Managing One or More AIX Systems - Overview

Document Number GG24-4160-01

December 1994

International Technical Support Organization
Austin Center

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 (December 1994)

This edition applies to the RISC System/6000 for use with the AIX Operating System Version 3.2 and Version 4.1.

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

An ITSC 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 Center
Dept. 948S, Building 821 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1993,1994. All rights reserved.**
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Abstract

This document is intended to give guidance to customers, IBM system engineers and third-party professionals concerned with installation and administration of a RISC System/6000 platform.

It covers system management concepts strategies, and gives an overview of products. It illustrates the IBM offering to manage customer installations from single to multiple RISC System/6000′s and the AIX Version 3.2.5 and Version 4.1 Operating System.

AX (257 pages)

© Copyright IBM Corp. 1993,1994

Contents

© Copyright IBM Corp. 1993,1994

Figures

© Copyright IBM Corp. 1993,1994

Tables

© Copyright IBM Corp. 1993,1994

Special Notices

This publication is intended to help customers, systems engineers and RISC System/6000 administrators to understand the strategies and products to manage a full RISC System/6000 installation. The information in this publication is not intended as the specification of any programming interfaces that are provided by AIX Version 3.2 and Version 4.1 on RISC System/6000′s. See the PUBLICATIONS section of the IBM Programming Announcement for RISC System/6000 and AIX products names 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 Commercial Relations, 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 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.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

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, which are denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ADSTAR	AIX
AIX/ESA	AIX/6000
AIXwindows	CICS/6000

© Copyright IBM Corp. 1993,1994

DB2/6000	IBM
InfoExplorer	LoadLeveler
NetView	NetView DM
OS/2	PowerOpen
RISC System/6000	Scalable POWERparallel Systems

The following terms, which are denoted by a double asterisk (**) in this publication, are trademarks of other companies:

Andrew File System	Transarc Corp
Apollo, HP, Hewlett Packard, HP-UX	Hewlett-Packard Company
Apple, MacIntosh	Apple Computer, Inc
AT&T	AT&T, Inc.
BULL, BOS/X	Machines BULL
CATIA	Dassault Systemes
DEC, Alpha	Digital Equipment Corporation
Framemaker	FRAME Technology
IRIS	Silicon Graphics
Palladium, Athena, Kerberos,	Massachusetts Institute Technology
X-Windows	Massachusetts Institute Technology
Legato, Networker	Legato Systems, Inc.
Microsoft, Windows	Microsoft Corporation
Yellow Pages	British Telecom
Oracle	Oracle Inc.
Motif, DCE, OSF/1, DME, ANDF, OSF	Open Software Fundation
SCO	Santa Cruz Operation
SunOS, SPARCstation, Network File System, NFS	Sun Microsystems, Inc.
SUN, NIS, Solaris, Sun Microsystems, SUNSoft	Sun Microsystems, Inc.
1-2-3, Lotus	Lotus Development Corporation.
NetWare, Novell	Novell Inc.
UNIX	X/OPEN Company, Ltd.

Others trademarks are trademarks of their respective companies.

Preface

This document is intended to assist the reader in managing their RISC System/6000 more effectively. As well as a general discussion of techniques and considerations for managing the system installation, it contains descriptions and practical examples of the usage of this IBM offering for improving system management.

This document is intended to provide guidance to IBM SE′s, customer personnel, system administrators, consultants and others in planning for and achieving better administration and organization for their AIX systems.

How This Document is Organized

The document is organized as follows:

- Chapter 1, “Introduction to System Management”
- Chapter 2, “Building and Maintaining the AIX You Need”

  This chapter provides a description on how AIX Version 3 is organized with the definition of some of the terminology used in discussing system management. It gives an introduction to system administration concepts. It assists in defining AIX needs and choosing the proper maintenance level.
- Chapter 3, “Application Installation”

  This chapter presents various scenarios to organize and install applications. It contains a description of AIX NetView DM/6000 as an example of application distribution.
- Chapter 4, “Monitoring Applications”

  This chapter is a presentation of distributed organization and the ability to check one or several applications from one machine using Systems Monitor for AIX.
- Chapter 5, “Application Data Backup Strategies”

  One of the most important responsibilities of a system administrator is to organize the security of the data. This chapter is a presentation of the backup strategies followed by an example of backup with Legato NetWorker.
- Chapter 6, “Distributed Batch Job Management”

  This chapter is a description of job management and a presentation of IBM Loadleveler, which is a job scheduler.
- Chapter 7, “Configuration Management”

  This chapter approaches the configuration of RISC System/6000′s using the front end user products provided by AIX.
- Chapter 8, “Performance Measurement”

  This chapter considers performances problems and a description of the Performance Toolbox for AIX which provides a graphical view of system performance.
- Chapter 9, “Distributed Security”

© Copyright IBM Corp. 1993,1994

This chapter is a general presentation on distributed security and provides guidance on choosing the best tools to secure the network.

- Chapter 10, “Wide Area Network Security”

  This chapter is a general presentation of the security rules when exchanging electronic information through Internet. It gives an overview of the firewall concept and makes a presentation of the different strategies.

- Chapter 11, “Storage Management”

  This chapter is an overview of system storage administration, and shows where the IBM product offering is positioned.

- Chapter 12, “Centralized Problem Management”

  This chapter shows you a way to monitor and track problems of clients, systems and users.

- Appendix B, “CookBook on NetView for AIX and other products”

  This appendix develops a cookbook on the NetView for AIX V3, Systems Monitor for AIX V2 and Trouble Ticket/6000 V3 products.

Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document.

- *AIX FSF/6000 Installation Planning User's Guide*, SC23-2587
- *Performance Toolbox 1.2 and 2.1 for AIX Guide and Reference*, SC23-2625
- *AIX Commands Reference Volume 1*, GC23-2376
- *AIX Commands Reference Volume 2*, GC23-2366
- *AIX Commands Reference Volume 3*, GC23-2367
- *AIX System Management Guide*, SC23-2457
- *NetView for AIX Concepts*, GC31-6234
- *NetView for AIX Installation and Configuration*, SC31-6237
- *NetView for AIX User's Guide for Beginners*, SC31-6232
- *NetView for AIX Administrator's Guide*, SC31-7192
- *NetView for AIX Administrator's Reference*, SC31-8104
- *NetView for AIX Programmer's Guide*, SC31-6238
- *NetView for AIX Programmer's Reference*, SC31-6239
- *NetView for AIX Database Guide*, SC31-7190
- *Examples of Using AIX NetView/6000 APIs*, GG24-4059
- *Systems Monitor for AIX User's Guide*, SC31-7150
- *AIX Trouble Ticket/6000 User's Guide*, SC31-7160
- *IBM LoadLeveler User's Guide*, SH26-7226
- *IBM LoadLeveler Administration Guide*, SH26-7220
- *NetView Distribution Manager/6000 Installation and Customization Guide*, SH19-5002
- *NetView Distribution Manager/6000 User's Guide*, SH19-5003

- *NetView Distribution Management Agent/6000 User's Guide*, SH19-4071

International Technical Support Center Publications

A complete list of International Technical Support Center publications, with a brief description of each, may be found in:

Bibliography of International Technical Support Centers Technical Bulletins, GG24-3070.

Acknowledgments

The advisor for the second edition of this publication was:

Daniel François
International Technical Support Organization, Austin Center

The authors of this document are:

Andréa Li-Saï
IBM France

François Riche
IBM France

This publication is the result of a residency conducted at the International Technical Support Organization, Austin Center.

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

Dave Shogren
International Technical Support Organization, Raleigh Center

Barry Nusbaum
International Technical Support Organization, Raleigh Center

Rob Macgregor
International Technical Support Organization, Raleigh Center

The author for the first edition of this publication was:

Daniel François
International Technical Support Organization, Austin Center

The document was the product of contributions from several authors:

Yannick Bollati
IBM France

Roland Caris
IBM France

François Riche
IBM France

Jakob Streif
IBM Austria

Chapter 1. Introduction to System Management

The purchase of a new computer system requires a significant investment of financial and human resources. As with any other investment, this requires care and attention to get the maximum benefit. For this reason, a clear system management strategy is required.

Below is a list of common system management tasks. Because systems differ, and system management requirements differ, a unique system management strategy must be set up for each environment. Tasks which are important in one environment may not necessarily be important in another.

Software maintenance If new software is required, it has to be installed and tested. After successful installation and testing, user access has to be enabled. For new machines, it must also be decided, which software should be installed.

User management New users have to be created with the appropriate access rights. For security reasons, unused user id′s must be disabled or removed. In addition, current users often require changes to their environment.

Backup/recovery One of the most important tasks in any system is to make sure that all important data can be restored in case of a disaster. There should be recovery procedures in place for both the worst case, and also for simple cases, such as restoring a file which has been accidentally deleted.

Storage management In environments with a large amount of data, there will be a need for storage management products. In general, this involves migrating seldom used files onto less expensive media, such as tapes or optical media, while still keeping them accessible to users.

Network management This includes all the tasks which are necessary to enable the communication between all machines within an enterprise. These are mainly routing, nameserving and mailing.

Problem management Any kind of problem must be handled by the system administration. Problems can be system halts, application problems, network breakdowns, simple user questions, and so on.

Security Security is extremely important to the enterprise; unfortunately, it is one area which is often neglected. Security is especially important when there are connections to networks outside the enterprise and unauthorized access is to be prevented.

Accounting With accounting, one can control the usage of system resources, such as CPU time, disk space, and so on.

Configuration This is basically defining, configuring and updating system resources such as printers, network parameters, hardware, system parameters and so on.

© Copyright IBM Corp. 1993,1994

Change management	This involves planning, distributing and installing changes to system software, applications and data.
Other activities	Specific activities, such as providing documentation. Small programs for automation are often need to be written.
Planning	Since the system managers have the best knowledge about the environment and system usage, they must be involved to get a proper investment planning.

AIX provides many standard tools to manage your environment, for instance Visual System Management. Many additional applications also exist for this purpose.

This book provides an overview about system management tasks and the tools available.

Chapter 2. Building and Maintaining the AIX You Need

AIX* Version 3 is a set of software products designed to meet a wide variety of requirements. For a given business environment, you will have to *build* from the IBM* installation tapes the AIX system, which will include the exact list of options and fixes you need. You will also have to manage the software maintenance and changes. This chapter describes methods and tools you can use to manage your AIX system on your RISC System/6000s*.

2.1 Choosing the Options You Need

Installing AIX means installing many products, called *LPPs* (*Licensed Program Products*). This includes the *Base Operating System* called *bos.obj*, and many other products you need. Don't expect a file named bos.obj on your system, it isn't actually an object file, but only the name AIX is used to define the product. Each of these LPPs is divided into one or several options, called *OPPs* (*Optional Program Products*). These may or not may be selected at installation time. You will have to define the exact list of OPPs you need, within the different LPPs you bought. Figure 1 on page 4 describes the relationships between the fixes, the options, the LPPs and the whole system.

2.2 Choosing the Proper Level of Maintenance (Fix or Enhancement)

AIX Version 3 has a new built-in maintenance mechanism dealing with entities named *selective fixes*. These selectives fixes contain all the data needed by the installation program to properly apply them on the corresponding LPP. This mechanism allows use of the maintenance procedure on all the products of the system.

This internal maintenance mechanism is also used to add new features or hardware support to the system. These additions are called selective enhancements.

To ease the determination of a product's maintenance level, IBM provides cumulative packages called maintenance levels. They contain the update set of all the selective fixes and selective enhancements for a given product.

© Copyright IBM Corp. 1993,1994

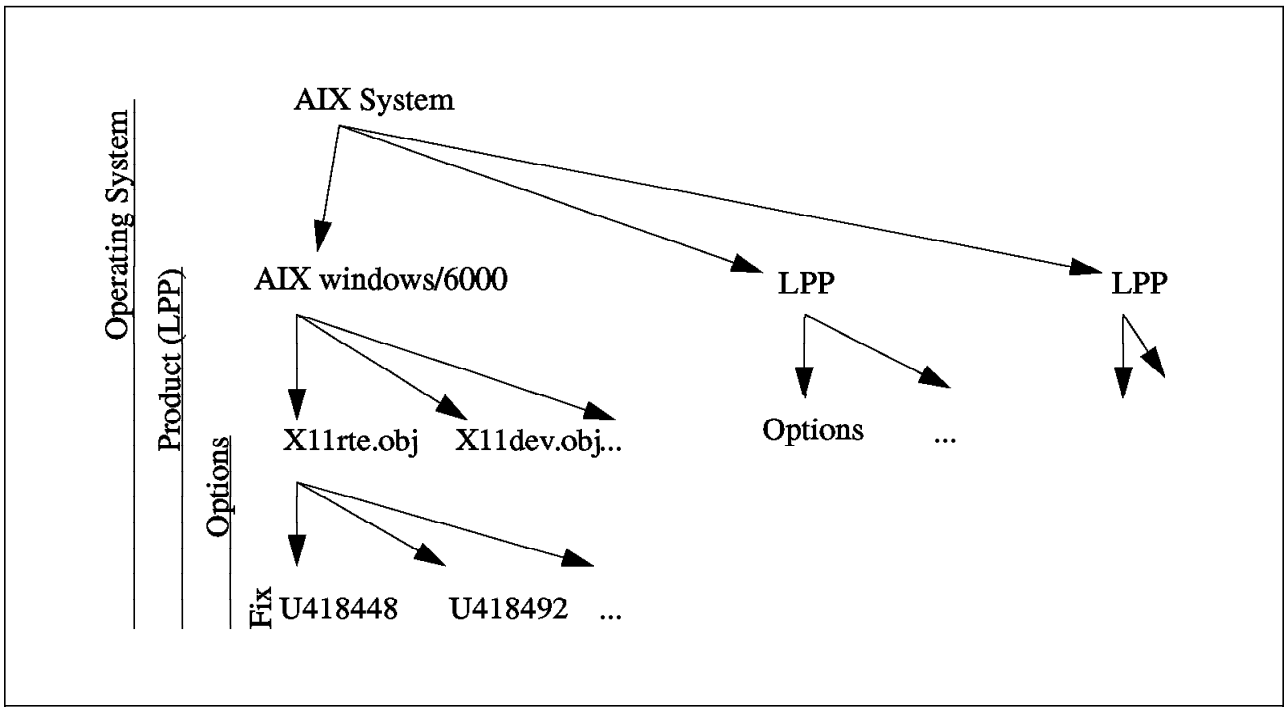

Figure 1. Architecture of LPPs, Options and Fixes

2.2.1 Product, Option or Fix Status

When an option or a fix is installed on a RISC System/6000, this operation is recorded in an object oriented database named ODM (Object Database Manager). See Chapter 7, “Configuration Management” on page 79.

When you install a product, option or a fix, you can choose to apply or commit the installation. Apply means the installation mechanism will preserve the old programs that would have been replaced before installing the new code. Then you can later reject this product, option or fix. Reject means the new programs will be erased and replaced by the old versions. Commit means install a product, option or fix, without saving the old version. You can’t reject a product, option or fix if you have committed it.

2.2.2 How to View the Maintenance Level

You can use the `lslpp` command to get formatted output from the ODM database about the products installation history.

For instance in Figure 2 on page 5, you have a list of the installed options for the AIXwindows/6000* product on an AIX 3.2.4 machine. The description field gives you the option name (X11dev.obj 1.2 for instance), the maintenance level name (3240 X11dev X11R4), the state of the option (C for committed) and the number of the last fix installed or this option.

```
# lslpp -L 'X11*'

 Description                                             State     Fix Id
 ------------------------------------------------------  ------    ------------

 X11dev.obj 1.2.
   3240 X11dev X11R4 Maintenance Level                       C    U491016

 X11dev.src 1.2.
   3240 X11dev X11R4 Maintenance Level                       C    U491016

 X11fnt.coreX.fnt 1.2.
   3240 X11fnt X11R4 Maintenance Level                       C    U491017

 X11fnt.ibm850.pc.fnt 1.2.
   3240 X11fnt X11R4 Maintenance Level                       C    U491017

 X11fnt.iso88591.aix.fnt 1.2.
   3240 X11fnt X11R4 Maintenance Level                       C    U491017

 X11fnt.iso88592.fnt 1.2.
   3240 X11fnt X11R4 Maintenance Level                       C    U491017

 X11fnt.iso88593.fnt 1.2.
   3240 X11fnt X11R4 Maintenance Level                       C    U491017

 X11fnt.iso88594.fnt 1.2.
   3240 X11fnt X11R4 Maintenance Level                       C    U491017

 X11fnt.iso88595.fnt 1.2.
   3240 X11fnt X11R4 Maintenance Level                       C    U491017

 X11mEn_US.msg 1.2.
   3240 X11mEn_US X11R4 Maintenance Level                    C    U491020

 X11rte.ext.obj 1.2.
   3240 X11rte X11R4 Maintenance Level                       C    U491015

 X11rte.obj 1.2.
   3240 X11rte X11R4 Maintenance Level                       C    U491015


State codes:
 A -- Applied.
 B -- Broken.
 C -- Committed.
 N -- Not Installed, but was installed/seen on some media.
 - -- Superseded, not Applied.
? -- Inconsistent State...Run lppchk -v.
```

Figure 2. List of Options for AIXwindows

Using the lslpp command, you can get more details about a given option. For instance in Figure 3 on page 6, you can see the components of the option X11dev.obj 1.2. and all the fixes on the components.

```
# lslpp -L -a X11dev.obj

 Description                                          State     Fix Id
 ---------------------------------------------------  ------    ------------

 X11dev.obj 1.2.
   3240 X11dev X11R4 Maintenance Level                   C     U491016
     Motif 1.1.4 Resource Manager                        C     U418465
                                                               Supersedes:
     Update Package                                      -        U412518
     X11-R4 Xm Widgets                                   C     U418466
                                                               Supersedes:
     Update Package                                      -        U408445
     X11-R4 UIL                                          C     U418467
                                                               Supersedes:
     Update Package                                      -        U415619
     Update Package                                      -        U406719
     X11-R4 Include Files                                C     U418468
                                                               Supersedes:
     Update Package                                      -        U416139
     Update Package                                      -        U410758
     Update Package                                      -        U411423
     X11-R4 Display Postscript                           C     U419140


State codes:
 A -- Applied.
 B -- Broken.
 C -- Committed.
 N -- Not Installed, but was installed/seen on some media.
 - -- Superseded, not Applied.
? -- Inconsistent State...Run lppchk -v.
```

Figure 3. Detail Information about the Option X11dev.obj

Using the lslpp command, you can list detailed information for an individual fix from the ODM database. For instance in Figure 4 on page 7, you have the exact explanation of the content of the fix. This text references the APAR numbers that identifies individual bugs in the IBM* RETAIN international database.

```
# lslpp -a -A U418468

Name                  Fix Id   State         Fix Information
  --------------------  -------  ------------  --------------------------------
Path: /usr/lib/objrepos
  X11dev.obj 01.02.00.00
                        U418468  COMMITTED
   ----------------------------------------------------------------------

X11-R4 Include Files
IX32166 X11R4: Xlib.h defines index function without function prototypes

 this is a header file problem.  Xlibos.h defines the index
 function without function prototypes.  the #ifndef sgi line
 defines the index function without its prototypes.

IX31520 Graphic application performance affected by cursor motion.

 The program builds two structures, one containing text and the other
 containing lines and arcs.  It then performs
two update workstation
 operations. The draw time associated with the second of these updates,
 was 30 seconds on the previous release but is around 70 seconds now
 unless you move the cursor out of and back into the window at the
 beginning of the draw, in which case it drops to just under 20 seconds.

IX27096 GL GETVALUATOR SUBROUTINE PERFORMANCE SLOW

 GL getvaluator() function performance is slow.

IX36033 X11-R4 Include Files Subsystem

 Rollup of all changes to the X11-R4 Include Files Subsystem.

   ----------------------------------------------------------------------
```

Figure 4. Detailed Information about One Fix of the Option X11dev.obj

2.2.3 Getting Preventive Maintenance Packages

When you build your AIX system, you should also install the available fixes. You can get those fixes two different ways:

- From time to time, IBM delivers a *refresh* tape containing the existing fixes tested together plus some enhancements, that are the preventive maintenance packages.
- If you have particular problems, you can call IBM Services to get intermediate fixes or preventive maintenance packages.

2.3 Maintenance Strategies

We have tried to give sample strategies for managing AIX versions and fixes for a system administrator. The number of machines involved and the type of organization of your company have a major impact on what those strategies can be. So we summarize them with three scenarios. Each scenario tries to represent a typical company organization with AIX machines.

2.3.1 Single Machine Scenario

Scenario abstract: A customer with only one RISC System/6000 machine. This machine is used to run business applications. The typical configuration is a RISC System/6000 with asynchronous terminals.

2.3.1.1 Maintenance Life Cycle

During its life, your system will go through many maintenance steps. This is the maintenance life cycle. The number and complexity of those steps depend on which scenario you are in. In Figure 5, you can see the simplest maintenance life cycle you usually have with a *single machine* scenario.

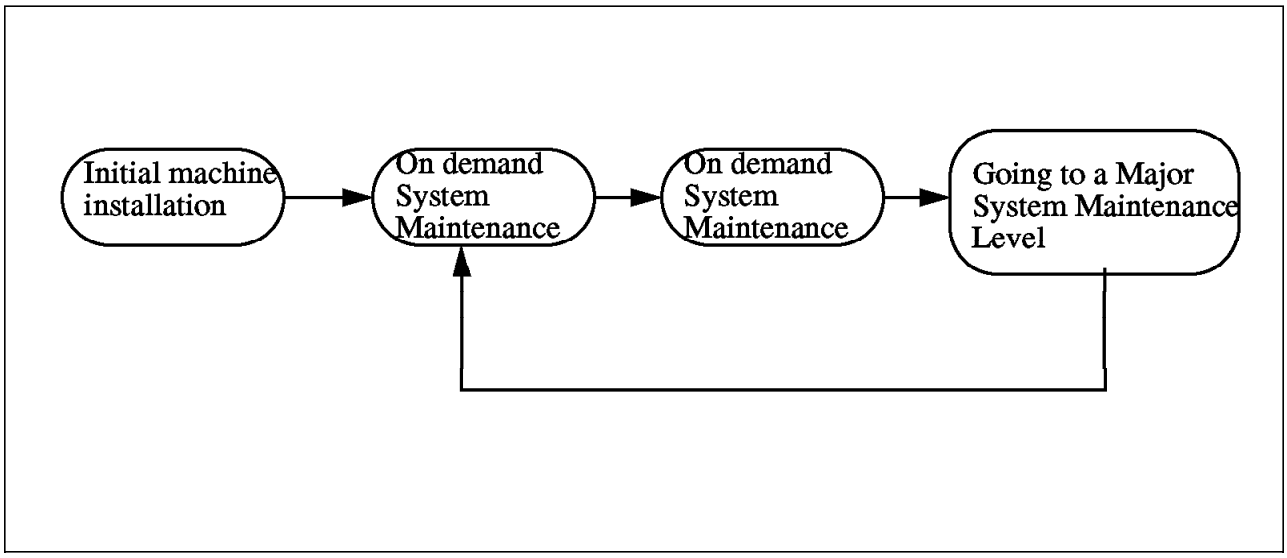

Figure 5. Simple Maintenance Life Cycle

2.3.1.2 Initial Machine Installation

This is the first step in your maintenance life cycle. You have just received the machine from IBM and you want to set it up with one or more applications. There are the typical tasks you should do (Figure 6 on page 9):

1. Install the currently available AIX base operating system from an IBM official tape.
2. Choose the LPPs you want.
3. Install only the needed options of the LPPs (for instance do not install X11dev option if you do not need it) from the IBM official tapes.
4. Install your applications.
5. Test your applications on this platform.
6. If you have problems in step 5, get an available fix, if it's a known problem, from IBM Services. Install the fix and go back to step 4.
7. Do a system backup of your system (`smitty mksysb`). Do it twice on two different tapes, never trust just one tape.
8. Do an applications code backup (do it twice on two tapes).
9. Do an initial application data backup (do it twice too).
10. Put clear labels on the tapes. Give a level name to your backups, for instance my system - version 1.

