, finger, w etc.	smSiaSystemUsers...
System Utilization	CPU	Information about CPU utilization on the SIA machine.	Percent CPU in user mode, system mode, idle mode and wait mode etc.	iostat, vmstat etc.	smSiaSystemUtilizationCPU...
	Kernel	Information about the kernel status of the SIA machine.	The number of context switch, system call, system read, system write, fork and exec etc.	iostat, vmstat etc.	smSiaSystemUtilizationKernel...
	Iostat	Information about I/O status of the SIA machine	Transfer rate, volume, read volume and write volume etc.	iostat etc.	smSiaSystemUtilizationIostat...
System Miscellaneous		Miscellaneous information about the SIA machine.	System time and unallocated space in vg etc.	date, lsvg etc.	smSiaSystemMiscellaneous...

Appendix B. Systems Monitor Traps

Systems Monitor uses a number of traps to signal important changes in its management region to a top level manager like NetView for AIX. Table 13 shows all the traps and what information they provide. These trap definitions can be accessed online from NetView for AIX via the `Options->Event Configuration->Trap Customization` Menu if you have a NetView for AIX on the desk, otherwise use this page.

Table 13. SM/6000 Traps - Enterprise ID 1.3.6.1.4.1.2.6.12

Event Name	Event	Severity	Status	Category	Source	Log message
SM_SessionDown	Specific 1	Critical	Default	Status Events	A	Session between manager node ″$A″ and agent node ″$1″ is down.
SM_SessionUp	Specific 2	Cleared	Default	Status Events	A	Session between manager node ″$A″ and agent node ″$1″ is up.
SM_StatusDown	Specific 11	Indeterminate	Down	Log only	n	Systems Monitor detected interface $3 down.
SM_StatusUp	Specific 12	Indeterminate	Up	Log only	n	Systems Monitor detected interface $3 up.
SM_FileStringFound	Specific 21	Indeterminate	Default	Status events	A	File monitor: $1 at $4 - File: $3 string: $10 found at line number: $1 omplete line: $13
SM_FileModified	Specific 22	Indeterminate	Default	Status events	A	File monitor: $1 at $4 - File: $3 modified, new size = $5, old size = $6
SM_FileChanged	Specific 23	Indeterminate	Default	Status events	A	File monitor: $1 at $4 - File: $3 status changed: newmode = $5, newuserID = $6, newgroupID = $7,\noldmode = $8, olduserID = $9, oldgroupID = $10
SM_FileNotExists	Specific 24	Indeterminate	Default	Status events	A	File monitor: $1 - File: $3 does not exist.
SM_FileExists	Specific 25	Indeterminate	Default	Status events	A	File monitor: $1 at $4 - File: $3 exists, size = $5, mode = $6, userID = $7, groupID = $8
SM_NewNodes	Specific 31	Indeterminate	Up	Log only	n	Systems Monitor detected ($#-1) new nodes.
SM_OldNodes	Specific 32	Indeterminate	Down	Log only	n	Systems Monitor detected ($#-1) old nodes.
SM_MACMismatch	Specific 33	Indeterminate	Up	Log only	n	Systems Monitor detected MAC address mismatch for $3. New MAC Address=$5 Old MAC Address=$7.

Table 14. SM/6000_Threshold Traps - Enterprise ID 1.3.6.1.4.1.2.6.12.5.1

Event Name	Event	Severity	Status	Category	Source	Log message
SM_ThresholdArm	Specific 1	Indeterminate	Default	Threshold Events	A	Arm threshold ″$3″ trap received from manager $A for node $9.\n Arm threshold was met for variable $11 - value $13 which was matched against ″$4 $5″.\nDescription: $1
SM_ThresholdRearm	Specific 2	Indeterminate	Default	Threshold Events	A	Re-arm threshold ″$3″ trap received from manager $A for node $9.\n Re-arm threshold was met for variable $11 - value $13 which was matched against ″$4 $5″.\nDescription: $1

Appendix C. Important Ports

NOTE

This table does not contain all the well known ports currently assigned. It lists those TCP ports who are bound to distributed network management in any way. Not all of the ports are probably well-known, which means listed in RFC.

Table 15 (Page 1 of 2). TCP/IP Port Related to Network Management

Port name	Port Number	Tranport Protocol	Comment
ftp-data	20	tcp	FTP data connection (initiated by ftp)
ftp	21	tcp	File Transfer Protocol
telnet	23	tcp	Virtual terminal
smtp	25	tcp	Mail Transfer Protocol
time	37	tcp	timserver
time	37	udp	timserver
domain	53	tcp	Domain name server. This port is used for Server synchronization.
domain	53	udp	Domain Name Server Protocol. This port is used for client requests.
bootps	67	udp	Boot Protocol, bootp server port
bootpc	68	udp	Boot Protocol, bootp client port
tftp	69	udp	Trivial File Transfer Protocol. Mainly used by bootp
kerberos5	88	udp	kdc
auth	113	tcp	authentication
snmp	161	udp	snmp request port - handles set/get/get-next requests
snmp-trap	162	tcp	NetView snmp-trap port. Used for reliable Manager-to-manager connections. The trap format is the same as over &udp.
snmp-trap	162	udp	NetView snmp-trap port. The ″standard″ port to receive traps.
nvtrapd-trap	162	udp	NetView trapd monitor trap port
cmot_manager	163	tcp	# NetView CMOT Manager port
cmot_manager	163	udp	# NetView CMOT Manager port
cmot_agent	164	tcp	NetView CMOT Agent port
cmot_agent	164	udp	NetView CMOT Agent port
nvtrapd-trap	165	tcp	NetView trapd monitor trap port
smux	199	tcp	snmpd smux port
login	513	tcp	BSD remote login server port
syslog	514	udp	
route	520	udp	routed/gated routing protocols like rip egp etc...
rmonitor	560	udp	
instsrv	1234	tcp	Network install service
nvtrapd-client	1661	tcp	NetView trapd client application port
nvcorrd	1666	tcp	NetView Correlation daemon port
actionsvr	1670	tcp	NetView Correlation Action daemon port
nvcold	1664	tcp	NetView Collection Facility port
nvsecltd	1667	tcp	NetView Security client daemon

Table 15 (Page 2 of 2). TCP/IP Port Related to Network Management

Port name	Port Number	Tranport Protocol	Comment
C5_server	1668	tcp	NetView C5 Consolidated Console and Threshold Mgmnt
nvlockd	1669	tcp	NetView General Topology Manager lock daemon
nvpagerd	1671	tcp	NetView Pager daemon
nvsecd	1663	tcp	NetView Security daemon port
gtmd	2112	tcp	NetView General Topology Manager port
pmd	2113	tcp	NetView Postmaster daemon port
xmquery	2279	udp	xmquery (???)
writesrv	2401	tcp	temporary port number
xxmd	3113	tcp	NetView General Topology Manager child process port
nvixacm	7111	tcp	AIX NetView Service Point
nvixclb	7112	tcp	AIX NetView Service Point
nvixcr	7113	tcp	AIX NetView Service Point
nvixsp	7115	tcp	AIX NetView Service Point
nvixspc	7116	tcp	AIX NetView Service Point
ovtopmd	8888	tcp	NetView IP Topology daemon
x_st_mgrd	9000	tcp	ibm X terminal
ovwdb	9999	tcp	NetView Object Database daemon

Appendix D. APM Distribution Status Indicators

This table gives an overview over the various distribution states an APM collection can have along with some suggested actions.