11. Restore your machine from those three backups. Check if everything has been correctly restored. If not, go back to step 1.
12. Store the tapes in two different safe places.

Then your system should be ready for production.

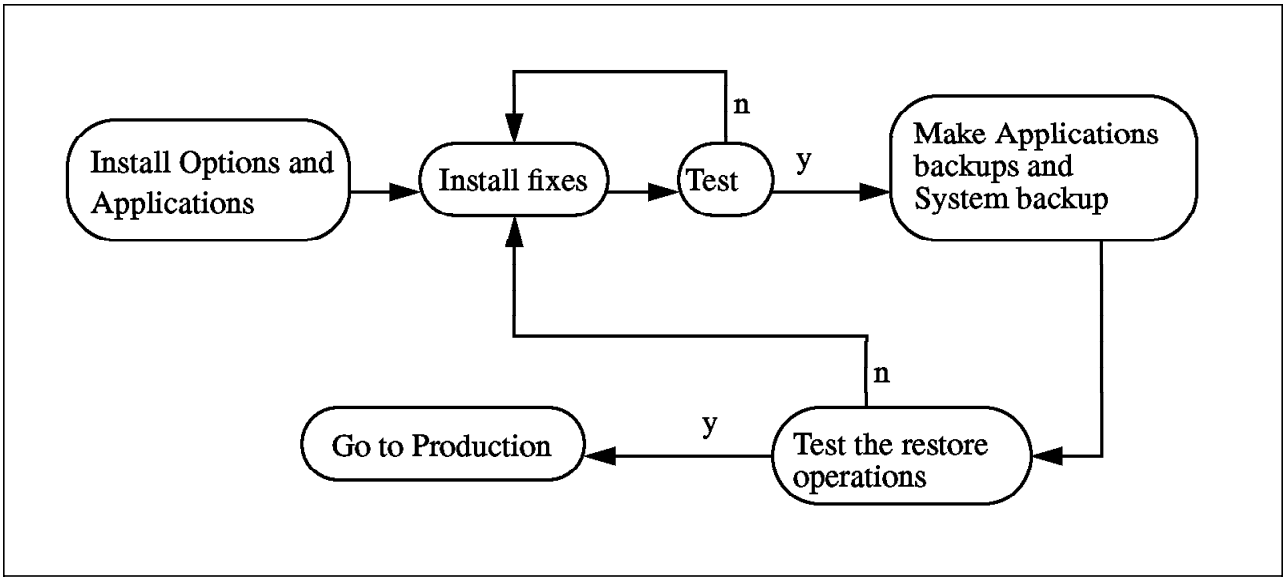

Figure 6. Initial Machine Installation Synopsis

2.3.1.3 On Demand System Maintenance

After a while you may have encountered a small problem with AIX or an *LPP*. If this is a known problem, you may install a fix delivered by IBM Services. This *on demand system maintenance* task (Figure 5 on page 8) can occur several times, depending of the number and complexity of the use of your AIX system. To do on demand system maintenance, do the following:

1. Be sure that the previous backups (`mksysb`, application code, application data) are good and available. Plan the operation at a time where you will be able to *restore the entire machine* from those backups if something goes wrong (do not plan to install a fix at 8am if your company opens at 9am).
2. Install the fix. If you have enough disk space, use the apply option from the SMIT installation menu.
3. Check if the problem you encountered is fixed. If not, you should probably *reject* the fix.
4. Make a new system backup.
5. Put your own clear label on the IBM fix tapes, for instance my system - version 1 / fix 1.
6. Keep a list of the fixes you have installed and in which order.

2.3.1.4 Going to a Major System Maintenance Level

You have been using your AIX level for long time and now you need to update to a brand new level. Updating is very close to starting a new system as described in 2.3.1.2, “Initial Machine Installation” on page 8. The differences are that you may be able to generate your new system by applying update tapes from IBM on top of your existing system (for instance upgrading from AIX 3.2.0 to AIX 3.2.4). Some other time you may need to use more complex migration tools (for instance from AIX 3.1.5 to 3.2.0). Figure 7 describes how this can be done.

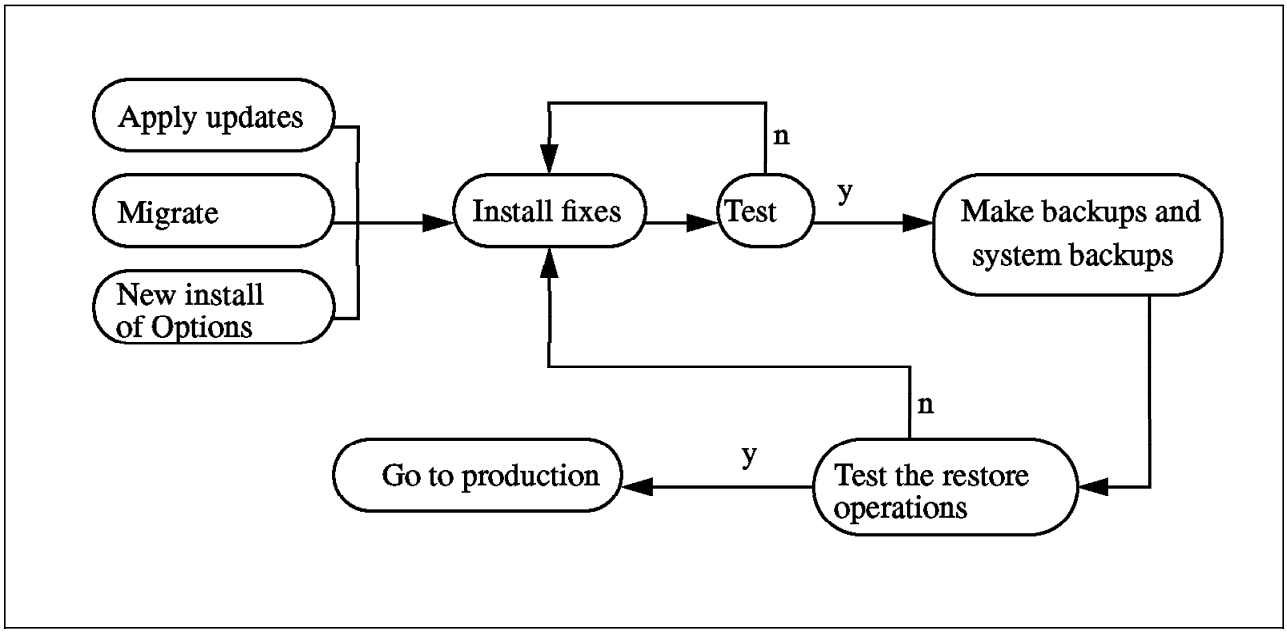

Figure 7. Going to a Major System Maintenance Level

2.3.2 Unattented Servers Scenario

Scenario abstract: A customer with many RISC System/6000s running unattended in many distributed computer rooms without any local operators. The management is centralized on one RISC System/6000 connected through a TCP/IP network to all the other ones. Figure 8 on page 11 give you a typical network architecture for this scenario.

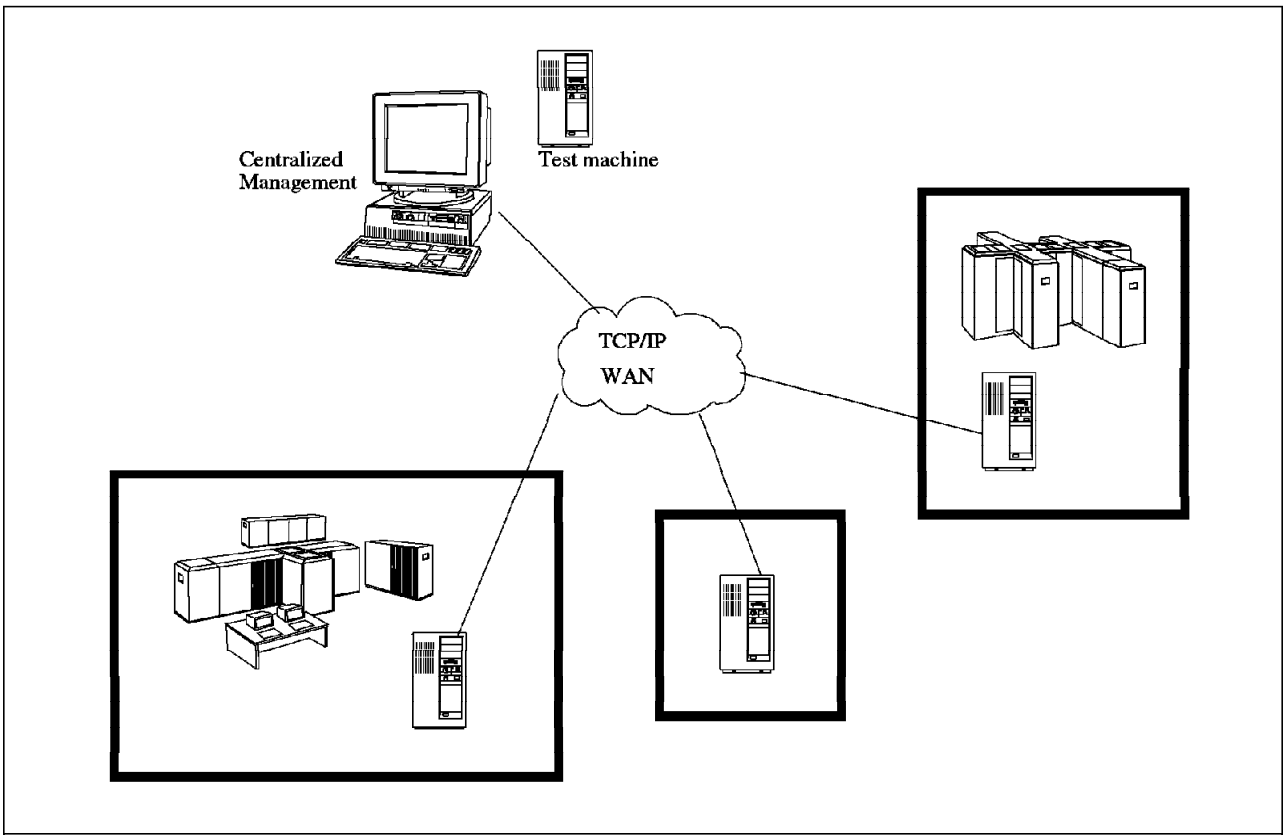

Figure 8. Unattended Servers Scenario

2.3.2.1 Maintenance Life Cycle

Compare to the *single machine* scenario, the unattented servers have some new steps to face the following new constraints:

- Multiple machines
- Wide geographical spreading of the machines (may be international)
- No local knowledgeable operators, just *tape mounting* operators

In such a situation, you will have to use a test machine to build and validate your system before sending it to operational sites. It is always very dangerous to apply maintenance directly on the production machines.

Figure 9 on page 12 gives you what could be a maintenance life cycle in such a scenario. This is a kind of permanent process that you should use for any new maintenance you need to apply on any of your systems. This process is sometimes difficult to manage but this is the price to have no knowlegeable operators to manage every single machine. This process tends to industrialize your maintenance process.

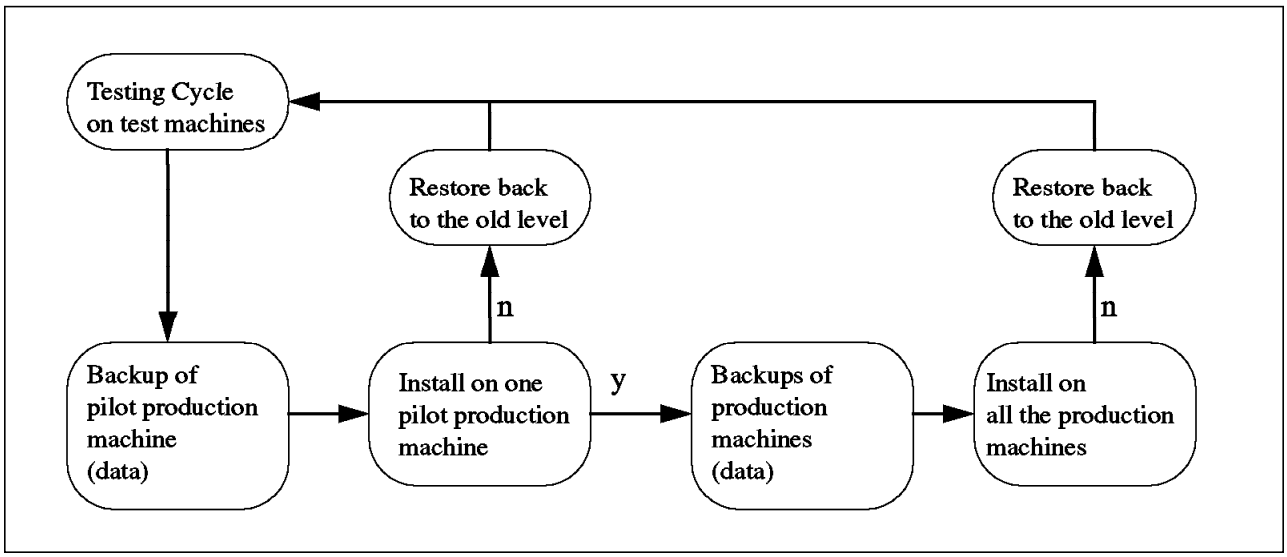

Figure 9. Unattended Servers Scenario Maintenance Life Cycle

2.3.2.2 Test Cycle on Test Machines

In this scenario, you have too many critical business machines spread over many geographical sites to apply the changes and fixes directly to production machines. You need test machines to prepare and test the changes, the fixes and the installation procedures.

The test cycle quality is very important. If maintenance has been applied to many production machines, and you have problems, they will be very difficult to diagnose. Your test cycle should be similar to the whole *single-machine scenario* described before. Your test machine design should be like a small production system.

2.3.2.3 Backups of Production Machines Data

Before applying any fixes or changes to a production machine, you must first make a valid backup of the application data. A system and application code backup should not be necessary because they should already exist in your central management site and nothing should be different in the production site than what you have in your central repository. However, the data on the production machines may be backed up every day at midnight or every week. So you must ensure that before applying any new maintenance, you've made a data backup. This may require some temporary changes in the backup policy. See Chapter 5, "Application Data Backup Strategies" on page 51.

2.3.2.4 Pilot Site

Even for simple maintenance, you should never send it to your whole production environment first. You should apply the fixes and changes to a pilot site and let it run for a period of time before deciding to install them on all the sites. Some problems may occur in very uncommon conditions which are difficult to reproduce in your test cycle. You need to check that everything is correct on the pilot site. You should always have a procedure ready to bring the pilot site machines back to their old level if you find a problem.

2.3.2.5 Distribution Mechanism

As your production machines are in different geographical sites, you may want to use the TCP/IP network for installing maintenance. To do that, you must first take into account the size of the files you have to send compared to the bandwidth of your network. See Chapter 5, “Application Data Backup Strategies” on page 51 for information on backups and bandwidth. Then you should decide which technique you will use to apply or reject the maintenance on the machines.

- If the maintenance is a whole system image that you will put in place of the old system with a `mksysb` image, you have to use a *network install* procedure. This could be done using built-in AIX functions or using the product AIX SoftDist/6000* (see Chapter 3, “Application Installation” on page 17). In both cases, you must include the test of the network install techniques in the test cycle to be sure that the result of your installation is identical of what you expect doing a tape installation.
- If the maintenance is only an update, the files size will probably be smaller. You must be very careful with the installation procedure. If it is just standard IBM fixes (using `installp`), the built-in logic should guarantee that the installation conditions are observed (for instance the free disk space or the prerequisite fixes). If you have made the installation procedure yourself, be sure (and test) that all the condition checkings are included. A good way for doing that is to build your own `installp` image.

When Should You Use an Update Mechanism or a Full System Image?

When you plan a software maintenance operation for a large number of machines, you must ask yourself what is the safer way to install this maintenance on the final target machines:

- Use an update mechanism on top of the existing system. This *update* could be an `installp` image or your own installation script.
- Build a complete image of the new level of the system (using the `mksysb` command) and reload the target machines from this image.

2.3.3 Networked Workstations (Decentralized Management) Scenario

Scenario abstract: A customer with many RISC System/6000 workstations. The workstations are under the control of their user(s). The central management department is working as a support center. Figure 10 on page 14 shows typical network architecture for this scenario.

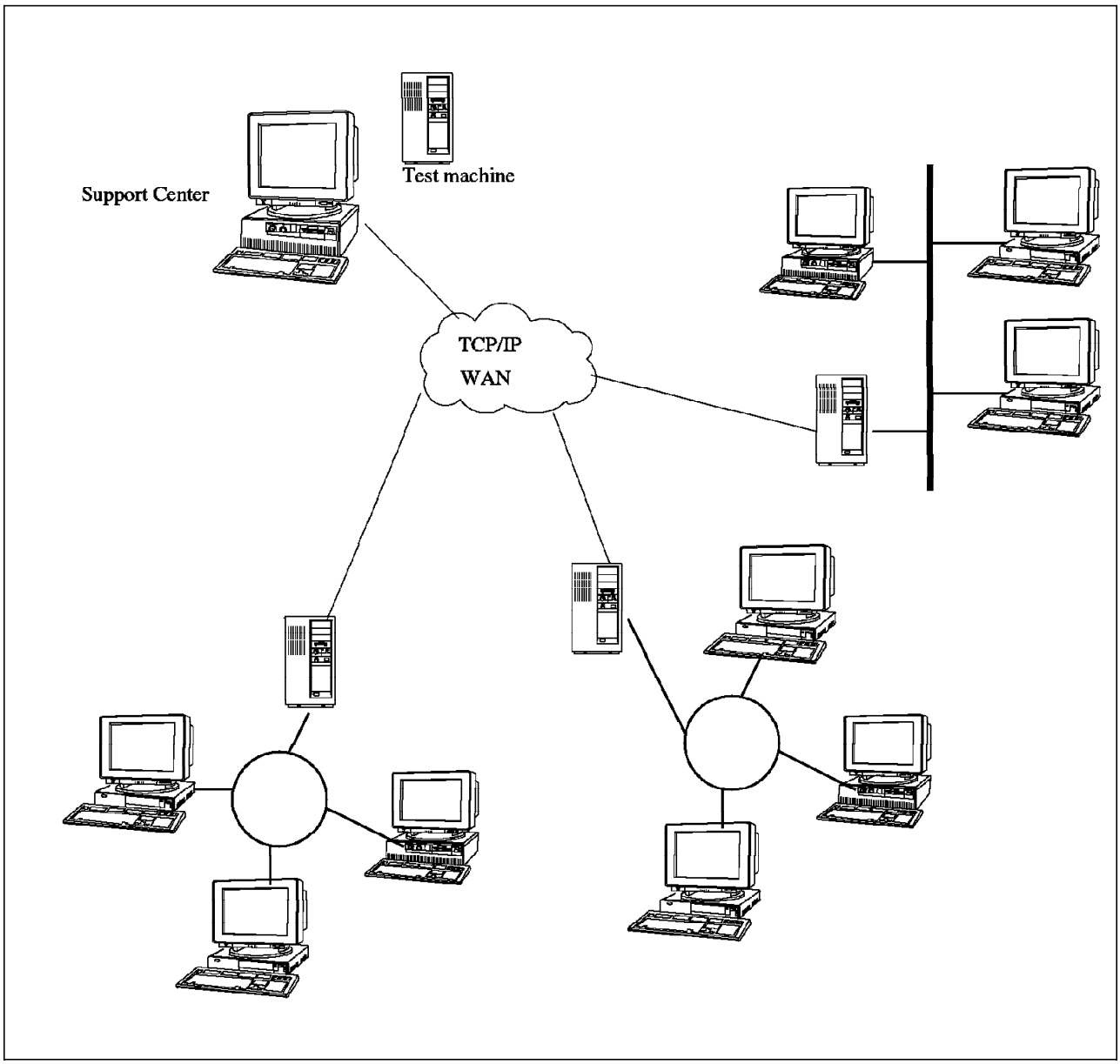

Figure 10. Networked Workstations (Decentralized Management) Scenario

2.3.3.1 Maintenance Life Cycle

In this scenario, the maintenance scheme could be seen as *client/server*. The maintenance is not applied on the users machine but the users have to request and apply what they need. The centralized maintenance organization behaves now as a support organization. This assumes experienced users with a good understanding of computers and a minimum understanding of computer maintenance. If this is not the case, then you'd be better off with a centralized management scenario.

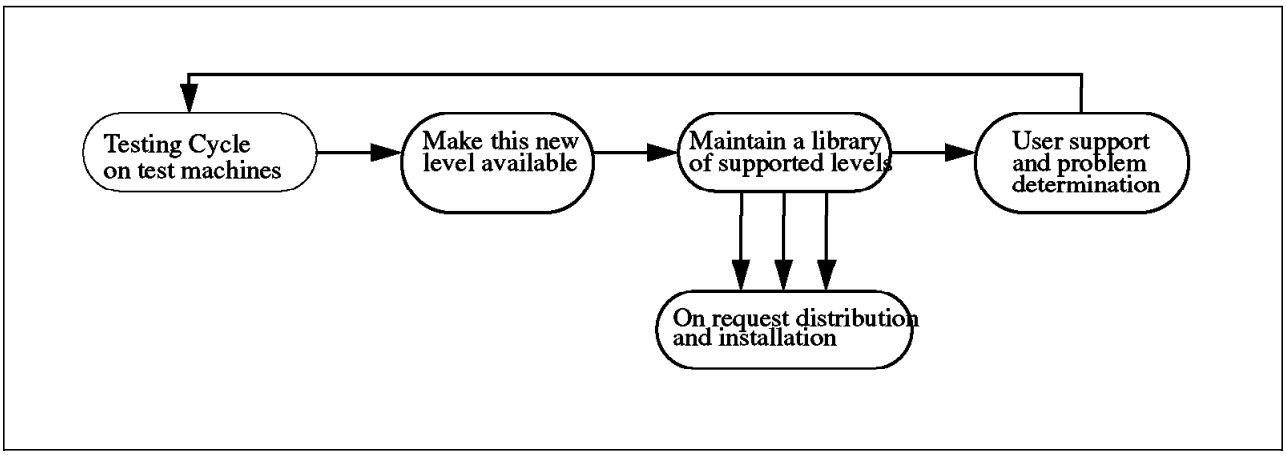

Figure 11. Maintenance Life Cycle for Decentralized Management (Support)

2.3.3.2 Test Cycle

Once again, the test cycle looks like the whole *single-machine* scenario. This test cycle on test machines let you verify that what you send to the user machines will not break them.