Table 16. Status Indicators for Distribution Agent Policy Manager Definitions

Status	**Meaning**
NeverDistributed	An attempt was never made to distribute this definition. Click on *Distribute* to distribute the collection.
Distributed	The definition was successfully distributed to all nodes in the collection.
PartiallyDistributed	An attempt was made to distribute the definition, but one or more nodes could not be modified. It could be that none of the nodes were reached. If the failure is due to a timeout, you might want to let Agent Policy Manager continue to redistribute on its own.
PartiallyDeleted	An attempt was made to delete the definition, but one or more nodes could not be modified. It could be that none of the nodes were reached. Click on **Delete** to retry the deletion. Click on **Undo** to discontinue the operation. You can also do nothing and allow Agent Policy Manager to continue to try to distribute the deletion.
PendingModifyDistribute	A modification was made to the definition, but it has not yet been distributed. Click on **Distribute** to distribute the definition.
PendingDeleteDistribute	The user selected a definition in the APM main dialog and clicked on **Delete**. Select **Distribute** to delete the definition on all nodes and delete the definition. Select **Undo** if you do not want to delete the definition.
DistributeInProgress	Agent Policy Manager was doing a distribution when the C5d daemon went down or a logic error occurred. If you see this error after C5d recycles, allow the C5d daemon to continue to distribute. If you are seeing several items with this status, recycle the daemon.
ModifyDistributeInProgress	Agent Policy Manager was distributing a modified definition when the C5d daemon went down or a logic error occurred. If you see this error after C5d recycles, allow the C5d daemon to continue to distribute. If you are seeing several items with this status, recycle the daemon.
DeleteDistributeInProgress	Agent Policy Manager was distributing a deletion when the C5d daemon went down or a logic error occurred. If you see this error after C5d recycles, allow the C5d daemon to continue to distribute. If you are seeing several items with this status, recycle the daemon.

Appendix E. Sample Shell Scripts

In this appendix you will find listings of three shell scripts that we used in the Chapter 7, “Systems Monitor Examples” on page 161.

These samples, plus all the others found in the book, are available via anonymous FTP:

```
File:    /pub/gg244398.tar.Z

Servers: netview.cary.ibm.com          for users on the Internet
         rsserver.itso.ral.ibm.com     for IBM personnel
```

E.1 smcmd Shell Script

```
#!/usr/bin/ksh
#
# smcmd Shell Script
#
# This takes the name of an SIA Command Table Entry, converts it into a
# dotted decimal MIB ID and then does an SNMP GET for it.
#
# From: IBM Systems Monitor, Anatomy of a Smart Agent, GG24-4398
# By:   Richard Hine and Rob Macgregor
#
# Copyright (C) IBM Corporation, 1994
#

USAGE="usage: smcmd -c community_name -h host_name -n command_name"

typeset -L1 char

function convert_ascii
{

#  Takes a single character and returns its ASCII code

typeset ascii_table

set -A ascii_table ! \" \# \$ \% \& \' \( \) \* \+ \' - . / 0 1 2 3 4 5 6 7 8 9
\: \; \< = \> \? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \\ ] \¬
 _ \ a b c d e f g h i j k l m n o p q r s t u v w x y z { \| } \

c=0
a=$1
while (( c < 94 ))
do
 if [[ $a = "${ascii_table[$c]}" ]]
 then
  ((c = c + 33))
  return $c
 fi
((c = c + 1))
done
}
```

Figure 280 (Part 1 of 3). smcmd Shell Script

```
function getasc
{

# takes a character string and returns it as dotted decimal

typeset -L1 char
line=$1
        length=${#line}
        cnt=0
        while ((cnt < length ))
        do
                char=$line
                if [ "$char" = " " ]
                then
                        echo "\n"
                        exit 0;
                fi
                cnt=$cnt+1
                if [ $cnt -ge 2 ]
                then
                        echo ".\c"
                fi
                line=${line#?}
                convert_ascii $char
                echo "$?\c"
        done
}

c="public"
root=".iso.org.dod.internet.private.enterprises.ibm.ibmProd.systemsMonitor6000."

cmd="smSiaCommand.smSiaCommandTable.smSiaCommandEntry."
name="smSiaCommandName"
rslt="smSiaCommandDisplayStringResult."

while (( $# > 1 ))
do
        case $1 in
                "?") print $USAGE
                        exit
                        ;;
                "-c") if [[ $2 != '' ]]
                                then
                                        c=$2
                                else
                                        c=$(getcommunity $HOST)
                        fi
                        ;;
                "-h") if [[ $2 != '' ]]
                                then
                                        h=$2
                                else
                                        print "No host name specified"
                                        print $USAGE

                                        exit
                        fi
                        ;;
                "-n") if [[ $2 != '' ]]
                                then
                                        n=$2
                                else
                                        print "No command name specified"
                                        print $USAGE
                                        exit
                        fi
                        ;;
                *) print "Invalid switch "$1" has been ingored"
                        ;;
        esac
shift 2
done
```

Figure 280 (Part 2 of 3). smcmd Shell Script

```
if [[ $h = '' ]] then
        h=$(hostname)
fi
if [[ $n = '' ]] then
                print "No command name specified"
                print $USAGE
                exit
fi

ascii=$(getasc $n)

result=`snmpget $h $root$cmd$rslt$ascii`

# Remove the MIB object description from the snmpget response

print "${result#* * * * * * *}"
```

Figure 280 (Part 3 of 3). smcmd Shell Script

E.2 send_fs_trap Shell Script

```
#!/usr/bin/ksh
#
# send_fs_trap Shell Script
#
# This is driven from the MLM Threshold Table. It takes the MIB OID of the
# SIA instance that triggered the threshold and converts it into text.
# Then it sends a trap, containing the instance name, the threshold value
# and the SIA node name.
#
# From: IBM Systems Monitor, Anatomy of a Smart Agent, GG24-4398
# By:   Rob Macgregor
#
# Copyright (C) IBM Corporation, 1994
#

typeset -L1 char

function convert_text
{
# Receives a decimal ASCII code, returns the character

typeset ascii_table
set -A ascii_table ! \" \# \$ \% \& \' \( \) \* \+ \' - . / 0 1 2 3 4 5 6 7 8 9
\: \; \< = \> \? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \\ ] \¬
 _ \ a b c d e f g h i j k l m n o p q r s t u v w x y z { \| } \

d=$1
((d = $d - 33 ))
print ${ascii_table[$d]}
}

mib_oid=$1
utl_value=$2
node_id=$3
spec_id=$4

# Here we strip off the MIB object ID from the total MIB instance,
# thereby leaving the instance in dotted decimal. This version is
# for the File Systems Table. For the Process Table we would replace
# the MIB object with .1.3.6.1.4.1.2.6.12.2.7.2.1.18.

inst=${mib_oid#.1.3.6.1.4.1.2.6.12.2.5.2.1.4.}

# Create the instance name from the ID

IFS="."
for dec_nbr in $inst
do
  psname=$psname$(convert_text $dec_nbr)
done
IFS=" "
trap_cmd=/usr/lpp/smmlm/original/snmptrap
sm_entid=.1.3.6.1.4.1.2.6.12.5.1

# Two variables in the trap - instance name and utilization
# figure that triggered the threshold. The source node is
# set to the SIA node - not the MLM

$trap_cmd localhost public $sm_entid $node_id 6 $spec_id 0 \
          .1.3.6.1.4.1.2.6.12.5.1.1.17 OctetStringASCII  \
          $psname \
          .1.3.6.1.4.1.2.6.12.5.1.1.17 OctetStringASCII  \
          $utl_value
```

Figure 281. send_fs_trap Shell Script

E.3 trap_plus_set Shell Script

```
node=$1
oid=$2
util=$3
cmd_to_run=$4

/u/raleigh/scripts/send_fs_trap $oid $util $node 1008
/u/raleigh/scripts/mlm_setcmd $node $oid $cmd_to_run
```