2.3.3.3 Make a New Level Available on Request

In the other scenarios, we have built and tested the remote installation techniques used to distribute the maintenance code. On top of that, now you need to make those installation procedures easy enough, well documented enough and safe enough for end-user usage. This may not be a simple task.

2.3.3.4 Maintain a Library of Supported Levels

Using this scenario, the user can decide when he will upgrade to a certain level of maintenance. You will have to maintain a wider number of maintenance levels. This is more complex than in the centralized scenario where you mainly have to maintain level n-1 and n.

You will have to implement well known policies about maintenance level support. The users should know the dates support will be withdrawn. The support center should help the users track the levels of their machine to keep them at supported maintenance levels. You should also assure *prerequisite* and *corequisite* chains for installation of new products and/or maintenance. SoftDist/6000 could provide both functions. See detailed information about SoftDist/6000 in Chapter 3, “Application Installation” on page 17.

2.3.3.5 On Request Distribution and Installation

If you want the user to order his maintenance himself, you need a *simple* and *easy* ordering mechanism. This user interface should also check most of the user errors. SoftDist/6000 gives you theses facilities.

Chapter 3. Application Installation

A complete system is made up of three components: the operating system, the applications and the data (Figure 12). The operating system itself could be divided into base operating system and operating system enablers. The base operating system, the operating system enablers and the applications components have machine specific configuration data. This description is necessary to understand why we manage those components in a different way. In Chapter 2, "Building and Maintaining the AIX You Need" on page 3 we discuss managing the operating system components. In Chapter 3, "Application Installation" we will discuss the application data components. In Chapter 7, "Configuration Management" on page 79 we will discuss the base operating system and operating system enablers configuration data. Example of components are:

- Base operating system: AIX Version 3
- Base operating system configuration data: user names and passwords kernel parameters
- Operating system enablers: TCP/IP (bosnet), AIX SNA Server/6000, AIXwindows/6000, Oracle** Database
- Operating system enablers configuration data: TCP/IP address and parameters, SNA profiles, X Window** user defined resources (`.xinitrc`)
- Applications: Lotus** 1-2-3** for UNIX**, a payroll program using Oracle Database, CATIA**
- Application configuration data: name of the company used by the payroll program, Oracle Database parameters (size of the log files, size of the memory buffers), CATIA user names
- Application data: user Lotus spreadsheet files, the salaries SQL tables, the files with new plane wings CATIA models

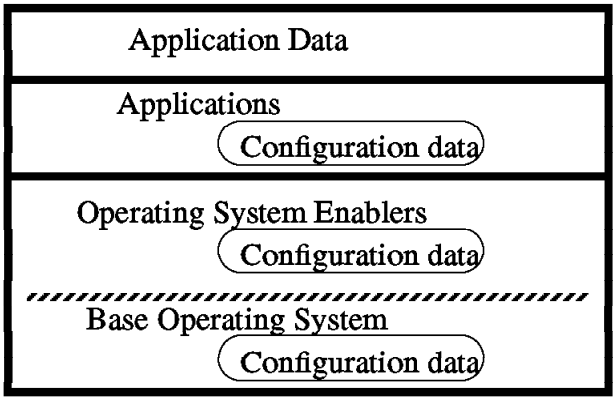

Figure 12. Components that Make a Complete System

Why make a difference between application, operating system and data?

© Copyright IBM Corp. 1993,1994

You have to differentiate those three layers because of differences in system managements techniques for those three layers. Maybe in a few years, with OSF/DME** becoming a broadly used tool, system management will become a homogeneous task across the three layers. Today, this is not the case.

In this chapter, we will see what techniques and tools are available to manage application code and application configuration for AIX machines in the enterprise.

3.1 Organization Considerations

In your enterprise the users will request new applications more and more quickly. As many applications are now available directly on the market, and not made by the enterprise′s internal development people, the users wants to see the application they just bought installed on all the enterprise machines by ... yesterday ! Even with UNIX machines, if you have many production machines centrally managed, you will have to set up a minimum organization to manage application installation. In a business environment, the application installation process is not just ″take the tape and load it″. If you do that, you will have many difficulties in maintaining and trouble shooting your systems.

There are three key elements for making the whole applications installation process successful:

- The application itself. The programs, the documentation, the installation procedures. This is different for any new application (Application specific).
- The enterprise organization responsible for managing the application installation process. This is specific to your enterprise (Enterprise specific).
- The way the application installation process works on a particular machine. Installing the application on machine A is not exactly the same as installing this application on machine B. The configuration will be different (Machine specific). According to the size of your enterprise and the complexity of the applications, those elements can be more or less complex, but they should always be there (Figure 13 on page 19).

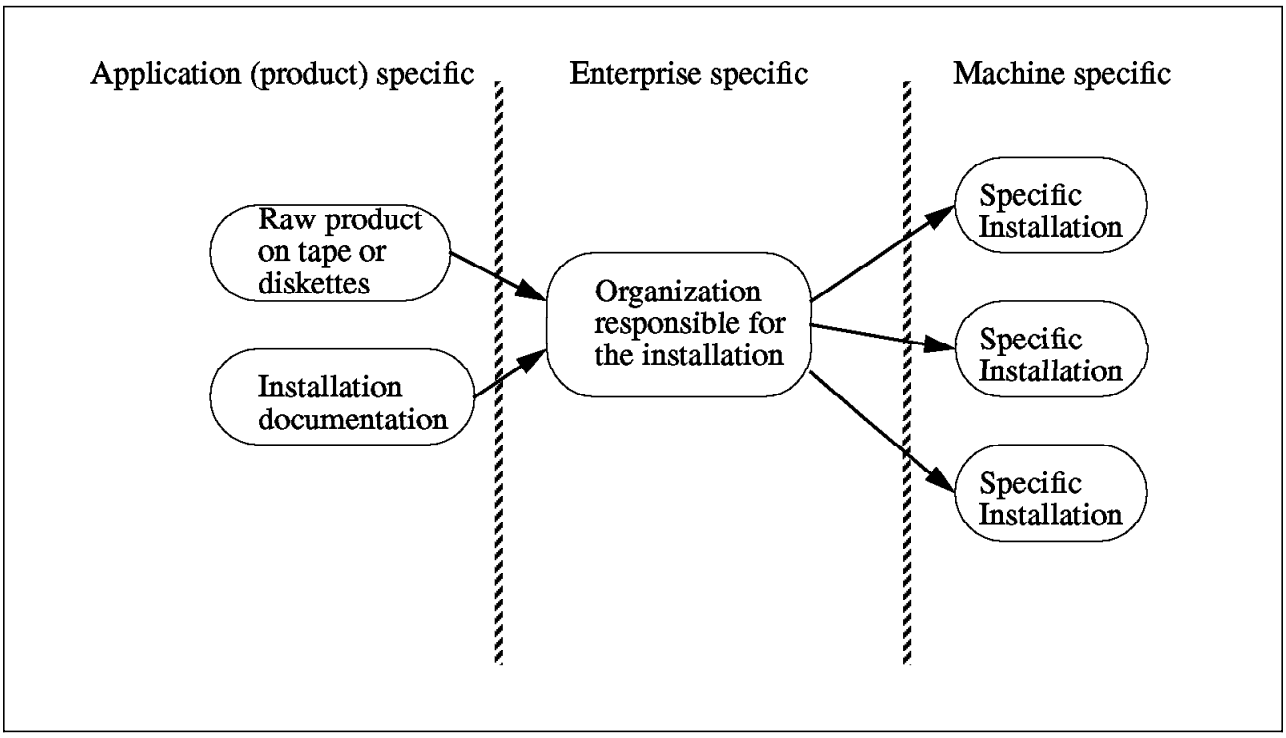

Figure 13. Typical Organization for Applications Installation

3.1.1 Application Needed Characteristics

An application must come with a defined set of documentation and software that gives the minimum confidence to the installation organization on what they are doing. Quality and accuracy of the documentation is very important. Very often, when the application is coming from the internal enterprise programmers, the installation documentation is very poor or not up to date. It is a good idea to have a written specifications document describing those requested items, especially for internal developers.

3.1.2 Organization Responsible for the Installation Process

The organization responsible for the installation process is key to assure the security of the operation. It's making the link between an independent application with the context of the enterprise information system. As described in Chapter 2, "Building and Maintaining the AIX You Need" on page 3, this organization is responsible for the integration tests between the new application and the existing system. Based on the complexity of the application and on the predictable impact of the application to the existing environment, the integration test will be more or less complete. Formal checking should always be done for any new application to verify if the correct level of integration testing and preparation work has been done. When the application is installed on the production machines, it will be too late.

3.1.3 Installation of an Application on a Particular Machine

In Figure 14, you can see that you will get many different *data sets* during the installation of an application. A data set is an elementary set of data (program and files). At the beginning you have the application from the vendor or your developers, then you have specific installation procedures to insert this application in your enterprise specific environment. Then, when you install this application to machine A, you get specific machine A configuration information for this application.

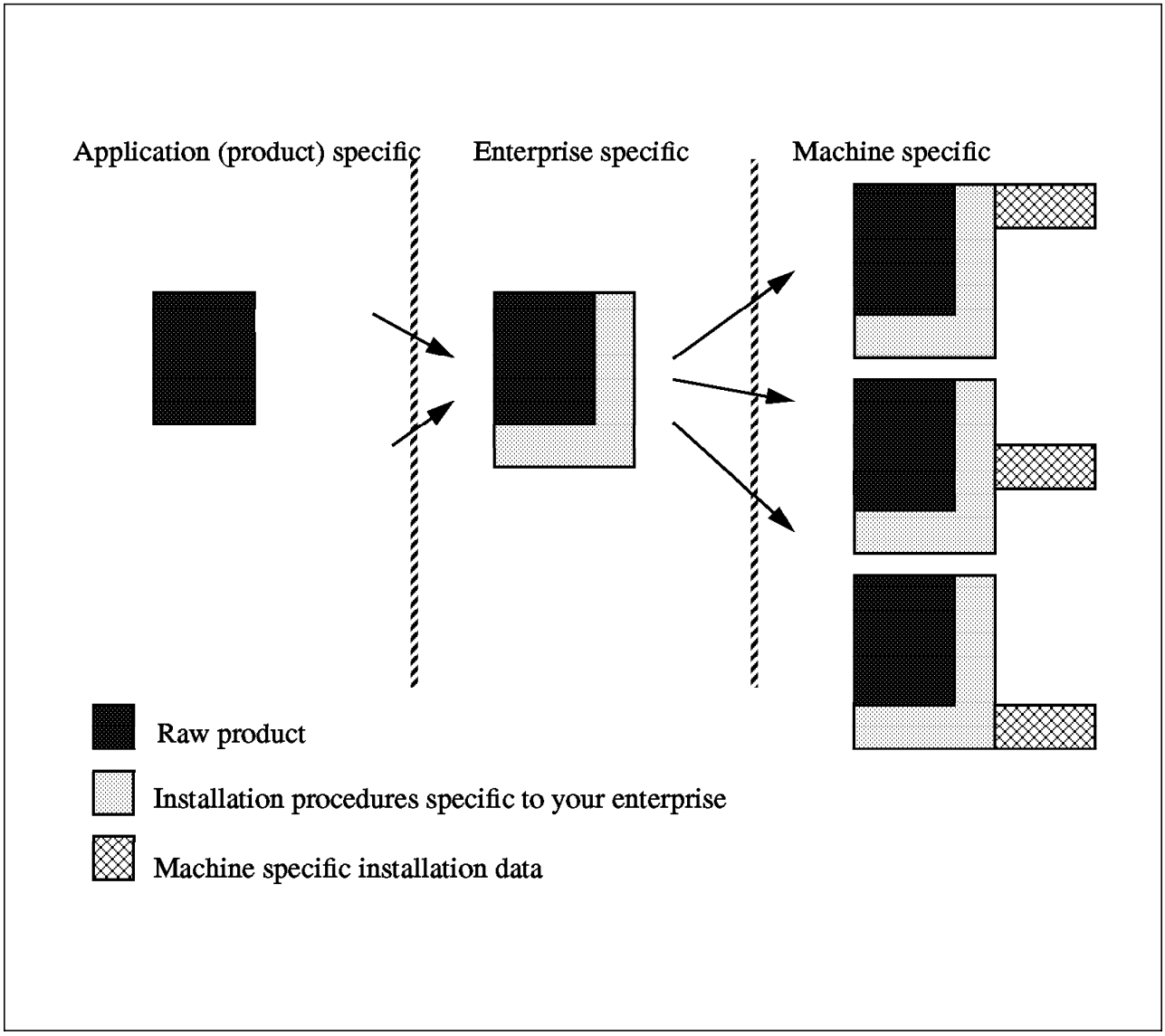

Figure 14. Application Data Sets During the Installation Phases

This customization is stored in some application files. These file are not really part of the application code itself but nor are they really application data. After being created, they must be backed up and associated with the application backup. In case of a crash, when you need a quick restore, you will have to find both the application code backup and the customization backup. Enterprise wide, having a backup of a given application means having backups off all individual data sets (Figure 15 on page 21). In case of problem, when you will need to restore the application on a given machine, you will have to recreate the particular data sets for this machine.

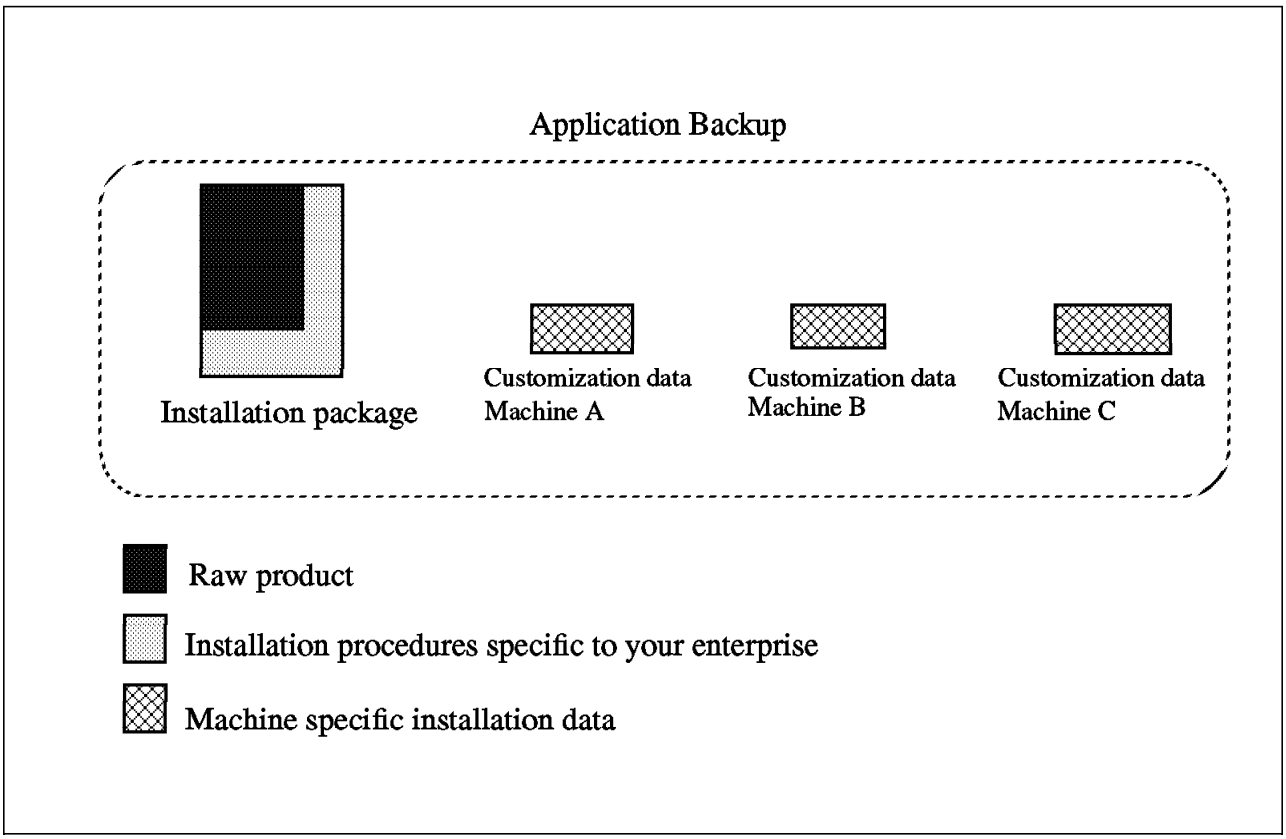

Figure 15. Full Application Backup Requirements

3.2 Single Machine Scenario

Scenario abstract: A customer has only one RISC System/6000 machine. This machine is used to run critical business applications. The typical configuration is a RISC System/6000 with asynchronous terminals.

In this scenario, there is not much choice. Usually the enterprise is not large enough to have a real *organization* specific to installation and integration tests. There is mainly one question you should answer: Do I trust this application enough to go through a direct installation or do I need a full sub-system installation test as described in 2.3.1, “Single Machine Scenario” on page 8?

There is no definitive answer to this question. The main problem is: how isolated is my application? For instance, if your new application is working inside a relational database engine like AIX DB2/6000* or inside a transaction monitor like AIX CICS/6000*, the impact on the base operating system should be very little. But if your application is complex C written code, running sometimes under root privileges, you should go for a full test.

3.3 Unattended Servers Scenario

Scenario abstract: A customer with many RISC System/6000s running unattended in many distributed computer rooms without any local operators. The management is centralized on one RISC System/6000 connected through a TCP/IP network to all others. See Figure 8 on page 11.

In such a scenario, your organization should be strong. The cases where you do not go through a full test should be very limited. The local customization data generated during the installation of the machines should be backed up centrally.

As you won't have on site knowledgeable personnel to take care of your installation and backups, you will need to use distributed installation and backups tools like AIX NetView DM/6000 and Legato** Networker**.

3.4 Networked Workstations (Decentralized Management) Scenario

Scenario abstract: A customer with many RISC System/6000 workstations. The workstations are under the control of their users. The central management department is working as a support center. See Figure 10 on page 14.

In this case, the installation of a particular application will be initiated by the user. That's why it is very important that the installation procedure was as automatic as possible and takes care of all possible error conditions occurring during the installation. In more centralized scenario, the installation on the machines are always done under control of knowledgeable people (even remote). In this scenario, the installation must be simple and safe to be made by the user. Tools like NetView DM/6000 and Legato also let your users make their installations and backups in a user friendly environment.

3.5 Using AIX NetView Distribution Manager/6000 for Application Distribution

AIX NetView Distribution Manager/6000 Release 1.1 is a product to help large network system administrators (AIX*, OS/2*, DOS and DOS Windows**, SUN**, and HP**) deal with the application installation and distribution tasks. In a network of target workstations, a central RISC/6000 acts as a *Change Control server* (CC server), providing services for software and data Distribution and Change Control.
Some of the characteristics and functions are:

- AIX NetView DM/6000 is client/server based. Each client workstation must have installed the *Change Control client* (CC client) program, which operates in a client/server relationship with the CC server NetView DM/6000. Available clients are NetView Distribution Management Agent/6000 (NetView DMA/6000), NetView DMA/2, NetView DMA for Windows and NetView DMA for HP-UX**.
- The user interface is a Motif** or a command line interface. This same interface is used for administration of AIX NetView DM/6000 and requests for services from users.
- The server is a repository where all the applications are stored in a NetView DM/6000 format.

- NetView DM/6000 can perform the Change Control operations: install, accept, uninstall, remove, activate, execute, authorize/unauthorize on change files at local or remote targets and target groups over a TCP/IP or an SNA network.
- NetView DM/6000 can perform the Distribution operations: send, replace, retrieve, delete, authorize/unauthorize files to local or remote targets and target groups.

You can use AIX NetView DM/6000 in a centralized or decentralized manner:

- If you are decentralized, the user *pulls* the application from the server to his machine (see Figure 16, right). The pull operation is a Client user initiated installation.
- If you are centralized, the central installation organization *pushes* the application from the server to the target machines (see Figure 16, left). Distribution and installation may be scheduled.

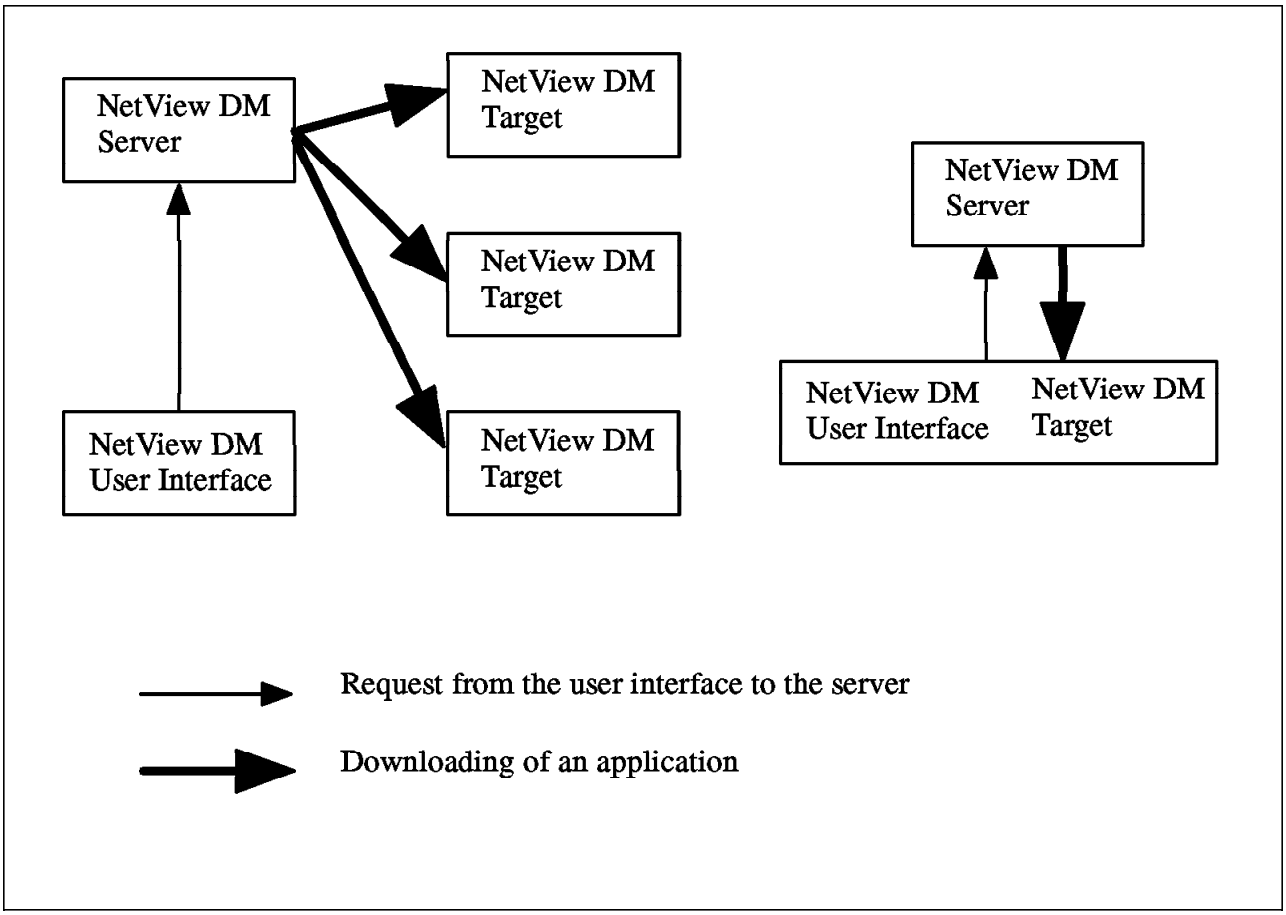

Figure 16. AIX NetView DM/6000 Client/Server General Architecture

3.5.1 User Interface

The user interface for the administrator on the server (NetView DM/6000) and for the user on the client (NetView DMA) is the same (running either Motif or the command line interface). Having the administrator sending (push) an application to a user machine, or the user requesting this application to be installed on their machine is basically the same operation.