Figure 282. trap_plus_set Shell Script. This calls send_fs_trap, listed above, and mlm_setcmd, listed below.

```
#!/usr/bin/ksh
#
# mlm_setcmd Shell Script
#
# This is driven from the MLM Threshold Table. It takes the MIB OID of the
# SIA instance that triggered the threshold and converts it into text.
# Then it sends an SNMP SET for the name of a Command Table entry passed
# as an argument, setting the displaystring output to the instance ID it
# just converted. (confused? me too!)
#
# From: IBM Systems Monitor, Anatomy of a Smart Agent, GG24-4398
# By:   Rob Macgregor
#
# Copyright (C) IBM Corporation, 1994
#
#

typeset -L1 char
function convert_ascii
{

# takes a character and returns its ASCII code in decimal

c=0
a=$1
while (( c < 94 ))
do
 if [[ $a = "${ascii_table[$c]}" ]]
 then
  ((c = c + 33))
  return $c
 fi
((c = c + 1))
done
}
```

Figure 283 (Part 1 of 2). mlm_setcmd Shell Script

```
function getasc
{

# Takes a character string and returns it in dotted decimal

typeset -L1 char
line=$1

        length=${#line}
        cnt=0
        while ((cnt < length ))
        do
                char=$line
                if [ "$char" = " " ]
                then
                        echo "\n"
                        exit 0;
                fi
                cnt=$cnt+1
                if [ $cnt -ge 2 ]
                then
                        echo ".\c"
                fi
                line=${line#?}
                convert_ascii $char
                echo "$?\c"
        done
}

# args are: the node on which to SNMP SET, the value to set, as a dotted
# decimal OID, and the Command Table name to SET

node_id=$1
mib_oid=$2
cmd_name=$3

typeset ascii_table

set -A ascii_table ! \" \# \$ \% \& \' \( \) \* \+ \' - . / 0 1 2 3 4 5 6 7 8 9
\: \; \< = \> \? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \\ ] \¬
 _ \ a b c d e f g h i j k l m n o p q r s t u v w x y z { \| } \

# Strip the MIB OID off the value passed as an argument, leaving the
# instance in dotted decimal form.  In the example in the book, this is
# the name of a filesystem.

inst=${mib_oid#.1.3.6.1.4.1.2.6.12.2.5.2.1.4.}

# Convert the instance OID into text

IFS="."
for dec_nbr in $inst
do
  (( dec_nbr = $dec_nbr - 33 ))
  fsname=$fsname${ascii_table[$dec_nbr]}
done
IFS=" "

cmd_inst=$(getasc $cmd_name)
sm_cmdid=1.3.6.1.4.1.2.6.12.4.1.1.13.$cmd_inst

set_cmd="/usr/sbin/snmpinfo -m set -c ITSC"

$set_cmd -h $node_id $sm_cmdid=$fsname
```

Figure 283 (Part 2 of 2). mlm_setcmd Shell Script

Appendix F. IP Trace for MLM Failure Scenario

Below is a formatted IP trace showing the activity when an MLM node (midman, address 9.67.32.123) is detected to have failed by NetView for AIX (on rs60002.itso.ral.ibm.com, 9.24.104.28). We have inserted comments into the trace to explain what is happening at each point.

```
=====( packet transmitted on interface tr0 )=====Wed Nov 16 10:01:20 1994
IP header breakdown:
        < SRC =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
        < DST =     9.67.32.123 >  (midman)
UDP header breakdown:
        <source port=4287, <destination port=161(snmp) >

00000000     302b0201 00040449 545343a0 20020300     |0+.....ITSC. ...|
00000010     bdf00201 00020100 30133011 060d2b06     |........0.0...+.|
00000020     01040102 060c0101 01030005 00           |.............   |

=====( packet received on interface tr0 )=====Wed Nov 16 10:01:20 1994
IP header breakdown:
        < SRC =     9.67.32.123 >  (midman)
        < DST =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
UDP header breakdown:
        <source port=161(snmp), <destination port=4287 >
00000000     303a0201 00040449 545343a2 2f020300     |0:.....ITSC./...|
00000010     bdf00201 00020100 30223020 060d2b06     |........0" ..+.|
00000020     01040102 060c0101 01030004 0f30322e     |.............02.|
00000030     30312e30 3030302e 30303030              |01.0000.0000    |

NOTES >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
This exchange shows the polling that NetView for AIX does of the
MLM.  It is an SNMP GET for MIB object .1.3.6.1.4.1.2.6.12.1.1.1.3,
which is the version number of the MLM.  The response is
"02.01.0000.0000".
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<NOTES

=====( packet transmitted on interface tr0 )=====Wed Nov 16 10:01:50 1994
IP header breakdown:
        < SRC =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
        < DST =     9.67.32.123 >  (midman)
UDP header breakdown:
        <source port=4287, <destination port=161(snmp) >
00000000     302b0201 00040449 545343a0 20020300     |0+.....ITSC. ...|
00000010     bdf10201 00020100 30133011 060d2b06     |........0.0...+.|
00000020     01040102 060c0101 01030005 00           |.............   |

=====( packet received on interface tr0 )=====Wed Nov 16 10:01:50 1994
IP header breakdown:
        < SRC =     9.67.32.123 >  (midman)
        < DST =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
UDP header breakdown:
        <source port=161(snmp), <destination port=4287 >
00000000     302b0201 00040449 545343a2 20020300     |0+.....ITSC. ...|
00000010     bdf10201 02020101 30133011 060d2b06     |........0.0...+.|
00000020     01040102 060c0101 01030005 00           |.............   |

NOTES >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
30 seconds (the polling interval) later, the same poll returns a
"not found" response, because midmand has shut down.  This will cause
NetView for AIX to take over polling.
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<NOTES
```

Figure 284 (Part 1 of 4). Annotated Trace of Polling Activity

```
=====( packet transmitted on interface tr0 )=====Wed Nov 16 10:01:50 1994
IP header breakdown:
        < SRC =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
        < DST =     9.67.32.123 >  (midman)
ICMP header breakdown:
        icmp_type=8 (ECHO_REQUEST)  icmp_id=43026  icmp_seq=43396

=====( packet transmitted on interface tr0 )=====Wed Nov 16 10:01:50 1994
IP header breakdown:
        < SRC =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
        < DST =      9.67.32.84 >  (rs60005-e.itsolab.ral.ibm.com)
ICMP header breakdown:
        icmp_type=8 (ECHO_REQUEST)  icmp_id=43026  icmp_seq=43397

=====( packet transmitted on interface tr0 )=====Wed Nov 16 10:01:50 1994
IP header breakdown:
        < SRC =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
        < DST =      9.67.32.43 >
ICMP header breakdown:
        icmp_type=8 (ECHO_REQUEST)  icmp_id=43026  icmp_seq=43398
=====( packet transmitted on interface tr0 )=====Wed Nov 16 10:01:50 1994
IP header breakdown:
        < SRC =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
        < DST =      9.67.32.13 >
ICMP header breakdown:
        icmp_type=8 (ECHO_REQUEST)  icmp_id=43026  icmp_seq=43399

=====( packet transmitted on interface tr0 )=====Wed Nov 16 10:01:50 1994
IP header breakdown:
        < SRC =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
        < DST =      9.67.32.12 >  (hp.itsolab.ral.ibm.com)
ICMP header breakdown:
        icmp_type=8 (ECHO_REQUEST)  icmp_id=43026  icmp_seq=43400

NOTES >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
As the MLM is down, NetView for AIX has to take over status polling for
the 9.67.32 network.  Here we see it starting to do ICMP echo polls to
all the nodes that midman was polling, including midman itself.
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< NOTES

=====( packet received on interface tr0 )=====Wed Nov 16 10:01:50 1994
IP header breakdown:
        < SRC =      9.67.32.84 >  (rs60005-e.itsolab.ral.ibm.com)
        < DST =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
ICMP header breakdown:
        icmp_type=0 (ECHO_REPLY)  icmp_id=43026  icmp_seq=43397

=====( packet received on interface tr0 )=====Wed Nov 16 10:01:50 1994
IP header breakdown:
        < SRC =     9.67.32.123 >  (midman)
        < DST =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
ICMP header breakdown:
        icmp_type=0 (ECHO_REPLY)  icmp_id=43026  icmp_seq=43396

=====( packet received on interface tr0 )=====Wed Nov 16 10:01:50 1994
IP header breakdown:
        < SRC =      9.67.32.12 >  (hp.itsolab.ral.ibm.com)
        < DST =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
ICMP header breakdown:
        icmp_type=0 (ECHO_REPLY)  icmp_id=43026  icmp_seq=43400

=====( packet received on interface tr0 )=====Wed Nov 16 10:01:50 1994
IP header breakdown:
        < SRC =      9.67.32.13 >
        < DST =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
ICMP header breakdown:
        icmp_type=0 (ECHO_REPLY)  icmp_id=43026  icmp_seq=43399

=====( packet transmitted on interface tr0 )=====Wed Nov 16 10:01:52 1994
IP header breakdown:
        < SRC =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
        < DST =      9.67.32.43 >
ICMP header breakdown:
        icmp_type=8 (ECHO_REQUEST)  icmp_id=43026  icmp_seq=43398
```