Before you start the NetView DM/6000 user interface, you have to configure the server and client by editing their base configuration file, /usr/lpp/netviewdm/db/nvdm.cfg, in the following way for the server:

```
WORKSTATION NAME:     andrea
MESSAGE LOG LEVEL:    N
LAN AUTHORIZATION:    0
CONFIGURATION:        REMOTE_ADMIN_SERVER
MACHINE TYPE:         AIX
LOG FILE SIZE:        500000
TRACE FILE SIZE:      1000000
API TRACE FILE SIZE:  500000
TCP/IP PORT:          729
MAX TARGETS:          600
MAX CONNECTIONS:      50
MAX USER INTERFACES:  20
SERVER:               andrea
REPOSITORY:           /usr/lpp/netviewdm/repos
SERVICE AREA:         /usr/lpp/netviewdm/service
BACKUP AREA:          /usr/lpp/netviewdm/backup
WORK AREA:            /usr/lpp/netviewdm/work
```

Figure 17. Server Base Configuration File nvdm.cfg

An example of base configuration file for an AIX DMA/6000 is:

```
WORKSTATION NAME:     francois.austin.ibm.com
MESSAGE LOG LEVEL:    N
LAN AUTHORIZATION:    0
CONFIGURATION:        CLIENT
MACHINE TYPE:         AIX
LOG FILE SIZE:        50000
TRACE FILE SIZE:      1000000
API TRACE FILE SIZE:  100
TCP/IP PORT:          729
SERVER:               andrea.itsc.austin.ibm.com
REPOSITORY:           /usr/lpp/netviewdm/repos
SERVICE AREA:         /usr/lpp/netviewdm/service
BACKUP AREA:          /usr/lpp/netviewdm/backup
WORK AREA:            /usr/lpp/netviewdm/work
```

Figure 18. AIX Client Base Configuration File nvdm.cfg

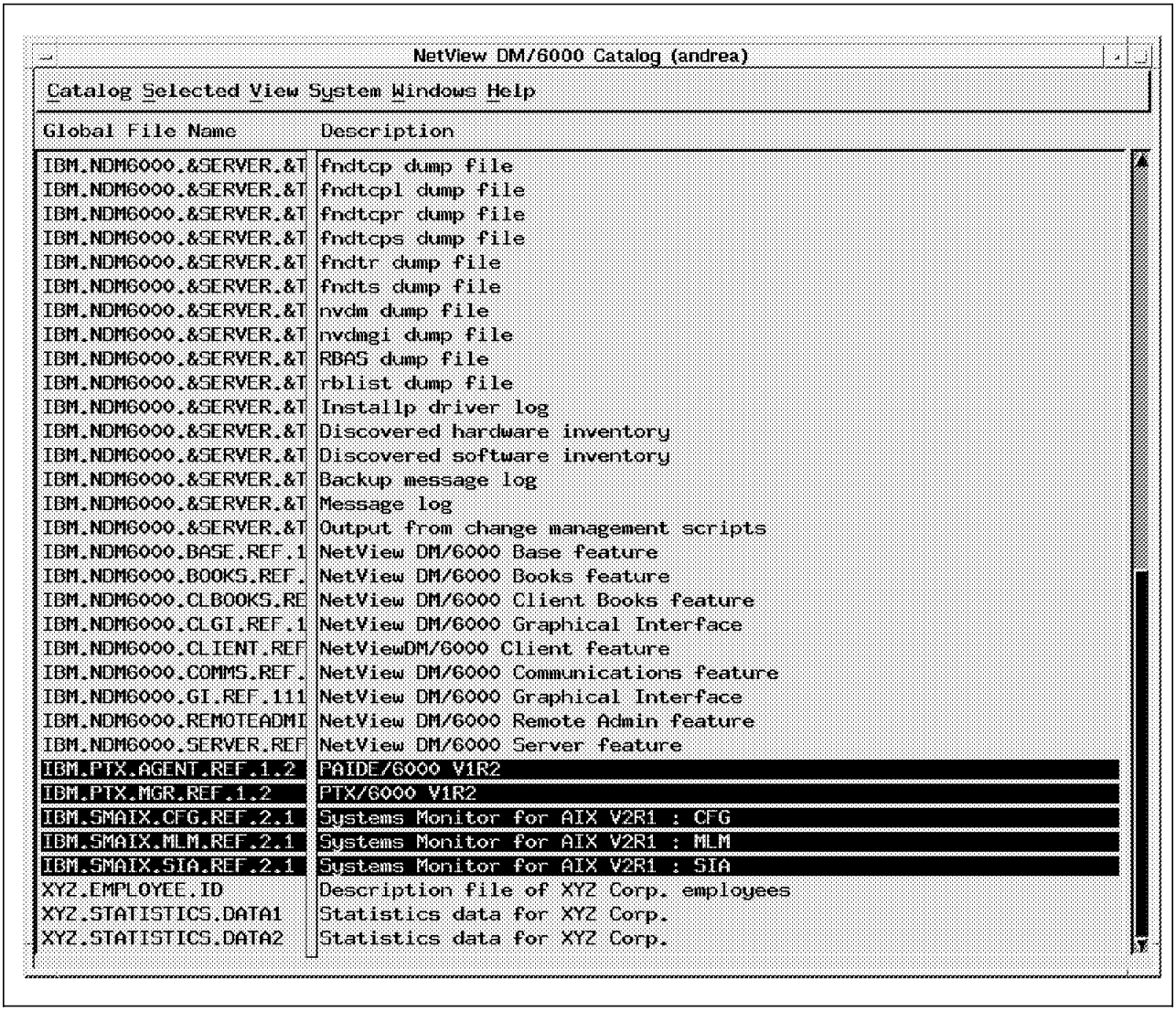

Figure 19. NetView DM/6000 Catalog Window

Now, you can start the background daemons with the `nvdm start` command (to stop them, use the `nvdm stop` command). Running the `nvdmgi` command on the CC server starts the graphical user interface, giving access to the Catalog window (see Figure 19). The other windows are the Targets window, the Queues window, the Message Log window, and the Help window.

Starting the GI from a CC client lets you access the same windows as on the CC server, as shown in Figure 20.

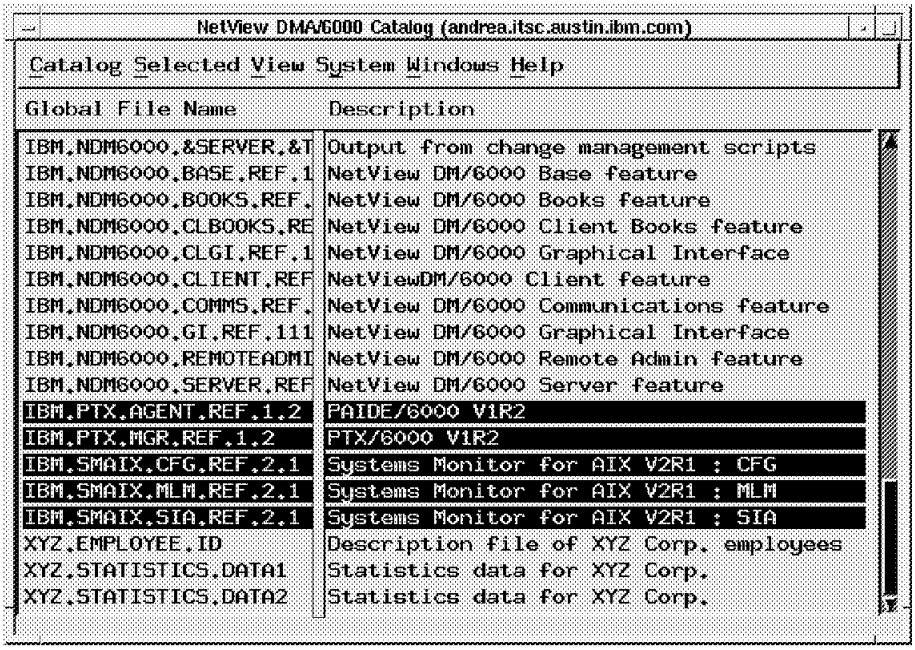

Figure 20. NetView DMA/6000 Graphical Interface

3.5.2 Defining NetView DM Users

To allow a user to run NetView DM/6000 graphical user interface (on the CC server or target machine), you have to define it to the operating system at both the target and the CC server. The user must belong to one of NetView DM/6000 user groups:

- FNDADMN: Defines the *administrator* profile
- FNDBLD: Defines the *builder* profile
- FNDUSER: Defines the *user* profile

You must give one of these values to the primary user group and the administrative user group, while creating or modifying the user on the CC server.

Administrators have access to *all* operations, including the administrative and configuration functions (during the CC server installation, the root user is automatically defined with the administrator profile). Builders are authorized to perform CC preparation functions. They can prepare and build change files. Users can run distribution tasks and display the configuration.

The graphical user interface (GUI) for the builder and the user are obviously different from the administrator's GUI. The user profile has indeed very limited choices in his windows (no targets window, no creation of change files allowed, and so on).

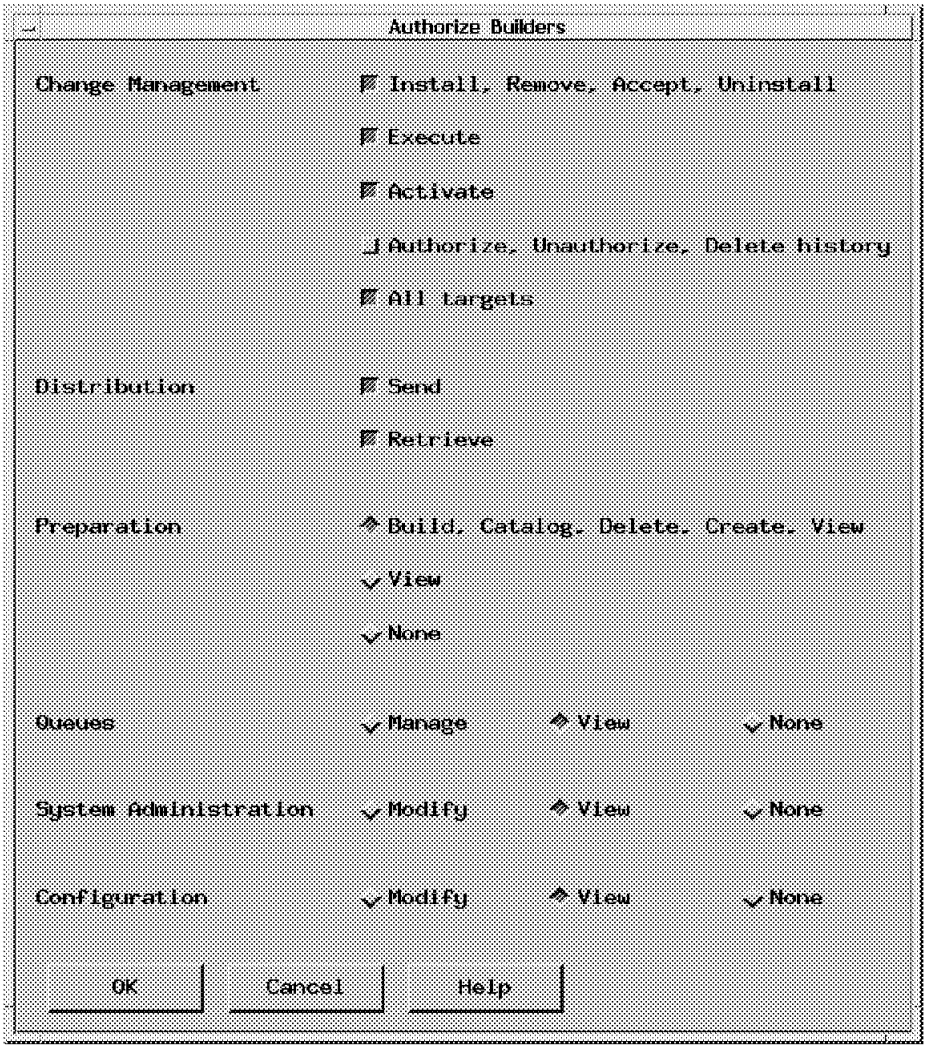

Figure 21. NetView DM/6000 Builder Profile

Figure 21 shows an example of a builder profile.

3.5.3 Defining NetView DM Targets

At this point, you must define the targets to which change control and distribution activities will be directed. There are three types of targets:

- Local target: Targets in the same CC domain as the CC server they are being configured for. Two distribution modes are available: the push mode (tasks are controlled by the administrator or builder at the CC server or from a focal point target), and the pull mode (a user at the target initiates the tasks).
- Remote target: Targets in a different CC domain as the CC server they are being configured for,
  accessible over TCP/IP and/or SNA networks (see Figure 22 on page 29).
- User interface only (UI only) target: Targets used to run the NetView DM/6000 user interfaces (GUI or command line). This mode is useful when you have an environment in which more than one CC server exists. It allows an

administrator to access all CC servers from the same target either to perform administrative tasks or to schedule distributions to targets.

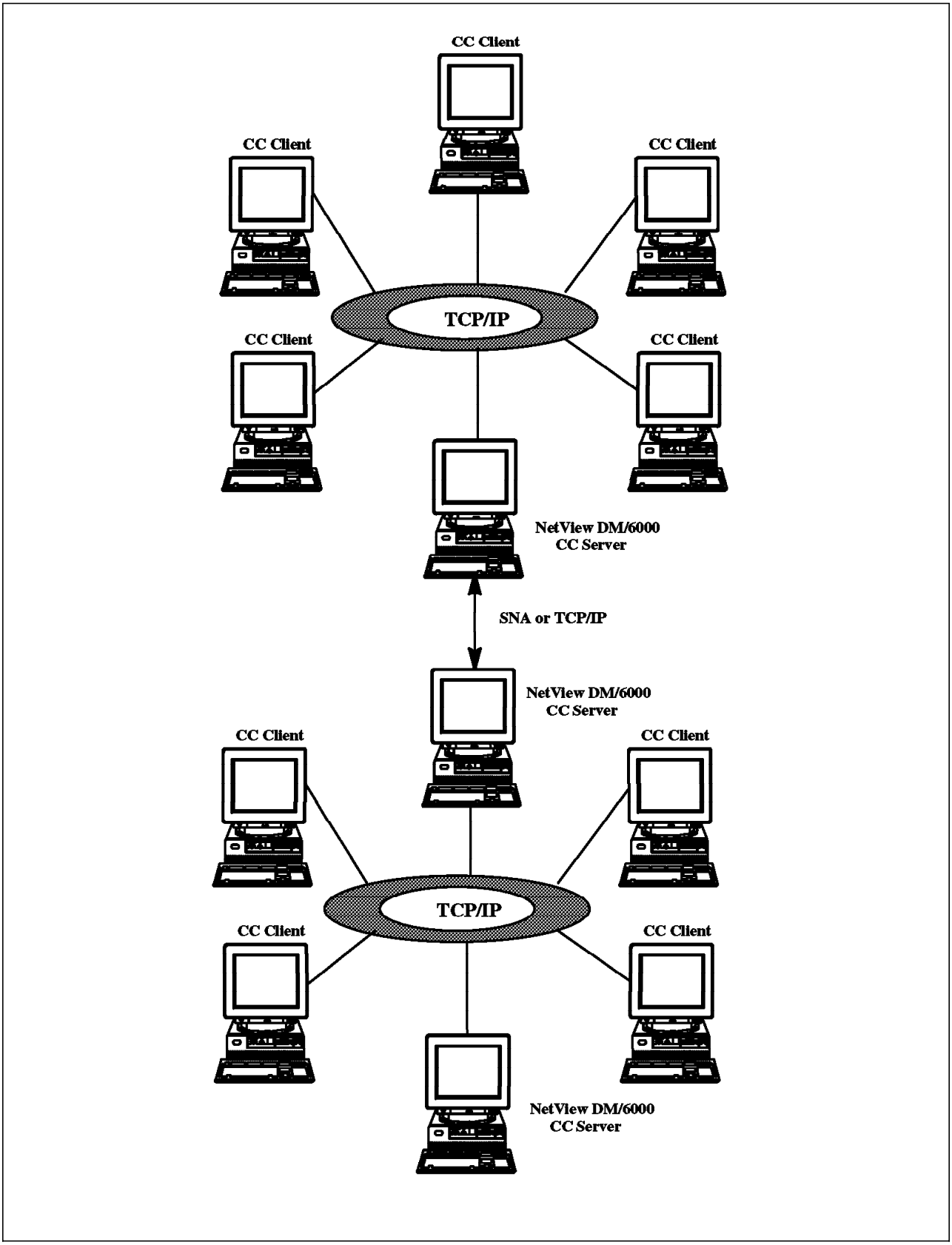

Figure 22. Interconnected CC Domains

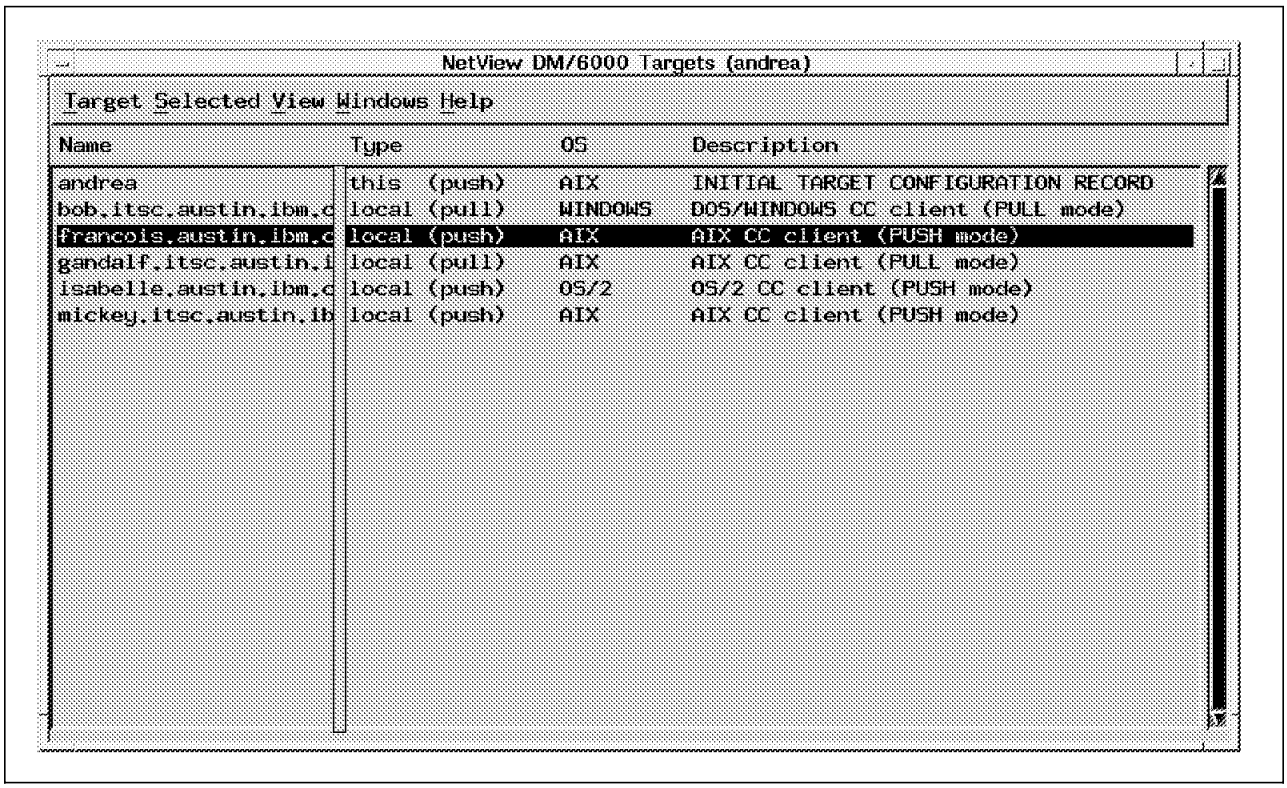

Figure 23. NetView DM/6000 Targets Window

Figure 23 shows a list of created local target entries. Some of them installed the AIX CC Client code, others the NetView DMA/2 or NetView DMA/Windows agent programs. The push or pull distribution mode is specified. It is also in the targets window that you specify which users on the target can perform administrator, builder or user tasks.

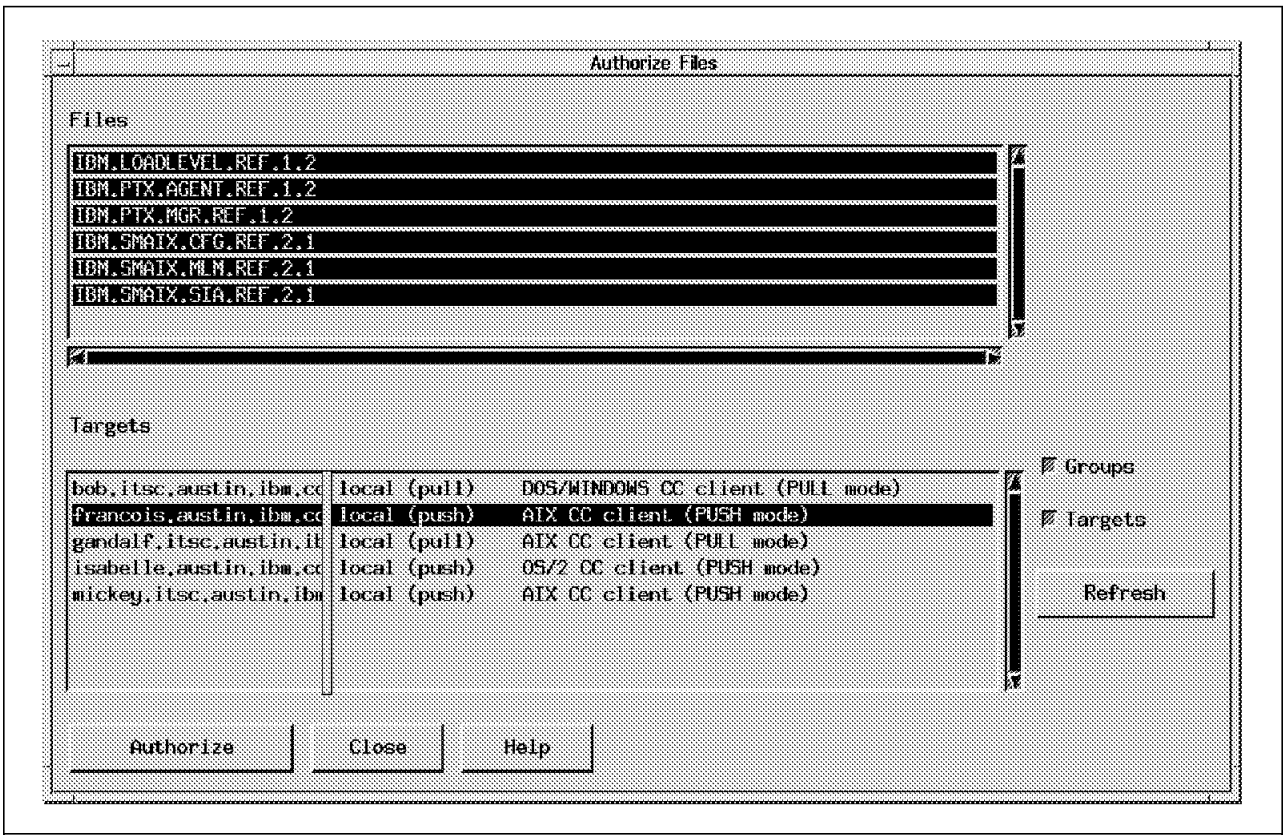

Figure 24. Authorizing Files to Targets

After having defined all the targets in your CC domain, you may want to authorize catalog entries so that only specified targets can access them. Figure 24 shows how to perform this action.