Figure 284 (Part 2 of 4). Annotated Trace of Polling Activity

```
NOTES >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
NetView for AIX has received some responses to its ICMP echo polls,
but did not receive a response from 9.67.32.43, so it retries.  This
will eventually time out, and the node will be marked as down.
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< NOTES

-------------------------------------------------------------------------

At this point in the trace the ECHO_REQUEST, ECHO_RESPONSE sequences
continue for each of the nodes in the network.  We have removed them
for clarity.

-------------------------------------------------------------------------

=====( packet transmitted on interface tr0 )=====Wed Nov 16 10:02:20 1994
IP header breakdown:
        < SRC =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
        < DST =     9.67.32.123 >  (midman)
UDP header breakdown:
        <source port=4287, <destination port=161(snmp) >
00000000     302b0201 00040449 545343a0 20020300     |0+.....ITSC. ...|
00000010     bdf20201 00020100 30133011 060d2b06     |........0.0...+.|
00000020     01040102 060c0101 01030005 00           |.............   |

=====( packet received on interface tr0 )=====Wed Nov 16 10:02:20 1994
IP header breakdown:
        < SRC =     9.67.32.123 >  (midman)
        < DST =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
UDP header breakdown:
        <source port=161(snmp), <destination port=4287 >
00000000     302b0201 00040449 545343a2 20020300     |0+.....ITSC. ...|
00000010     bdf20201 02020101 30133011 060d2b06     |........0.0...+.|
00000020     01040102 060c0101 01030005 00           |.............   |

NOTES >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
As well as doing the ICMP polling, NetView also carries on checking
to see if the MLM is up, using the SNMP GET request.  Here we see that
it is still down, so the ICMP echo requests continue...
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< NOTES

=====( packet transmitted on interface tr0 )=====Wed Nov 16 10:02:26 1994
IP header breakdown:
        < SRC =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
        < DST =      9.67.32.43 >
ICMP header breakdown:
        icmp_type=8 (ECHO_REQUEST)  icmp_id=43026  icmp_seq=43411
```