3.5.4 Performing Change Control Tasks

Change Control tasks consist of change file creation and their installation on targets.

3.5.4.1 Creating Change Files

Before you install or update software ona target workstation, you have to create a change file entry for the software. The change file will include software files together with instructions concerning how they are to be installed. When you create a change file, you have to give it one of the following types:

- AIX installp
- OS/2 CID
- UNIX generic
- OS/2 FS generic
- Windows generic

Change files are also categorized in three kinds:

- Refresh file: Contains a completly new copy of the software.
- Update file: Contains an update to a software. The updates will change the level of this software.

- Fix file: Contains a fix to a software. Fixes do not change the level of the software.

A change file will include specifications like:

- Software prerequisites: Specifies other change files that must already be installed on the target system
- Hardware prerequisites: Gives the hardware which a target must be equipped with before the change file can be installed (for example, `Display = VGA`, `system_ram > 32000`)
- Compression options: Compression during transmission converts the change files into a format that requires less space on the disk, and enables you to transmit them faster to the targets. Compression techniques available are SNA or LZW.
- Scripts: Commands to be executed before and/or after change control or distribution operations (see Figure 25)

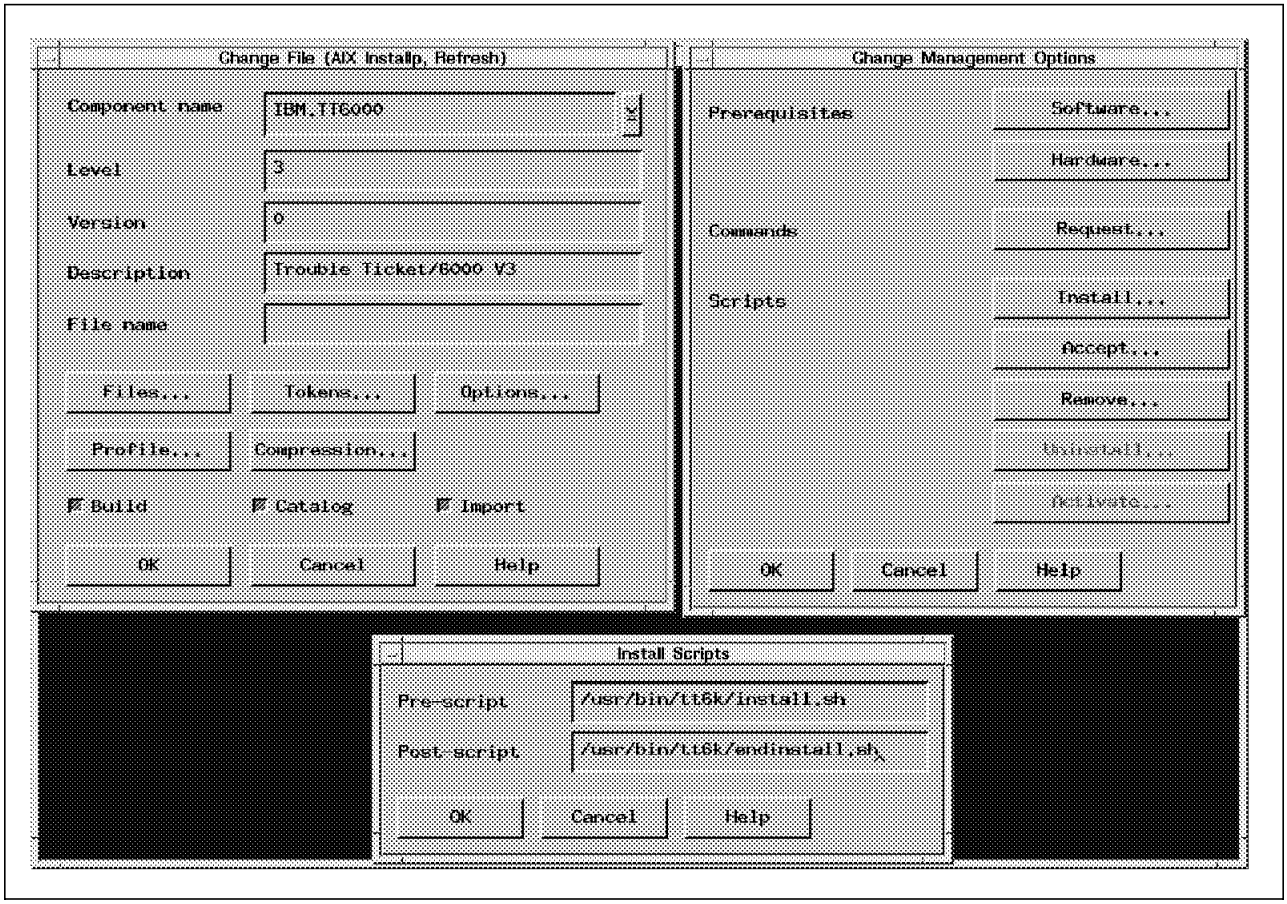

Figure 25. Creating a Refresh File

To build and catalog a change file, you have to give it a global name composed of a component name (for example, IBM.TT6000 in Figure 25), a change name indicating what kind of file it is (refresh, update or fix) and which level it has, and a version (optional). You can ask to build it on your local workstation, catalog it into the Catalog window, and import (send) it to the CC server.

3.5.4.2 Installing Change Files

After storing the change file on the CC server, you can process its installation onto desired targets. You can choose to schedule or do that immediatly. You have to specify some installation options like wait for an activation before making changes to the hard disk, install the change file removable, auto-accept it after installation is done, install it with other change files in a single operation (install them as *corequisites*), and others.

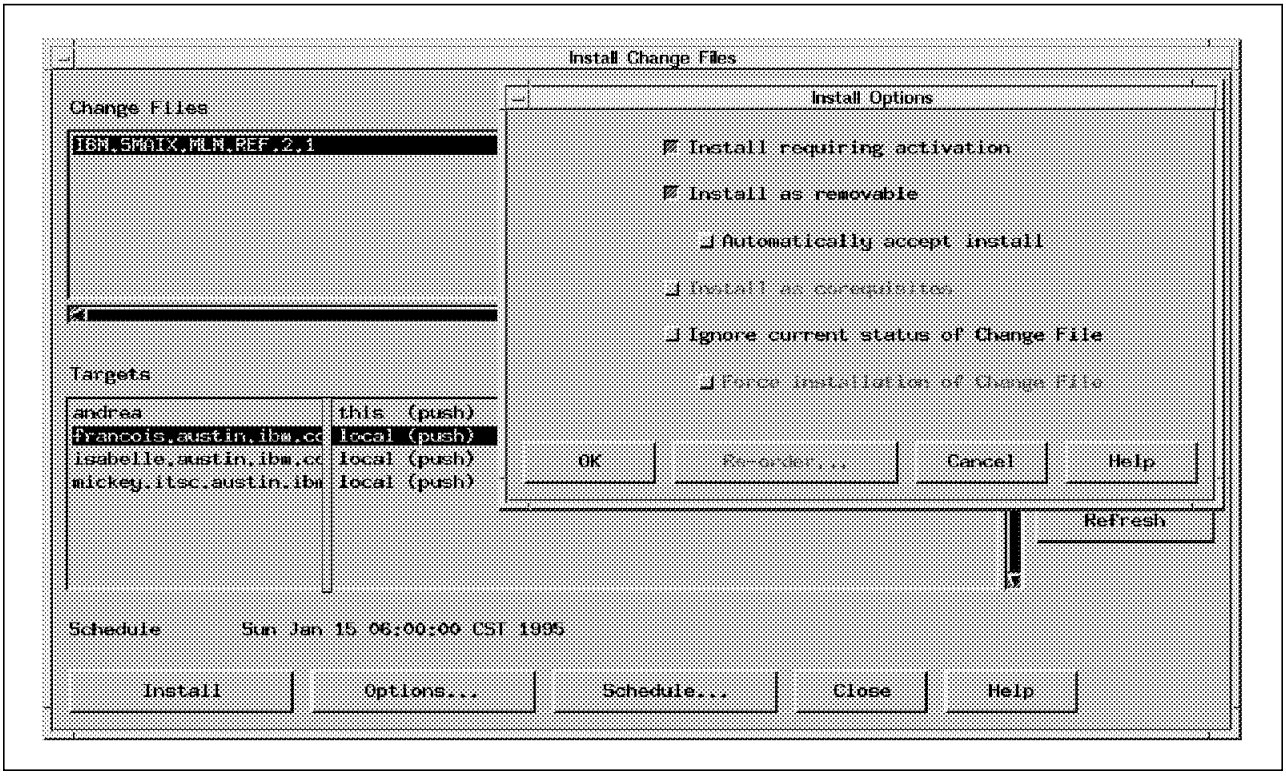

Figure 26. Installing on a Target

Figure 26 shows the NetView DM/6000 installation panel.

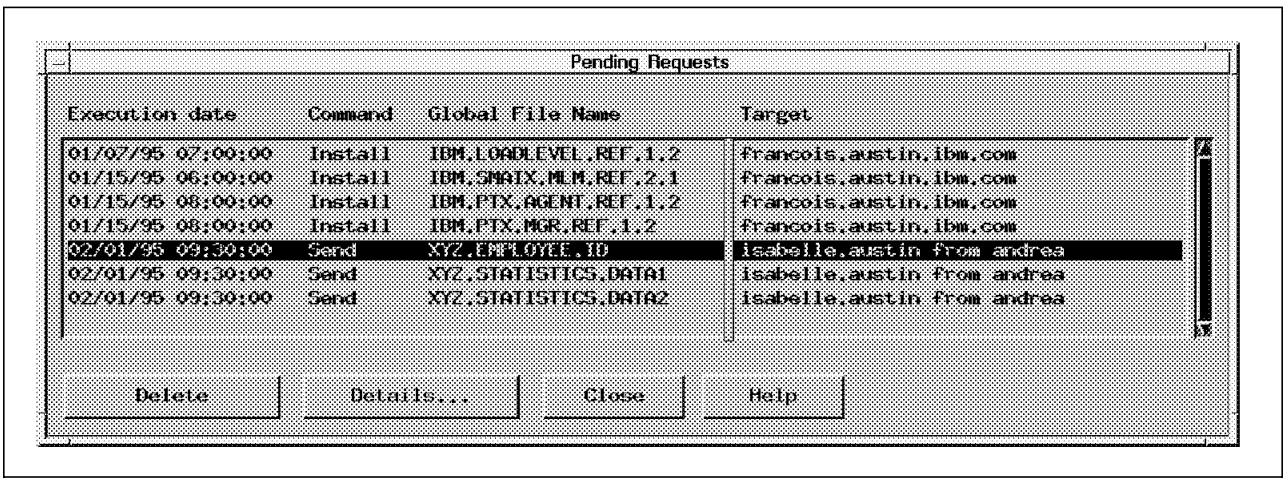

Figure 27. Pending Requests Panel

It is possible to visualize at any moment all scheduled operations in your distribution system, this per target and per change file, by activating the View pending requests menu on NetView DM/6000 GUI main panel (see Figure 27).

3.5.5 Performing Distribution Tasks

Apart from installing change files, NetView DM/6000 is also used to perform distribution tasks, that is, send or retrieve files between targets connected by TCP/IP and/or SNA networks. Like for change files, you need to catalog your data file in the Catalog window, by giving a global name, the file name under which it is stored on the file system of the workstation, and an object type from the following list:

- Flat data
- Software
- Microcode
- Procedure
- Reldata
- Dump
- Configfile
- Trace
- Errlog

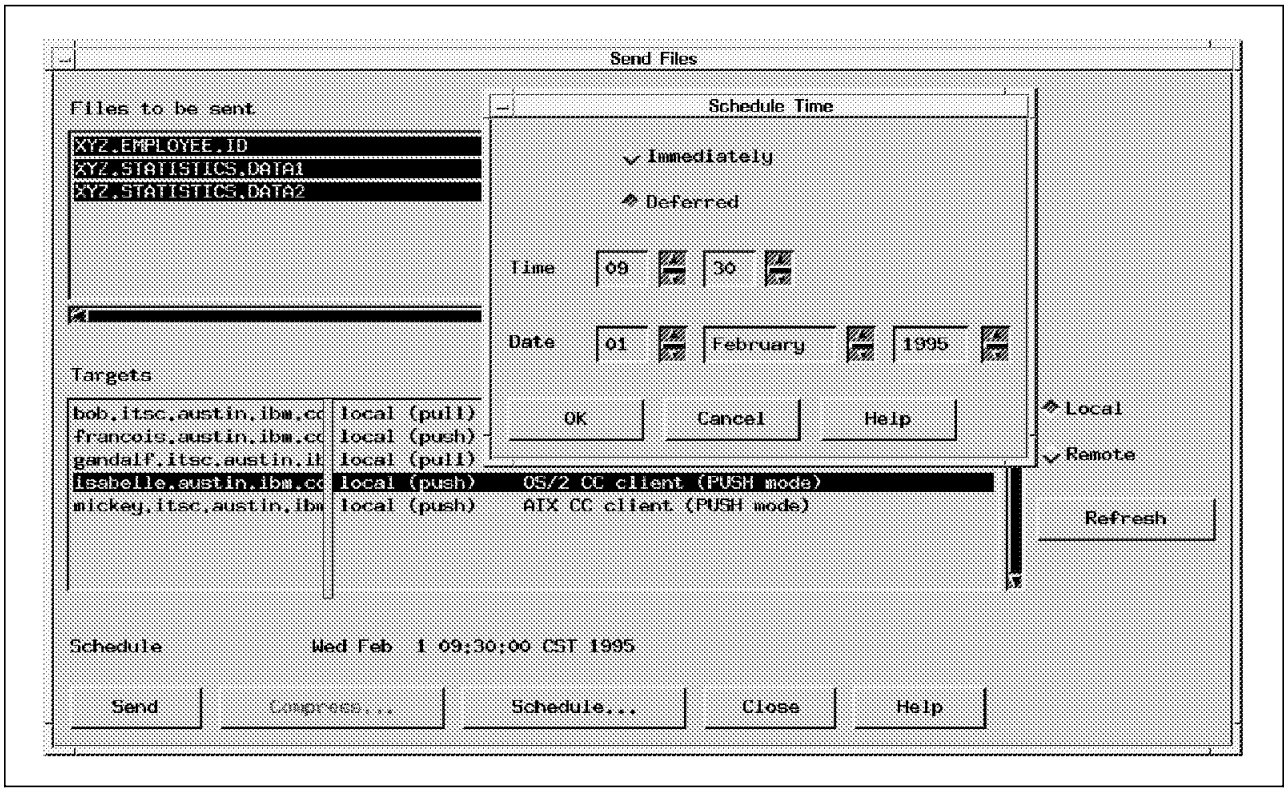

Figure 28. Sending Data Files

You also have to indicate whether the file consists of character or binary data, and which Codepage is used to know how the characters should be interpreted.

Figure 28 gives you the panel you have to customize to send data files over your network.

3.5.6 Other Operations

Every change control or distribution operation is accessible for a target on its Target History panel in the targets window application. Choosing a particular change file in the panel, you can install (immediatly or at a scheduled time), remove, accept or uninstall it on the target.

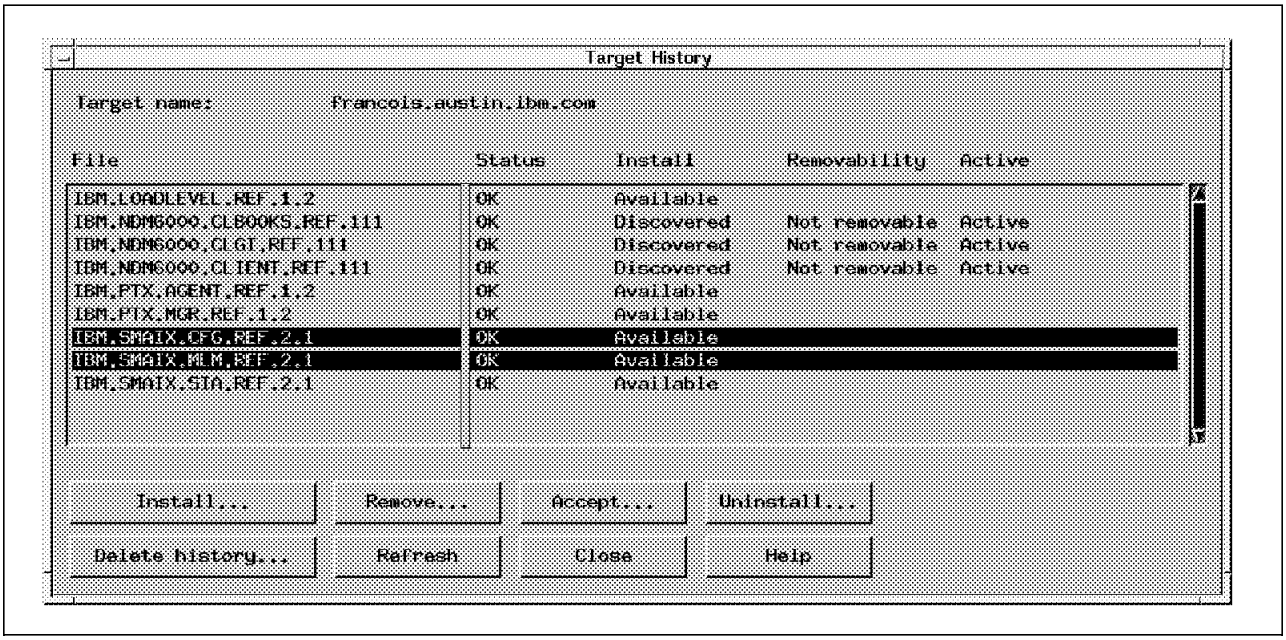

Figure 29. Viewing a Target History

Figure 29 shows an example of a target history.

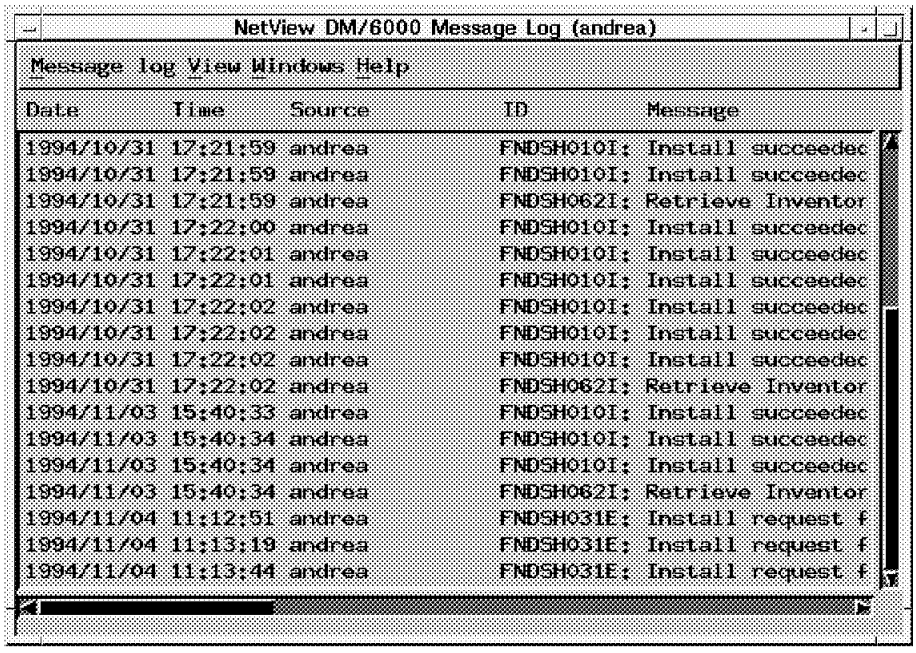

Figure 30. NetView DM/6000 Log Window

AIX NetView DM/6000 keeps a log of all the operations done. This log could be browsed using the graphical interface as shown in Figure 30.

There are many other AIX NetView DM/6000 features not covered in this document. See the *AIX NetView DM/6000 User's Guide* for more information.

Chapter 4. Monitoring Applications

A very common need of a distributed organization is the ability to check one or several applications from one machine. For instance, from a manager machine, you would like to know how the applications are running on three or four critical servers.

A very simple customization of Systems Monitor for AIX allows you to monitor many systems. This product uses the SNMP protocol.

4.1 Simple Network Management Protocol - SNMP

A very popular protocol to manage a network is called Simple Network Management Protocol, SNMP. It is a simple and very useful protocol, because it is the common denominator for many managed systems. This protocol is like a language with verbs and a vocabulary, used for communication between a managed system called an *Agent* and a management station called a *Manager*.

There are just five verbs. Three are requests that can be sent from the manager:

get which retrieves the value of a particular variable and returns the answer to the requester (with response).

set which updates a variable on the managed system (and change the system value that it represents); set is not available for all variables.

get-next Which retrieves the *next-variable*. We don't know how many variables or instances we have for a specific agent. This powerful command allows to scan the variables.

The other two verbs are used by the agent:

response which answers the manager after receiving a *get, get-next or set*.

trap which sends an alarm message to the manager, describing an extraordinary event.

All these verbs use a special vocabulary, which is called *Management Information Base (MIB)*. This is the standard set of object definitions that all SNMP agents are required to support. The popular MIB-II contains 171 object definitions. The value of the object instances within the MIB are maintained by many different system functions (kernel, subsystem and so on...).

The internet standard MIB-II is divided into groups:

1 System Group Describing the environment and capabilities of the managed system.

2 Interface Group Details of the physical network interfaces.

3 Address Translation Group mapping between physical and addressing schemes (obsolete).

4 IP Group Details about activities and capabilities of the internet protocol on the managed system.

5 ICMP Group Statistics about Internet Control Message Protocol.

© Copyright IBM Corp. 1993,1994

6 TCP Group	Statistics and capabilities of the Transmission Control Protocol.
7 UDP Group	Statistics and capabilities of the User Datagram Protocol.
8 EGP Group	Statistics and configuration of the External Gateway Protocol.
10 Transmission Group	Definition of a new specific type of interface (tokenring, loopback and so on...).
11 SNMP Group	Definition of status and process about SNMP objects.

The existing 171 objects are not enough to define all the situations necessary to manage a network; especially if a specific hardware or software is used to route or connect parts of the network. Therefore, it is possible to extend the MIB definition for an enterprise to define some specific variables concerning a special hardware, a software or a specific management strategy.

To use an extended MIB on the a RISC System/6000, a subagent or a proxy-agent has to be set up. Systems Monitor for AIX is a typical proxy agent working in connection with NetView for AIX.

4.2 Systems Monitor for AIX

An SNMP (Simple Network Management Protocol) subagent or proxy-agent is a program to transfer the non-SNMP objects in SNMP variables, or to extend the definition of your standard MIB-II (Management Information Base). When you want to add some variables to the MIB-II, you add a new branch in the MIB tree normally called `private enterprises` MIB. Each time you use a `private entreprises` MIB on a RISC System/6000, you need to use an SNMP subagent, because the only MIB known by the SNMP agent (snmpd) is the MIB-II. To manipulate the new SNMP objects and talk to the SNMP agent about them, you have a program called a subagent or proxy-agent. This program used the SMUX (SNMP Multiplexing) protocol to keep in touch with the SNMP agent (snmpd). This protocol is defined in RFC1227 (Request For Comment). You can see in the Figure 31 on page 39 the relation between subagent and agent. The concept of subagent allows an SNMP manager to manage some non-SNMP objects.

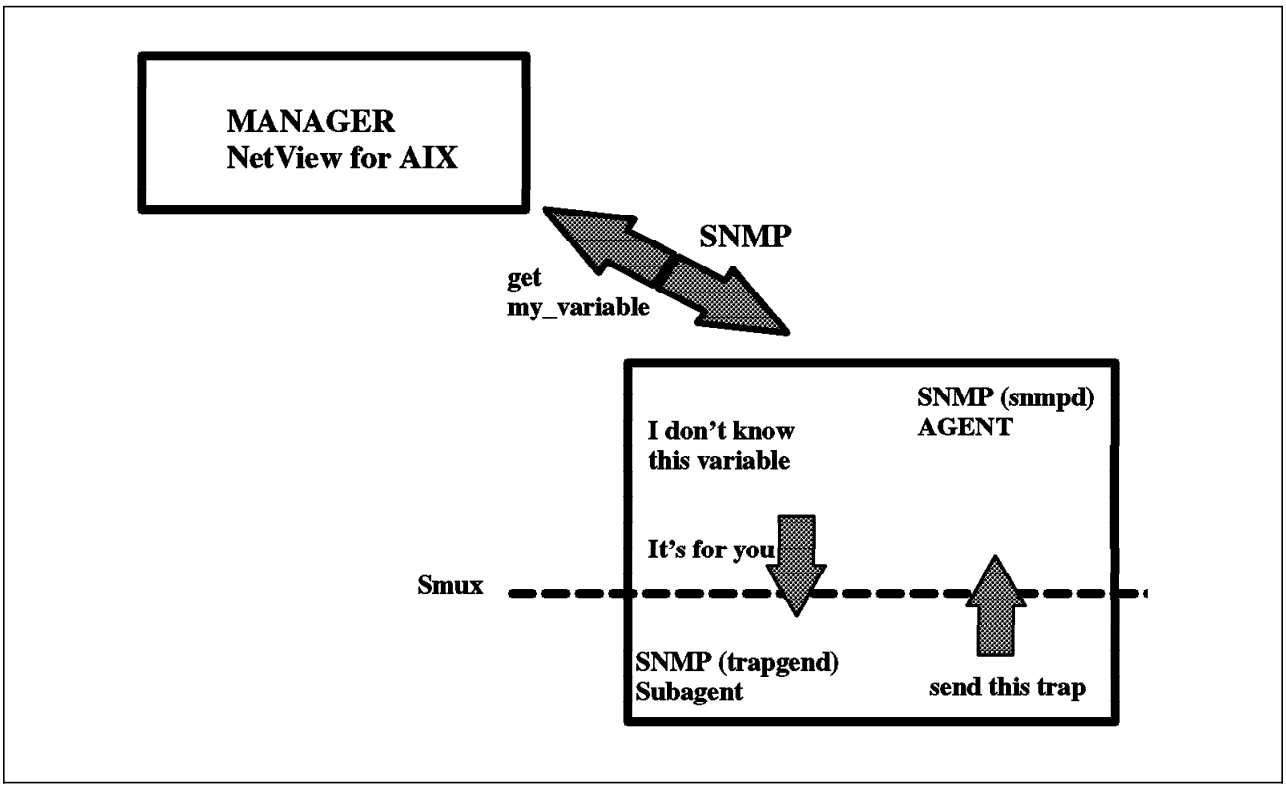

Figure 31. Relation Agent/Subagent

If you try to get a special variable from the manager not included in the MIB-II, the snmpd program doesn't know how to answer your request; but it does know there is a subagent running in connection with it, and will give it the request (the snmpd.conf and snmpd.peers configuration files exist for this reason). On the other hand, if a subagent wants to send a trap, it can be sent to the agent which sends it to a manager. At the end the subagent can process some action in connection with the SNMP agent. For instance the *trapgend* program of NetView for AIX can convert all the errors logged in your error log in SNMP traps understood by the SNMP manager as NetView, as explained in Section 12.1, "Using AIX NetView/6000 as Problem Monitor" on page 161.

Systems Monitor is a SNMP subagent running today on the AIX, Sun Solaris**, HP-UX and UNIX for NCR** platforms.

4.2.1 Functions

The two major functions of Systems Monitor for AIX, as shown in Figure 32 are:

- Private enterprises ibmprod Systems Monitor MIB
- System management through Systems Monitor MIB

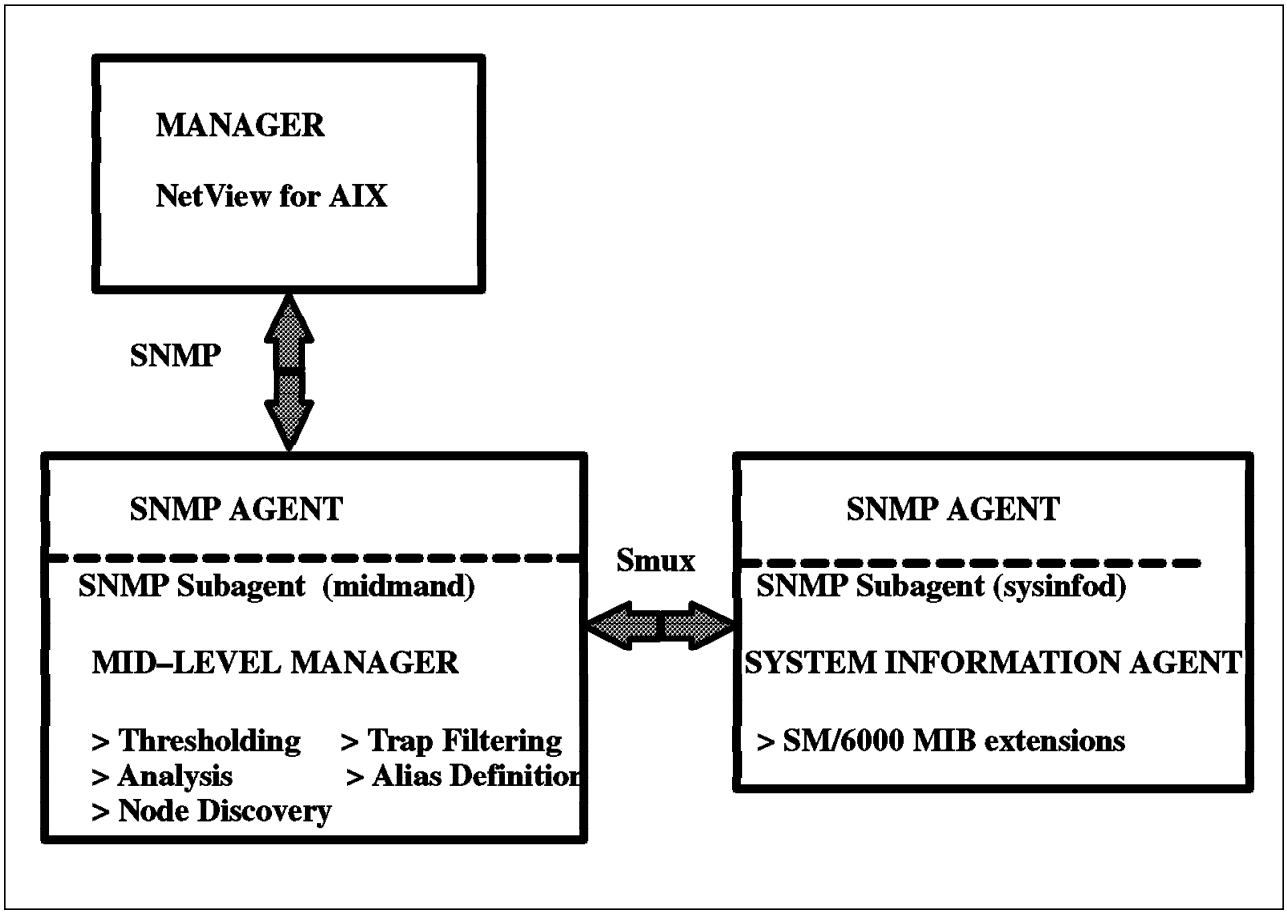

Figure 32. Functions of Systems Monitor for AIX

The first function (*System Information Agent* or *SIA*) is just an additional enterprise-specific MIB identified by the Object-ID: 1.3.6.1.4.1.2.6.12 to have more than 600 SNMP variables. This MIB extension is to know in detail the specifics of AIX V3.2: as program, system and network information. Systems Monitor installs some menus on your NetView for AIX to help you interrogate these variables easily.

The second function is a powerful one which delegates tasks from the SNMP manager to the agent using Systems Monitor. We can call the agent with the subagent Systems Monitor a *Mid-Level Manager*, because it can manage and act alone without interference from an SNMP manager. Its management allows:

- Create any command in the SNMP variables, which will be executed when you start a get or a set on this variable
- Collect the SNMP variables of local or remote nodes
- Create a threshold on an SNMP collect, which will activate a program or a trap
- Create some analysis definition about the SNMP collect
- Filter the traps and manage their destination
- Create the alias defining one or several nodes
- Discover and poll nodes belonging to its subnetwork (same subnetmask)

To help you in the definition of your management tasks, Systems Monitor has a user interface as a component working with NetView for AIX.

4.2.2 End User Interface

You can start the Systems Monitor End User Interface on your manager station, from the `smconfig` command on the command line, through smit or through NetView. This interface allows you to define your MIB variables, which contain the management definitions. After you set and test those variables, you can save them as a configuration file. It will be used by the Systems Monitor daemons (sysinfod and midmand) for the initialization of the MIB variables, each time the `sysinfod` and `midmand` programs restart.

After you start the interface you display the window shown in Figure 33. You can specify on which node you want to set your MIB variables, which contains the management definitions.

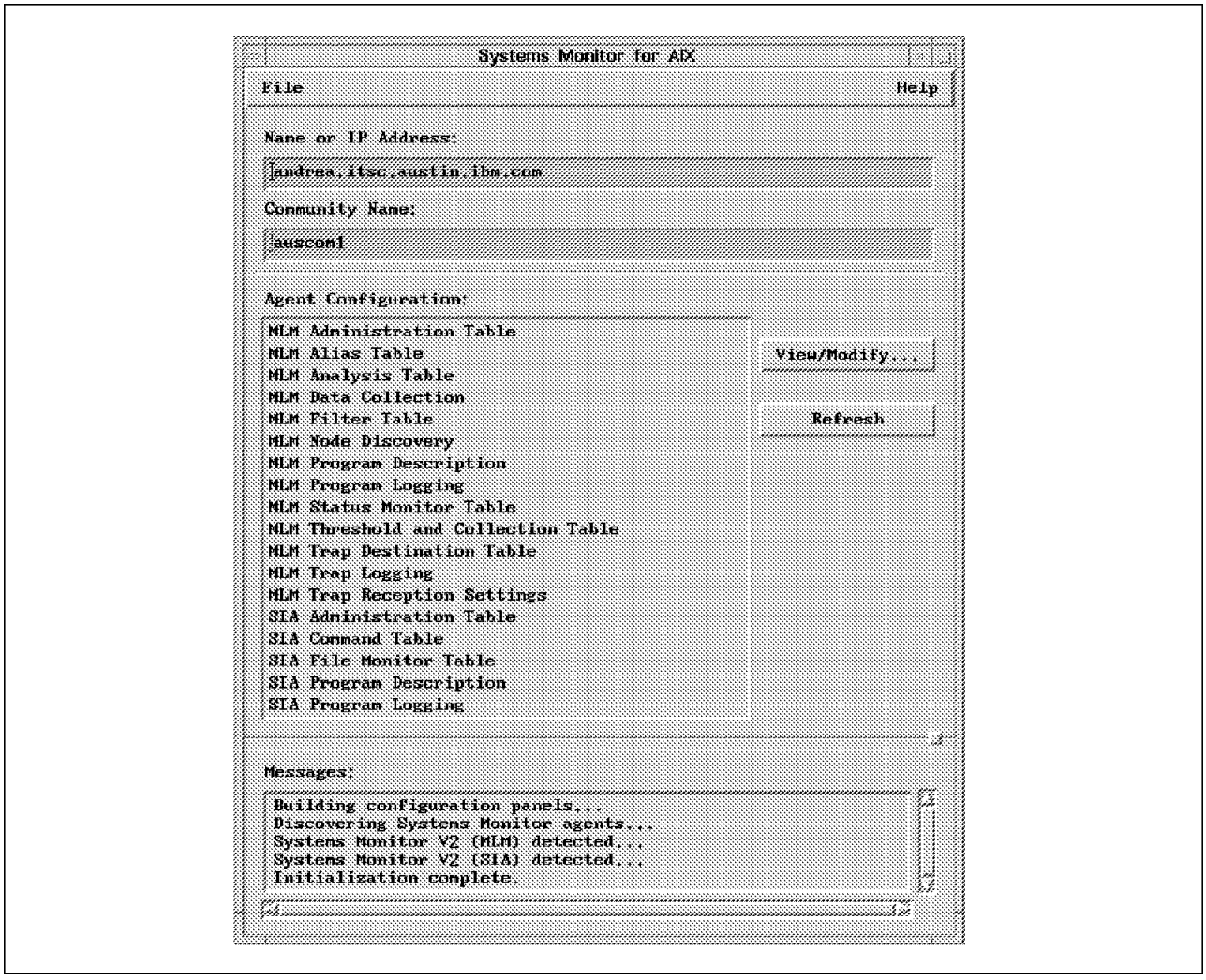

Figure 33. Starting with smconfig

The MIB variables are organized in a table, which means you have a table to define the different actions for each task (data collecting and thresholding, filtering, defining new MIB variables, and so on).

4.2.3 MIB Tables

A table is like an array where:

- Each row represents a separate and unique entry
- Each column represents a different MIB variable

You know a MIB variable can be an object with one or several instances. In Figure 34 you can see how a table is a set of variables with multiple instances.

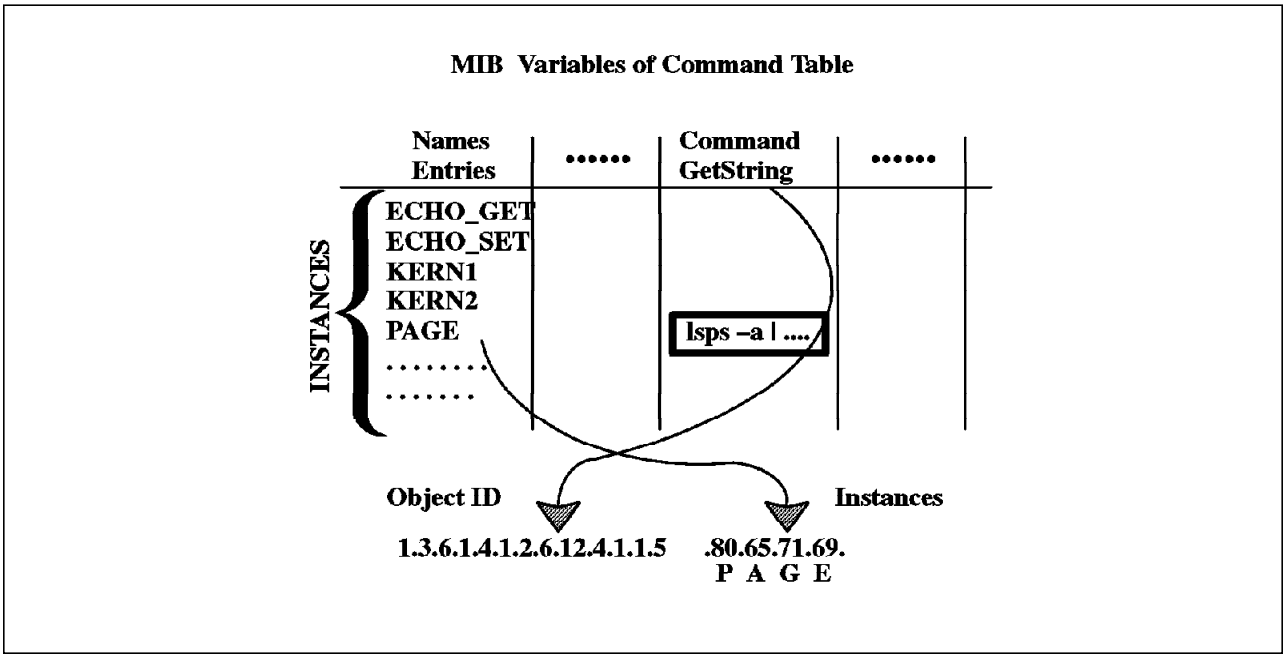

Figure 34. MIB Table

Each entry is an instance of the different variables of a specific table (Command, Threshold, Alias and so on). An entry is a character string and an instance is a set of figures separated by a dot. The conversion from the character to the figure is just the ASCII conversion with the following standard table:

```
032 sp| 044 , | 056 8 | 068 D | 080 P | 092 \ | 104 h | 116 t
033 ! | 045 - | 057 9 | 069 E | 081 Q | 093 ] | 105 i | 117 u
034 ″ | 046 . | 058 : | 070 F | 082 R | 094 ^ | 106 j | 118 v
035 # | 047 / | 059 ; | 071 G | 083 S | 095 _ | 107 k | 119 w
036 $ | 048 0 | 060 < | 072 H | 084 T | 096 ` | 108 l | 120 x
037 % | 049 1 | 061 = | 073 I | 085 U | 097 a | 109 m | 121 y
038 & | 050 2 | 062 > | 074 J | 086 V | 098 b | 110 n | 122 z
039 ′ | 051 3 | 063 ? | 075 K | 087 W | 099 c | 111 o | 123 {
040 ( | 052 4 | 064 @ | 076 L | 088 X | 100 d | 112 p | 124 |
041 ) | 053 5 | 065 A | 077 M | 089 Y | 101 e | 113 q | 125 }
042 * | 054 6 | 066 B | 078 N | 090 Z | 102 f | 114 r | 126 ~
043 + | 055 7 | 067 C | 079 O | 091 [ | 103 g | 115 s | 127 del

 So to convert PAGE P = 80, A = 65, G = 71, E = 69 and
       PAGE = 80.65.71.69
```

4.2.4 Command Table

When you want to set up the same instance (PAGE) in your smconfig session, you click on the Command Table to see the window shown in Figure 35. With this window you can add and delete a command or modify your PAGE command.

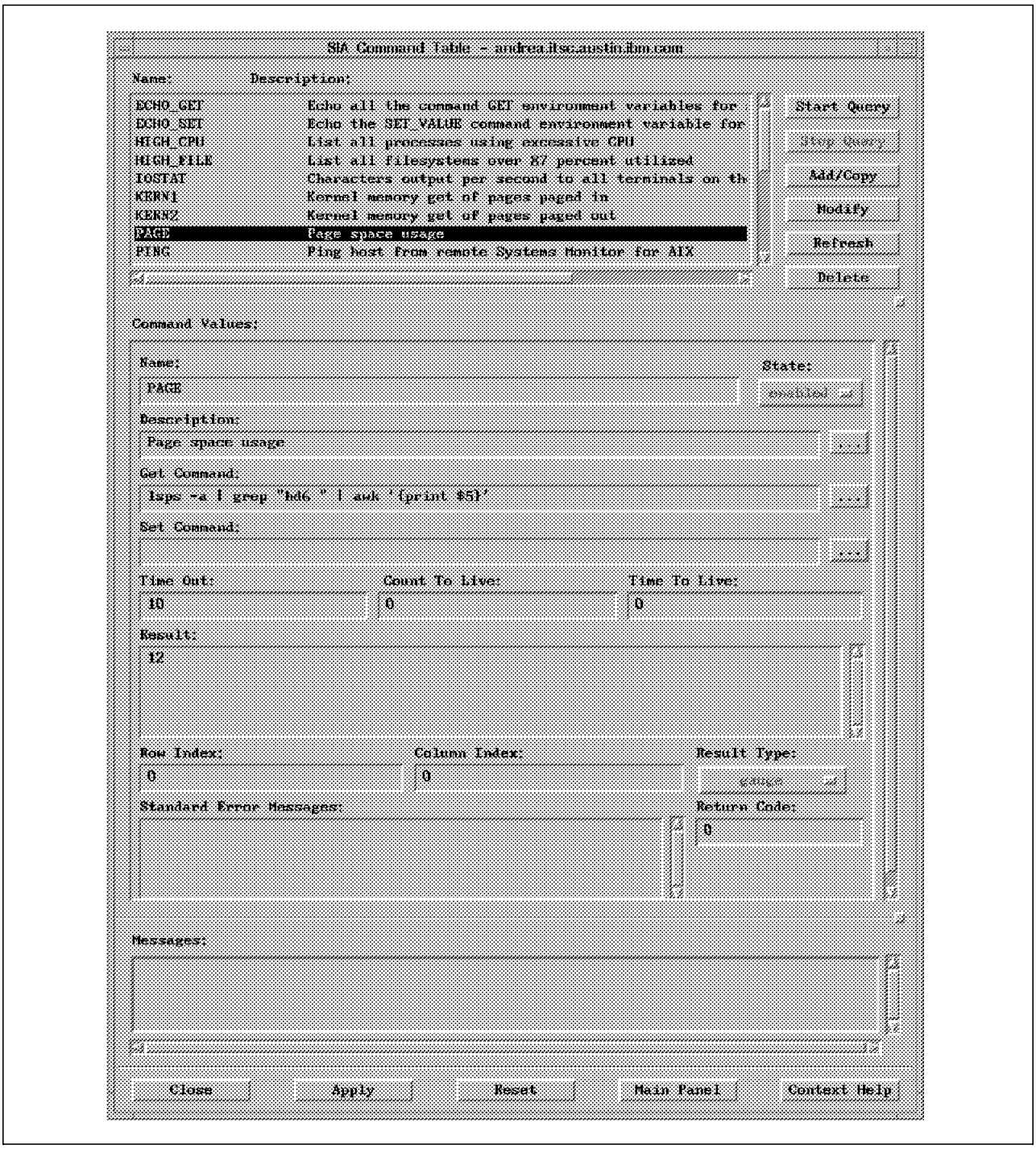

Figure 35. Command Table Modification

Each field of this panel is the PAGE instance (80.65.71.69) of all the variables concerning the command entry, with the Object ID 1.3.6.1.4.1.2.6.12.4.1.1, that you can see on Figure 36 on page 45.