Figure 284 (Part 3 of 4). Annotated Trace of Polling Activity

```
-------------------------------------------------------------------------

At this point in the trace the ECHO_REQUEST, ECHO_RESPONSE sequences
continue for each of the nodes in the network.  We have removed them
for clarity.

-------------------------------------------------------------------------

=====( packet transmitted on interface tr0 )=====Wed Nov 16 10:02:50 1994
IP header breakdown:
        < SRC =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
        < DST =     9.67.32.123 >  (midman)
UDP header breakdown:
        <source port=4287, <destination port=161(snmp) >
        · udp length = 53 | udp checksum = 56d “
00000000     302b0201 00040449 545343a0 20020300     |0+.....ITSC. ...|
00000010     be000201 00020100 30133011 060d2b06     |........0.0...+.|
00000020     01040102 060c0101 01030005 00           |............    |

=====( packet received on interface tr0 )=====Wed Nov 16 10:02:50 1994
IP header breakdown:
        < SRC =     9.67.32.123 >  (midman)
        < DST =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
UDP header breakdown:
        <source port=161(snmp), <destination port=4287 >
00000000     303a0201 00040449 545343a2 2f020300     |0:.....ITSC./...|
00000010     be000201 00020100 30223020 060d2b06     |........0”0 ..+.|
00000020     01040102 060c0101 01030004 0f30322e     |.............02.|
00000030     30312e30 3030302e 30303030              |01.0000.0000    |

NOTES >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Now the MLM daemon has restarted on midman, so we get a valid response
(”02.01.0000.0000”) to the SNMP GET for the MLM version.  NetView for
AIX on rs60002 will immediately stop polling the nodes in the 9.67.32
network for status (although some replies may still be pending - below).
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< NOTES

=====( packet received on interface tr0 )=====Wed Nov 16 10:02:50 1994
IP header breakdown:
        < SRC =      9.67.32.13 >
        < DST =     9.24.104.28 >  (rs60002.itso.ral.ibm.com)
ICMP header breakdown:
        icmp_type=0 (ECHO_REPLY)  icmp_id=43026  icmp_seq=43418
```

Figure 284 (Part 4 of 4). Annotated Trace of Polling Activity

Appendix G. SNMP-Related Requests for Comment

RFC texts may be obtained by sending an ERFC-INFO@ISI.EDU., containing the following lines:

```
Retrieve: RFC
Doc-ID:   RFCnnnn
```

The RFC ID is padded with leading zeroes to make four digits.

For IBM personnel there is a TOOLS disk containing RFC text. Enter the following from VM:

```
TOOLS SENDTO ALMVMA ARCNET RFC GET RFCnnnn TXT
```

Table 17 lists the RFC IDs for the basic SNMP standards and for the RMON and Host Resources MIBs.

Table 17. SNMP-Related Requests for Comment

RFC1155	Structure and Identification of Management Information for TCP/IP based internets.
RFC1156	Management Information Base Network Management of TCP/IP based internets
RFC1157	A Simple Network Managment Protocol
RFC1212	Concise MIB Definitions
RFC1213	Management Information Base for Network Management of TCP/IP based internets: MIB-II
RFC1215	A Convention for Defining Traps for use with the SNMP
RFC1271	Remote Network Monitoring Management Information Base (RMON MIB)
RFC1513	Token-Ring Extensions to the Remote Network Monitoring MIB
RFC1514	Host Resources MIB

Index

Special Characters

A

C

D

E

F

R

S

T

U

W

ITSO Technical Bulletin Evaluation RED000

International Technical Support Organization
IBM Systems Monitor
Anatomy of a Smart Agent
February 1996

Publication No. SG24-4398-01

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction	____		
Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) If you are an employee of IBM or its subsidiaries:

Do you provide billable services for 20% or more of your time? Yes____ No____

Are you in a Services Organization? Yes____ No____

b) Are you working in the USA? Yes____ No____

c) Was the Bulletin published in time for your needs? Yes____ No____

d) Did this Bulletin meet your needs? Yes____ No____

If no, please explain:

_____

_____

What other topics would you like to see in this Bulletin?

_____

_____

What other Technical Bulletins would you like to see published?

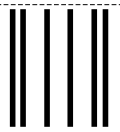
_____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

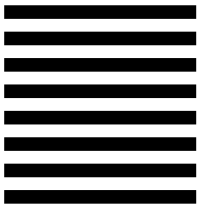
Name	Address
Company or Organization	
Phone No.	

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Department HZ8, Building 678
P.O. BOX 12195
RESEARCH TRIANGLE PARK NC
USA 27709-2195

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.