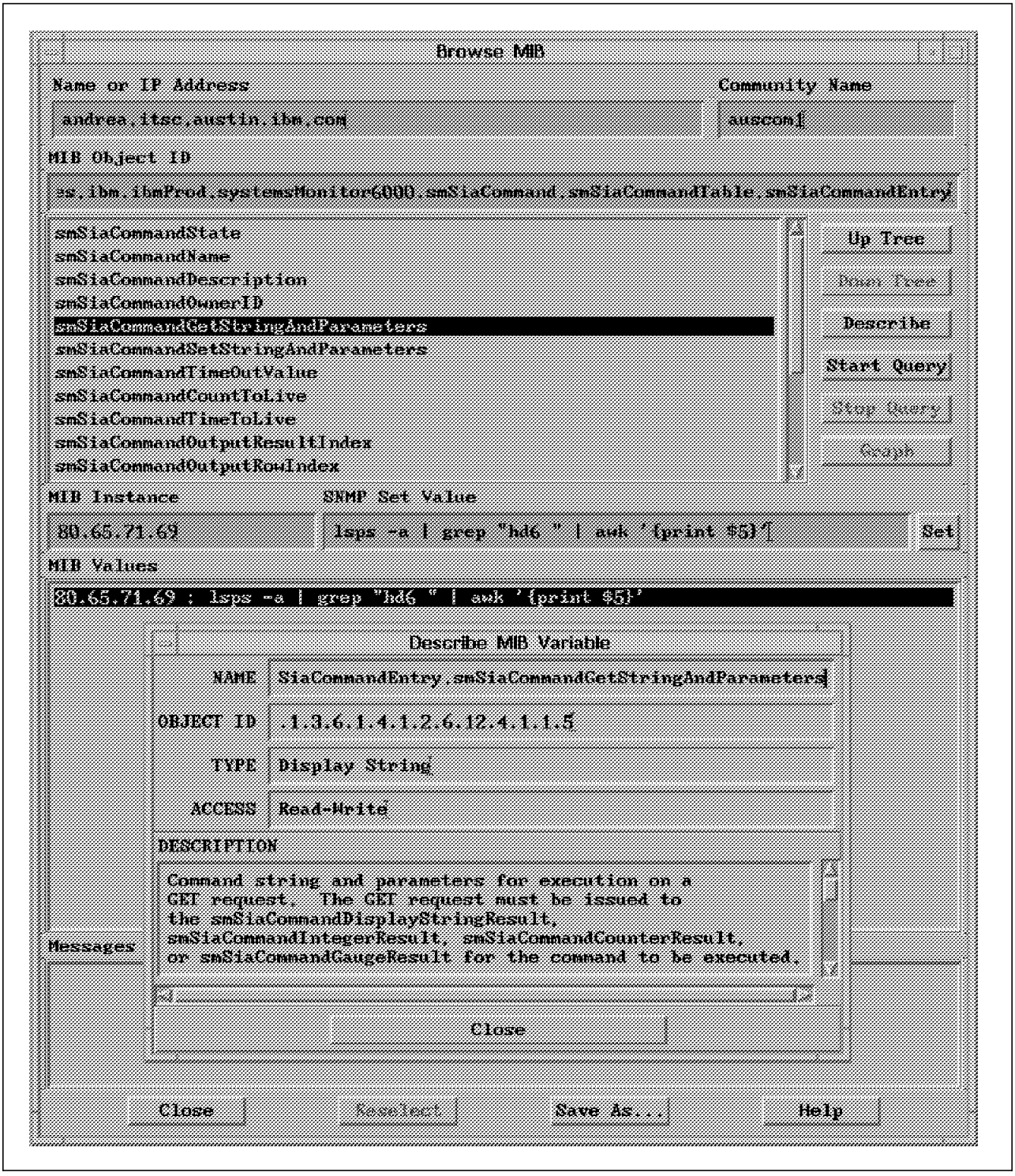

Figure 36. MIB Variable and MIB Instance

Each time you get (SNMP get) the PAGE instance of the smSiaCommandGetStringAndParameters you run the command:

```
lpsps -a | grep "hd6 " | awk '{ print $5 }'
```

And the command's result is in the PAGE instance of smSiaCommandGaugeResult like the Result Type setting shown in Figure 35 on page 44. You can modify your command, and test it with the Query option. You get the result in the result sub-window.

It is an easy way to set your table. After the testing you can save your definitions in a configuration file, which will be used for the next initialization of sysmond.

You can use the Command Table to watch your application with special monitoring programs. However, the easiest way to watch an application is to use the threshold table.

4.2.5 Threshold Table

The threshold table is mainly used to collect the SNMP variables. You can use the threshold facility, if you can define a specific acting value from a variable. The action can call your program or send an alarm or a trap. Indeed you have a Rearm action when you return to the normal situation.

There is a special variable in Systems Monitor for AIX called smSiaSystemProcessPID, which contains all the processes and their PID running on the machine. You can see this variable on the Browse MIB window in Figure 37 on page 47. Each instance of this variable is the name of the process and its PID such as:

```
Instance : 116.114.97.112.103.101.110.100.6796
Meaning  : t   r   a  p   g   e   n   d   PID
```

This give you a list of your current processes with their PID in the different instances of this variable. You can see an instance in this variable such as:

```
116.114.97.112.103.101.110.100.any_figure
```

This means the process is running on your machine. It is through a MIB variable the result of the ps -e command giving only the name and the PID of your processes.

To set up the SNMP collect, you call under the smconfig facility the threshold table window, as shown in Figure 38 on page 48. In this sample configuration of your threshold table you can select the Monitor_Process entry, to enable this entry with a modification of its state. You can modify the process name easily. Change trapgend with the process name you want to look at. You can check the MIB variable you want to collect: if you press the Select button, the Browse Window will be displayed as shown in Figure 37 on page 47.

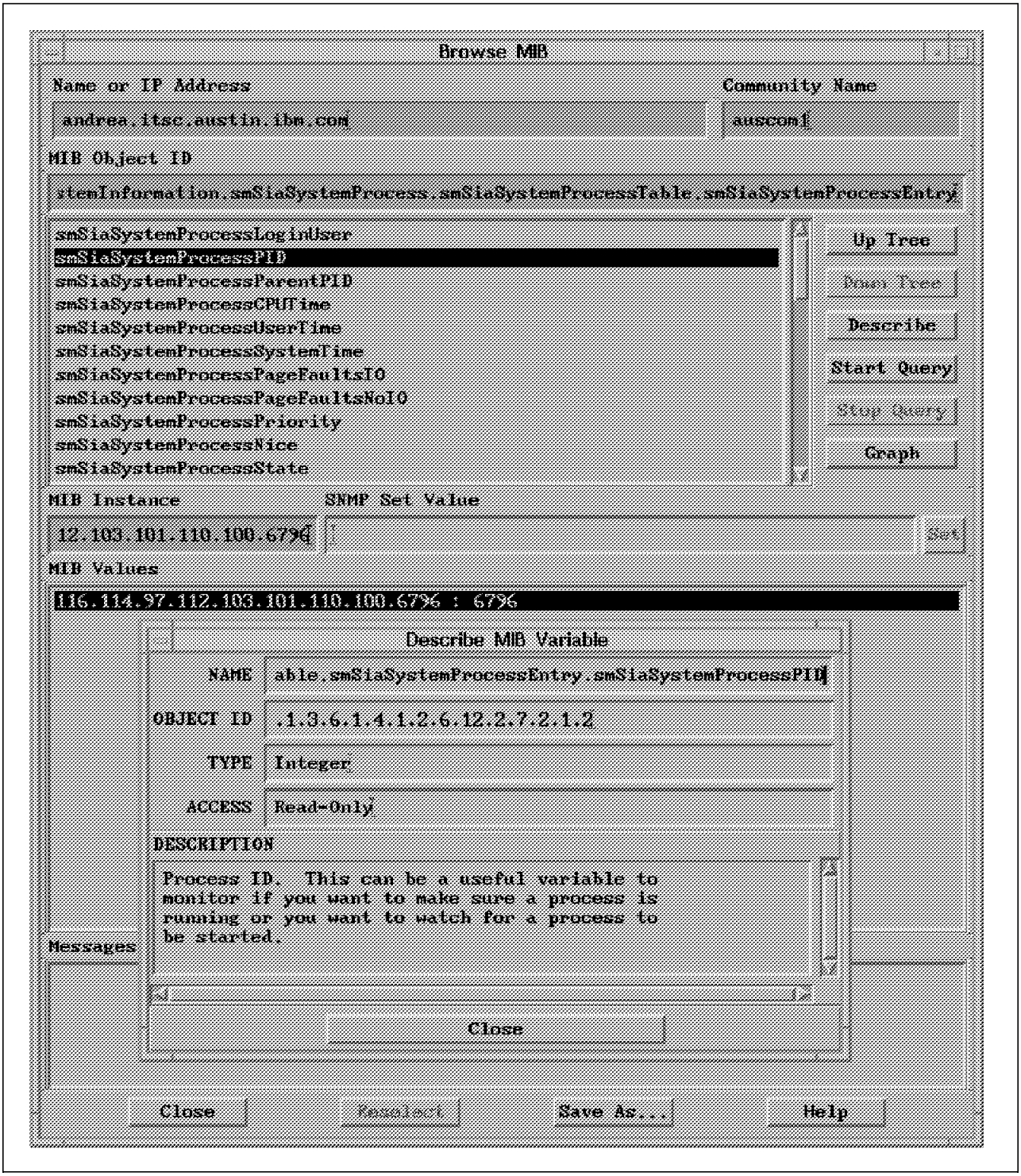

Figure 37. Browse MIB Window

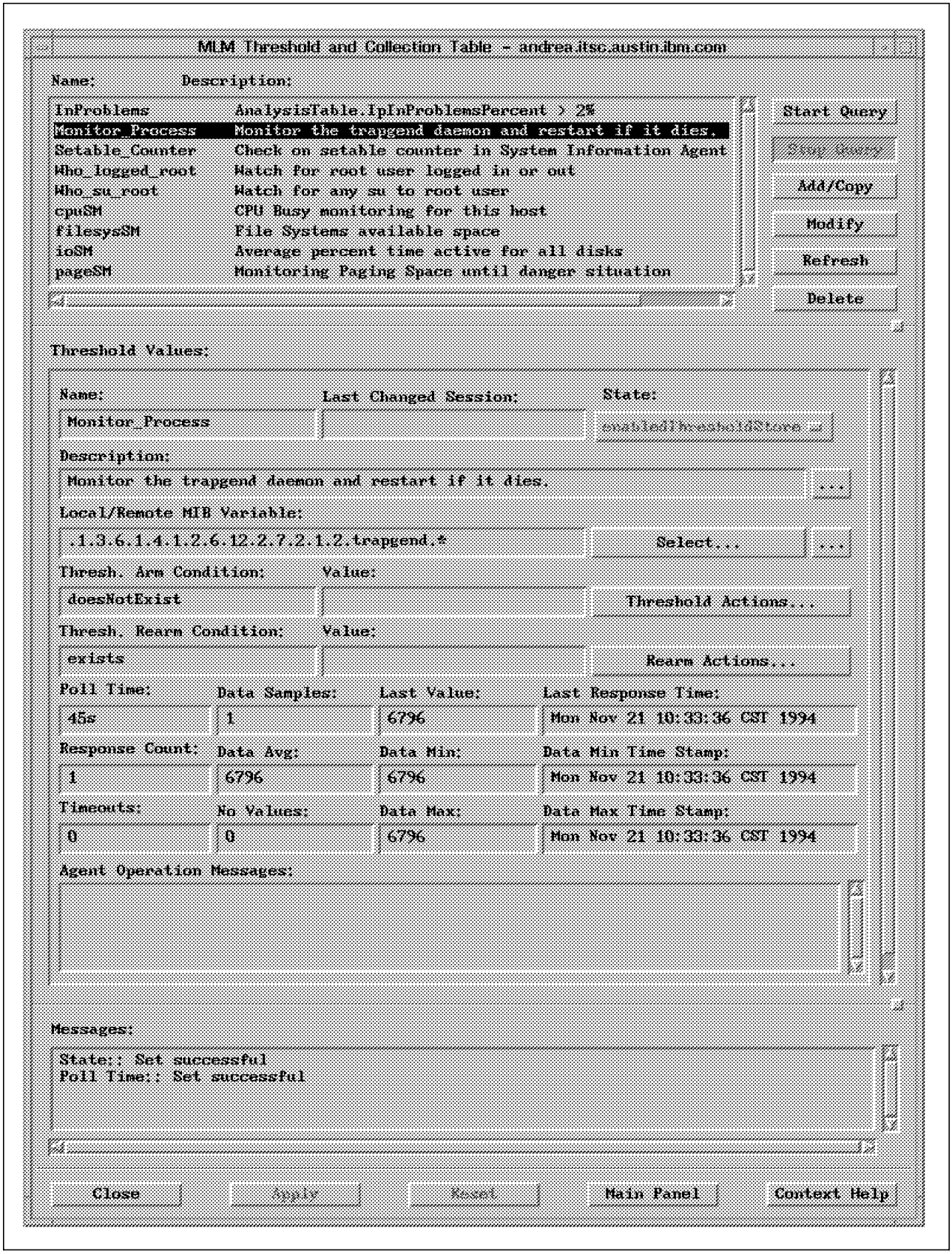

Figure 38. Threshold Table Window

The asterisk at the end of your MIB variable means you are looking for any PID. The State setting is enabledThresholdStore. This value is stored every 45 seconds (Poll Time) in the /usr/adm/smv2/collect/midmand.col file and your threshold is the existence of this instance (doesNotExist / exists). You can use the Query option to see the collection of your data increase in the Response Count field, the last value in Last Value field and the different value in the Data Min, Data Max, Data Average and Data Samples.

You check the existence of your trapgend process (in this sample) setting doesNotExist in the Thresh Condition field. You can see your programming action pushing the Threshold Actions button and the Threshold Actions window displayed in Figure 39. You can specify a trap to send with its description and a command or program to execute. In this sample the command is just the restarting of the trapgend program itself.

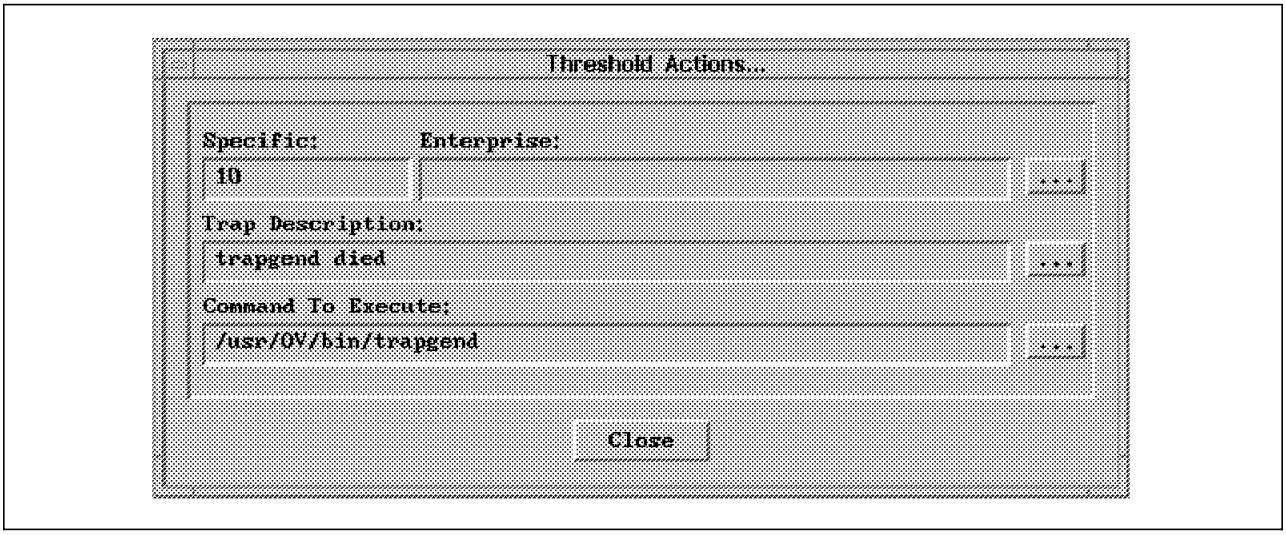

Figure 39. Threshold Actions

In the same way you can define a Rearm Condition (exists) with a Rearm Actions sending a trap or executing a program.

4.3 Conclusion

To look at any application you can follow this model, which sends a trap and executes a program each time it fails (disappears from the process current list) or starts (appears in the process current list). You can easily start a program with the Command Table to execute some checking tasks, and even share some memory with sharing memory techniques. So it has five minutes of customization to set up an alarm or a starting program if a process fails. The most important task is to create a recovery program for your application. In the example this recovery task is just restarting trapgend. This program depends on your application needs.

The Systems Monitor for AIX allows you, with the Command Table principle, to create some new variables, which are really new instances. You can interrogate these new variables in two ways, one to put some value in the result instance, the other to read this result value. With this part of the product Systems Monitor for AIX you don't need to write your own subagent or proxy-agent, you have an *Extensible Proxy-Agent*.

Chapter 5. Application Data Backup Strategies

Since there always could be an error, within your hardware, your software or from yourself, it is important to have recent backups available. Application data is the most critical data on your machine, and therefore you have to think about a reliable backup strategy. This chapter will show you different aspects, which you should take care of, when you are planing your backup strategy.

5.1 Different Types of Data

Before looking for software to make backups and restores, we must describe what kinds of data we are talking about. We can classify the data from various parameters:

- How important is this data?
- How often is the data modified?
- How big is the modified data?
- How is the data spread on the local machine?
- What is the technical format of the data (filesystems, raw devices)?
- What impact is there, when the data must be recreated?

There are also some operational factors like:

- Which media should be used for the backup?
- Do we have to back up data from one or many machines throughout the network?
- If we are networked, should the data backup policy be centralized or decentralized?
- Must the backup operations be *human operator free*?
- Is the data confidential?
- How much time can it take to restore the data?

5.2 Different Ways of Performing a Backup

In a UNIX environment, there are many ways to do a backup. Mainly, it is making a copy of data, so you can be sure that you are able to restore it in case of a failure of the original data. Here is a possible classification of backup techniques:

- Lazy backup: this is the most common. This is the kind of backup you make from time to time, just doing a `tar`, a `backup` or a `dd` command of a user filesystem onto a media. Or sometimes you make a copy of some directories to another machine through an NFS connection.
- Organized single machine backup. This is typically a tape backup with a clear strategy defined. For instance, you do a full backup every Friday night, and an incremental backup every other night. You make at least two copies, keep the three last weeks and use new tapes every ten weeks.

© Copyright IBM Corp. 1993,1994

- Organized and certified single machine backup. This is the same as the organized single machine backup, but you have tested all the possible recovery procedures.
- Distributed lazy backup. You are doing some backup through a network, using a network backup tool or something you have made yourself using NFS. Typically your tool will make a full or incremental backup of everything on a given machine.
- Distributed organized and certified backup. With this one you meet the same quality condition as in a single machine organized and certified backup, but with a network tool.

For your business critical data you should only use organized and certified backups, because you have to be sure, that you can recreate your data.

5.3 Backup Frequency

Usually, when you backup all the data on a given machine, the data is coming from different applications or different users. Every application or user has many files. The number of related files to a given application or a given user should be seen as a *data set*. To view these files as a data set makes it easier for you to develop your backup strategy. Keep in mind, it's just a model, because in UNIX there is no technical definition for a data set, you only have flat files. It is very important, that during the backup, all files which belong to a data set must be consistent. Otherwise your backup will be useless.

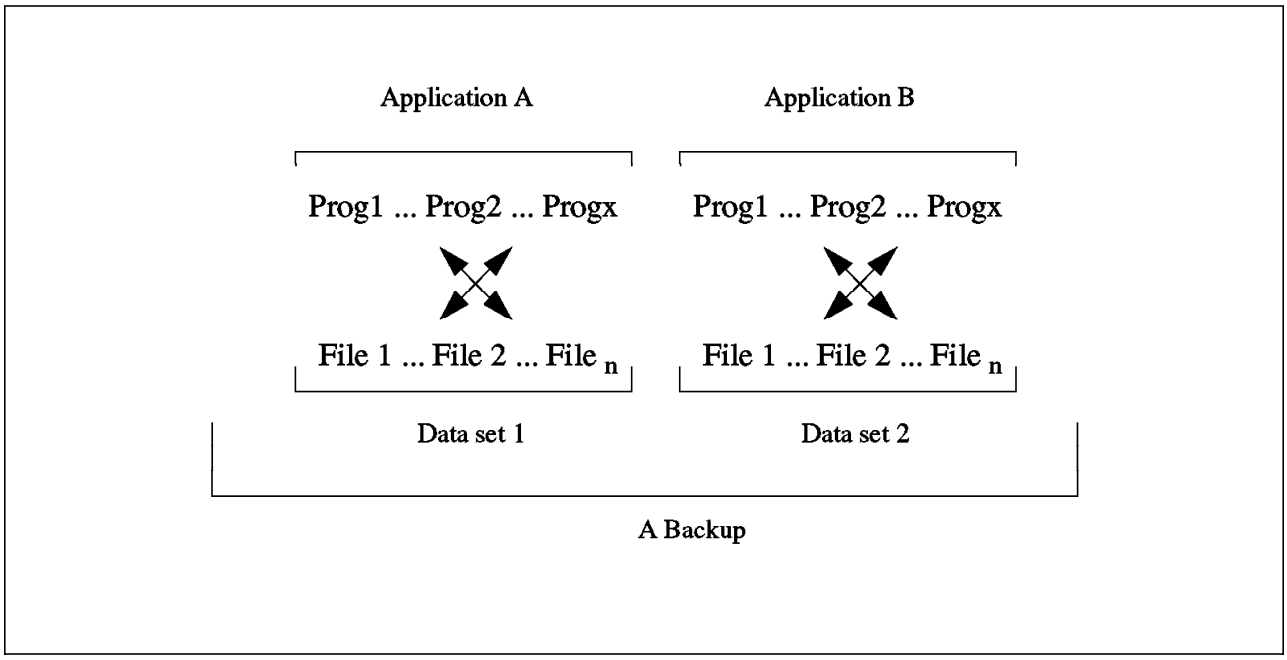

Figure 40. General Concepts about Data Sets

For instance you have a payroll application that uses a file named /u/payroll/employees/salaries and a file named /u/payroll/company/accounts, when a user updates the salary of an employee, both files are modified. If at the same moment a tape backup was running, it is probable that the copy on the tape will have the file /u/payroll/employees/salaries updated and the file /u/payroll/company/account not updated. This is an inconsistent data set, and your backup of these data is useless. The easiest way to get a consistent data

set is to stop any program activity for this data set during the backup time. Usually you stop all the applications during the night and then start your backup.

5.4 Special Case of Database Data

Most of the available backup solutions today are only able to save files. A file is a string of elementary data identified and organized by the operating system. A relational database product (like IBM AIX DB2/6000) has their own internal organization of the data which is not seen by the operating system. There is no way from outside the database to know if the data is at a given point in time consistent. The database products store application data into a multiple set of files or even in one or many *raw devices*. A raw device or logical volume is a reserved space in a volume group, without any filesystem structure. Database products have their own methods included, to manage such devices.

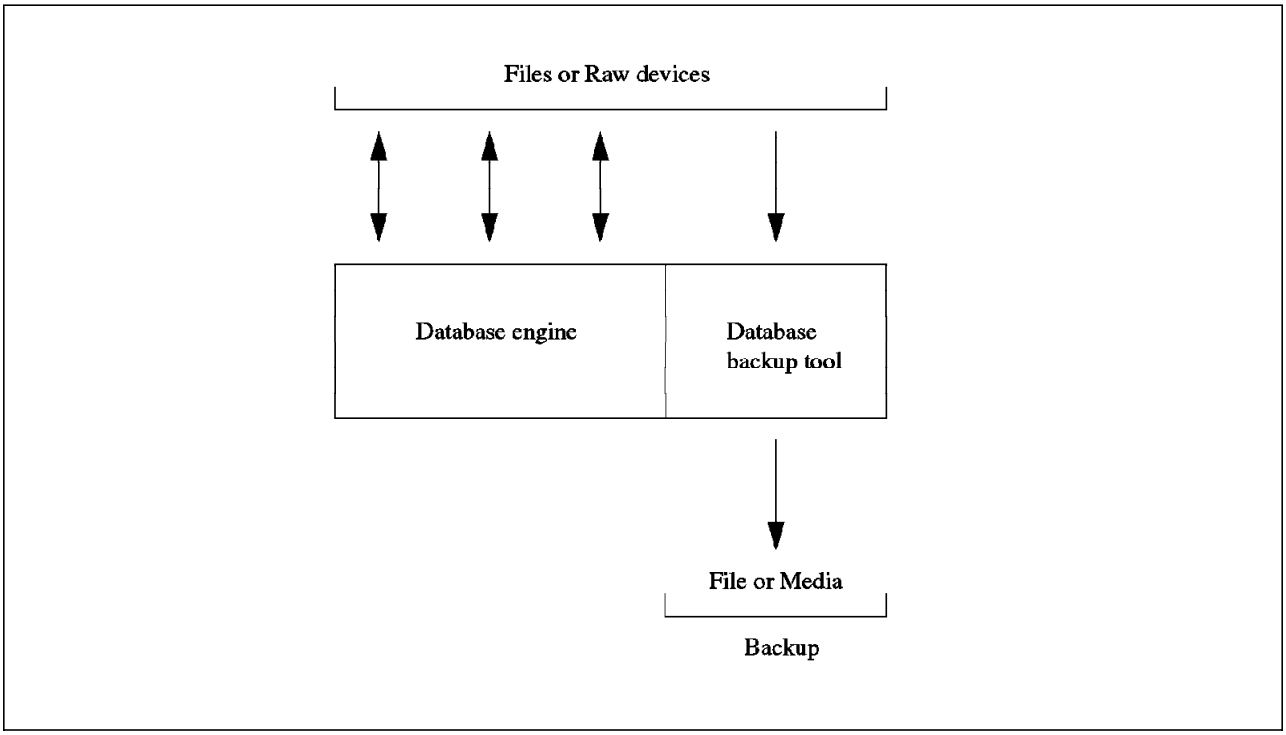

Figure 41. Backup of a Database Data

To have a safe backup, you *must* first use the database proprietary backup utility. This tool usually extracts a consistent image of the databases data into flat UNIX files. Then you can backup on tapes or through the network those flat files as usual. Sometime database products (like DB2/6000) proposes online backup tools, which allow you to make a backup of the database data without stopping the database itself, which means the application must not be stopped.

5.5 Available Software

During the creation of your backup strategy, you will come to the point, where you have to decide which software you want to use for your backup/restore. For a single machine, it will be enough to use the standard AIX commands, but the bigger your environment, the more you must invest into a backup tool. There are a lot of backup tools available, we just want to point out some of them.

5.5.1 Standard Commands

AIX standard commands are usually used on a single machine. In a networked environment with a lot of application and user data you want to setup an automated procedure under centralized control for the backup. If you want to do this with standard commands, you have to develop some shell scripts to handle your needs. These shell scripts are usually controlled by `crontab` entries to automate the backup process.

Available Commands are:

- `backup` (also for incremental backups)
- `tar`
- `pax`
- `cpio`
- `dd`

The main differences between these commands are the format on the media, the functionality (for instance, `tar` doesn't support incremental backup) and the relative performance of the backup/restore operations. Also keep in mind, when you are working in a heterogenous environment, that `backup` creates tapes readable only on AIX.

Detailed information about these commands is available in the Info Explorer* database.

5.5.2 Available Tools

As described above, to have an effective backup environment, you need to develop shell scripts. Instead of this, you also can use one of the many available backup tools. There are tools for every kind of environment (single machine, networked environment) available and some of them are also designed for working in an heterogenous environment. These tools give you a lot more comfort for customization. Before you are looking for a tool, you should already have an idea about your backup strategy.

Depending on your environment and backup strategy you should check for the following:

- Which backup medias are supported?
- Which interfaces are available (ASCII and/or Motif based)?
- Will the integrated backup strategy fulfill your needs?
- What kind of errorhandling is supported?
- Is the restore in case of an damaged media supported?

There are some products from IBM available, which can be used as backup/restore solutions. In this chapter we will give you an overview of the Legato Networker, for other solutions see Chapter 11, “Storage Management” on page 149.

5.5.3 Legato NetWorker

Legato NetWorker is a software product that reliably protects files against loss across an entire network of systems. A client/server architecture is the key to NetWorkers ability to support network-wide backup and recover. The servers provide a backup and recover service: they receive files from clients, store the files on backup medias and retrieve them on demand. Clients can backup their files to the server, browse the online index to find lost files and recover them from the backup media.

A server is a system equipped with one or more backup devices. The servers responsibilities are to:

- Write files from many clients to backup volumes
- Maintain indexes of backup history of all the files backed up and on which backup medias they are stored
- Allow the clients to browse the file index to locate files to recover and fill the requests
- Initiate automatic network-wide backups according to a specific schedule
- Give notices about NetWorker events for monitoring your network backup activity

The clients are all the other systems on the network that use the backup and recover service provided by the server. They have software installed that gives them access to the server for backup, restore and browsing the online index.

Following are some of the Legato NetWorker features:

- Support for heterogeneous platforms (from PCs and NetWare** - based LAN servers to UNIX desktop and multigigabyte server)
- Centralized operation and administration
- An X Window System user interface
- Support for different media (magnetic tape drives including 4mm DAT, QIC, 8mm; optical disk drives; several jukeboxes)
- Ability to tailor backups
- Automatic spanning of backups across multiple backup volumes
- Parallel backup of multiple client systems to multiple devices

For a detailed list of supported systems and medias call your local IBM Support organization.

In Figure 42 on page 56, you can see a typical configuration of an enterprise TCP/IP network with a Legato NetWorker backup and restore server. All the AIX RISC System/6000s in the network have installed the client part of Legato NetWorker and can therefore use these services.

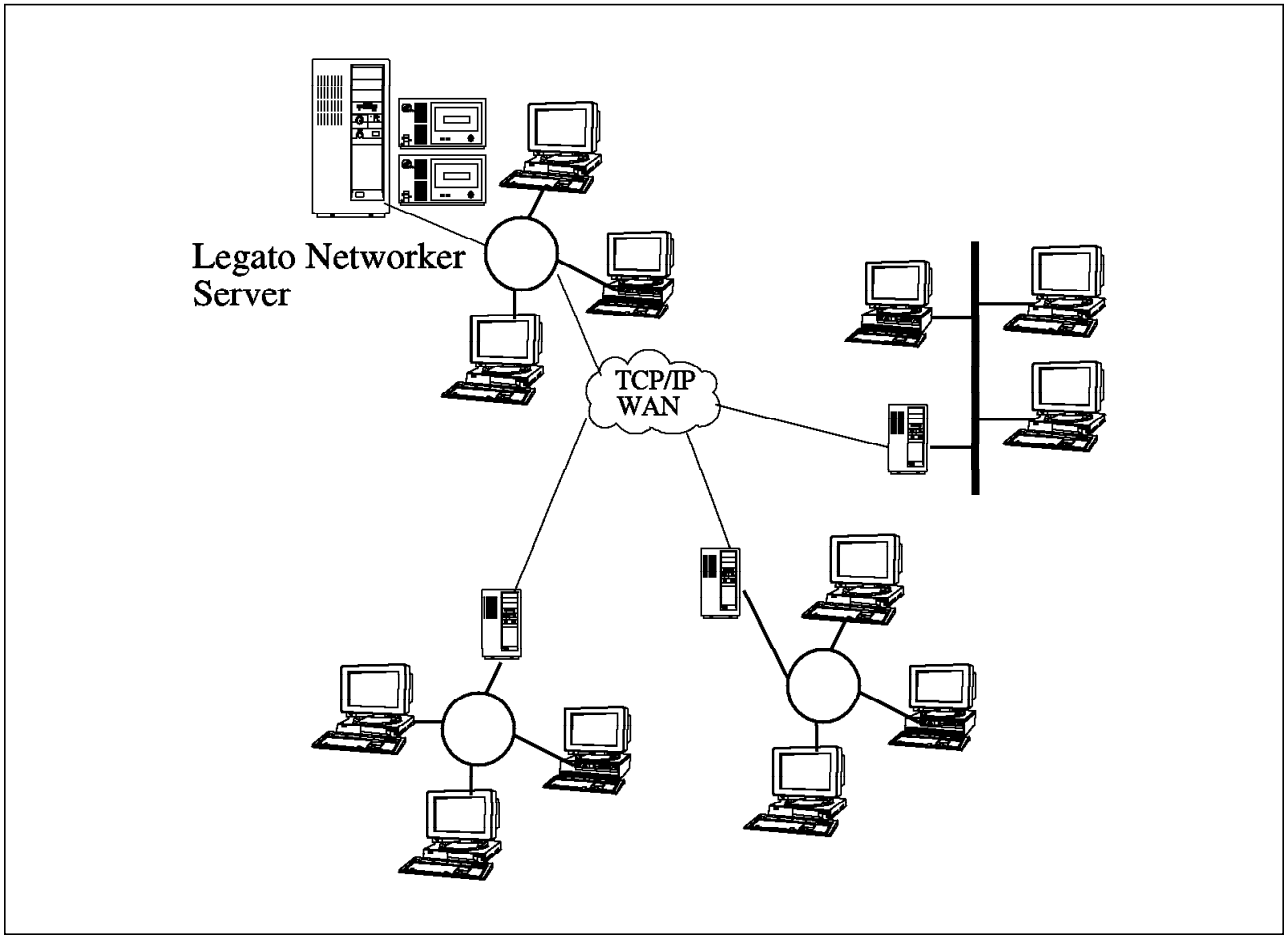

Figure 42. Using a Legato NetWorker Backup Server in a TCP/IP Enterprise Network

You will notice that some RISC System/6000s are on the same LAN as the server and some other are on different LANs interconnected through a WAN. In this case, you need to take care about the network bandwidth of your WAN connection, because the bandwith could not be enough to backup through the WAN.

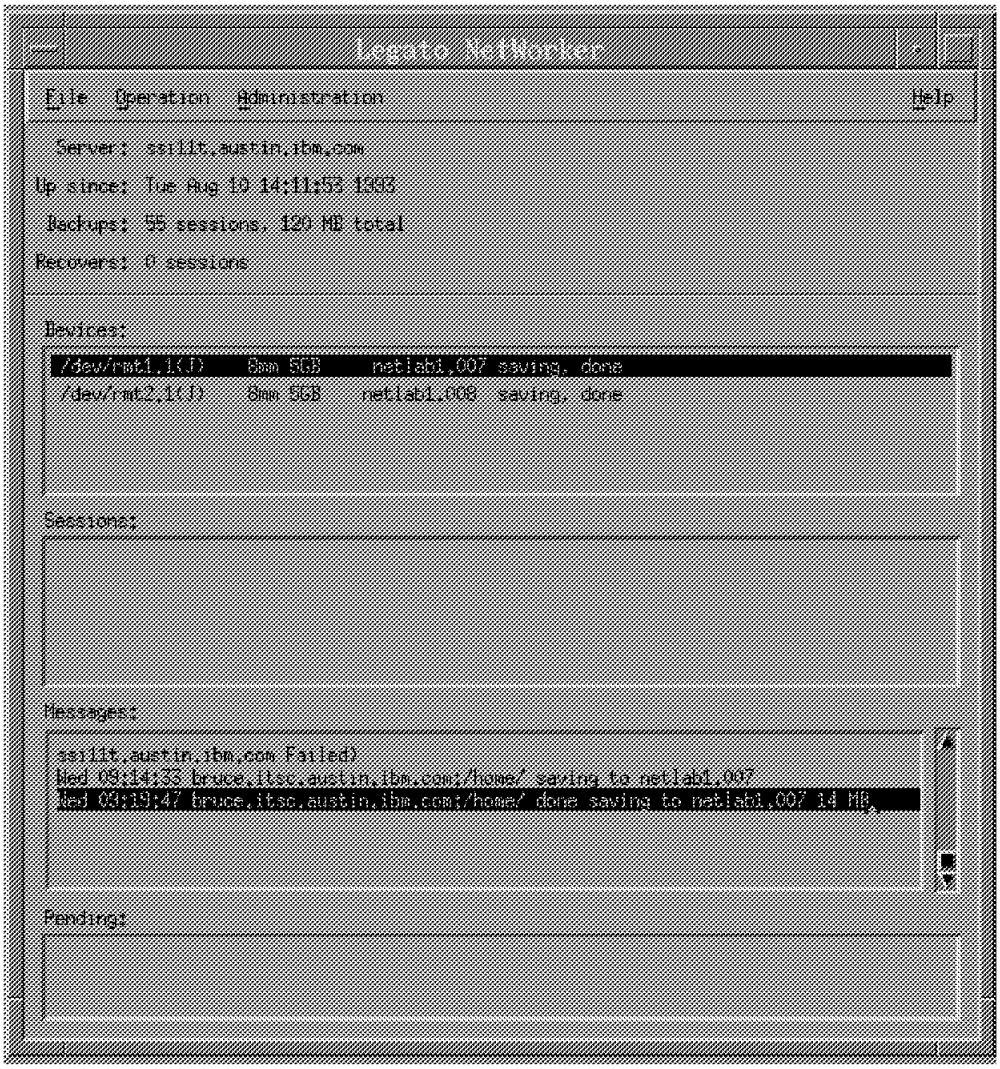

Figure 43. Legato NetWorker Main Window

The user interface of Legato NetWorker looks the same for a user and a administrator. For instance, in Figure 44 on page 58, a user can select what files or filesystems he wants to backup, while the screen of Figure 45 on page 59 shows, how an administrator can define the automatic backup policy for a particular user machine.

In fact, the user can decide to backup files at any time but the administrator can setup automatic backups that will ensure that minimum backups are always made.

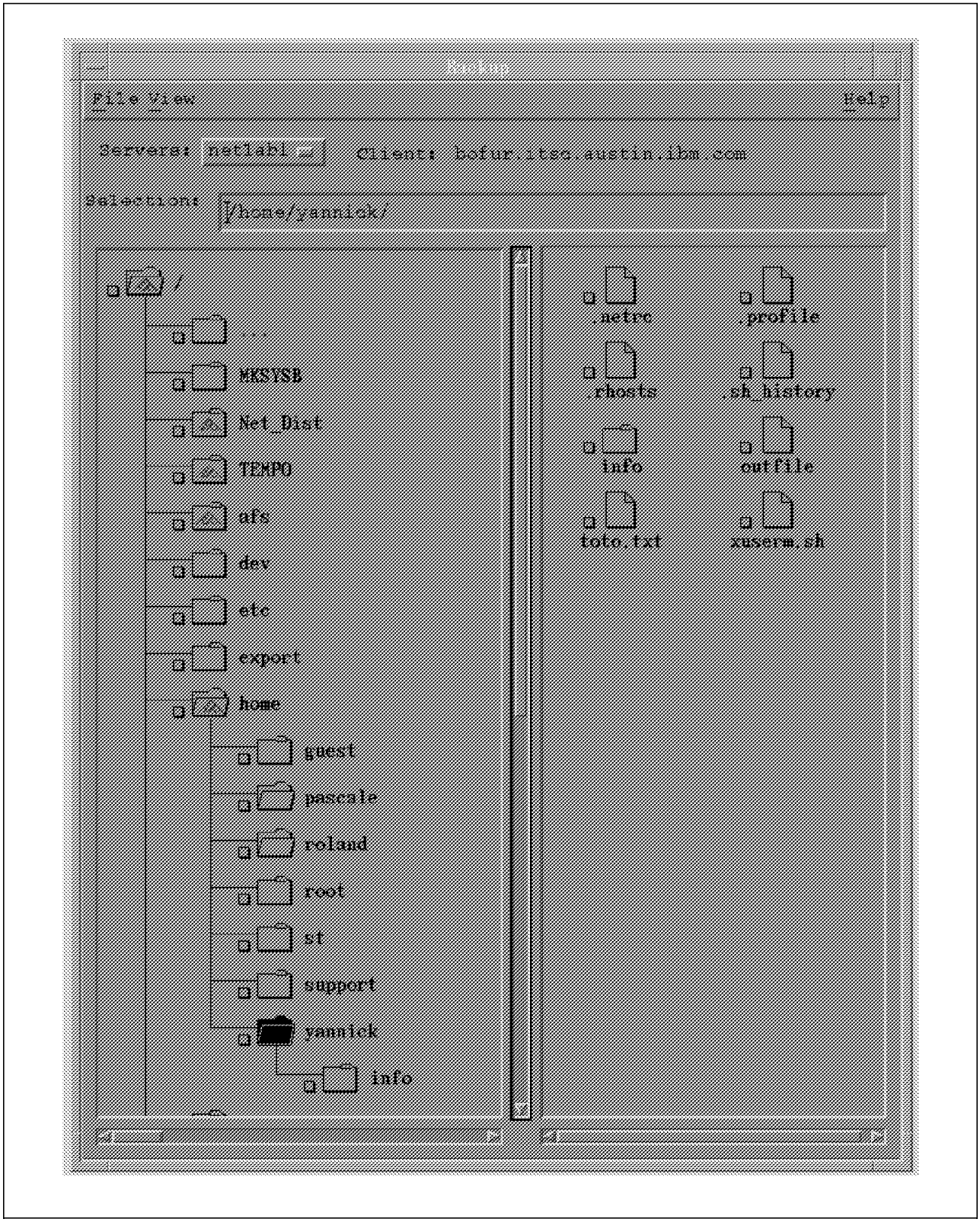

Figure 44. Customizing a Backup from Legato NetWorker Motif User Interface

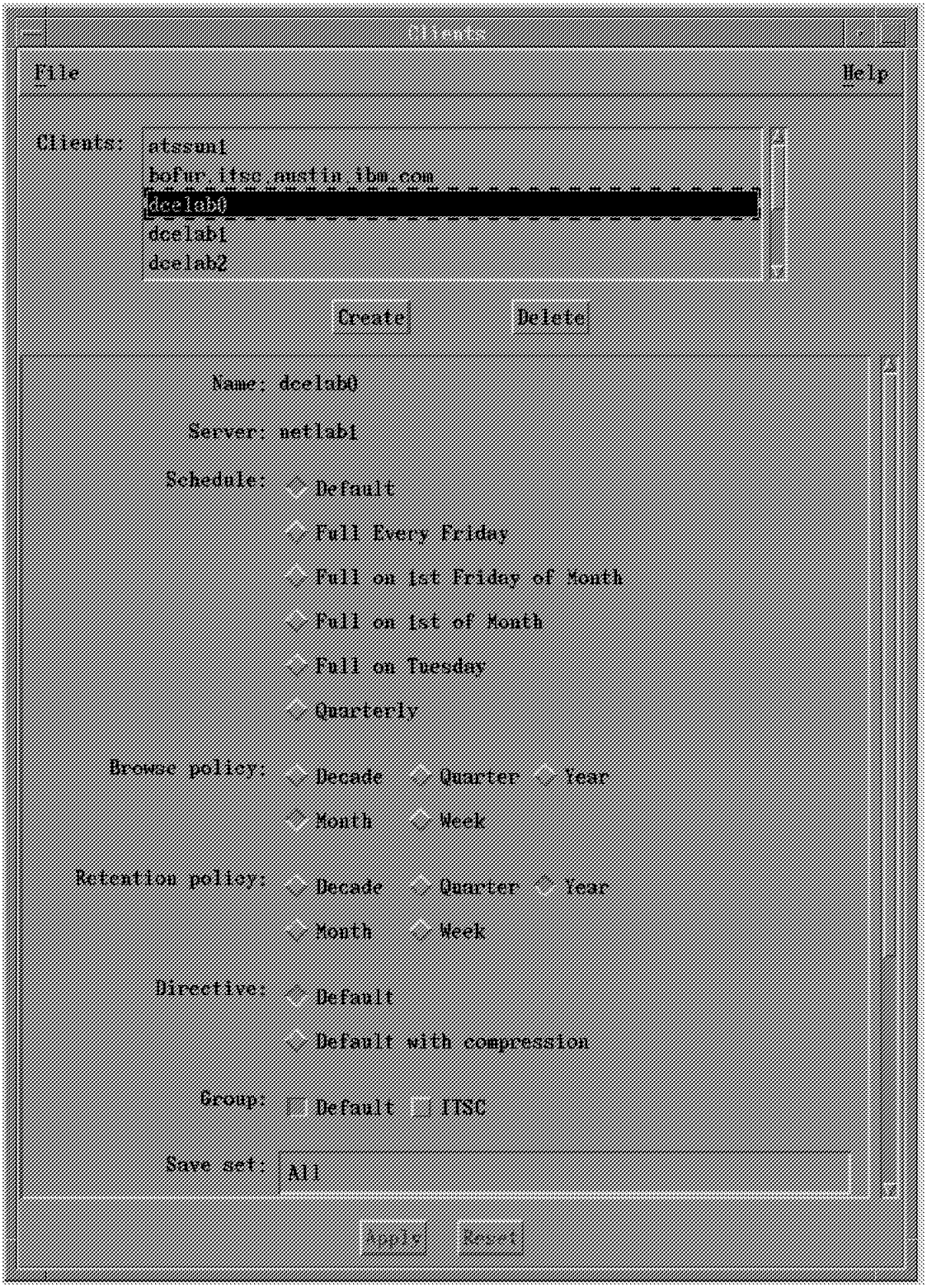

Figure 45. Setting Up Clients Definition for Automatic Backup

5.6 Time Considerations for Backups Across Networks

When you are using a product like Legato NetWorker to backup across a TCP/IP network, your backup time will be given by the three following times:

- Time for reading and compressing the data on the client machine. This time is related to the CPU speed of the client machine.
- Time for the transmission of the compressed files from the client to the server. This time come from the slowest part of your network between the client and the server.
- Time for the server to write on media the compressed files. This is related to the type and number of backup devices you have on the server. Legato NetWorker is able to multiplex the backup's data onto a tape drive or multiple tapes drive at a time.

5.6.1 Testing Legato NetWorker through a TCP/IP Network

For instance, here are the results of some basic tests done between a RISC System/6000 Legato NetWorker client and a Legato NetWorker server.

We used AIX Netview/6000 and AIX System Monitor/6000 to measure the TCP/IP traffic and the CPU loads during this test.

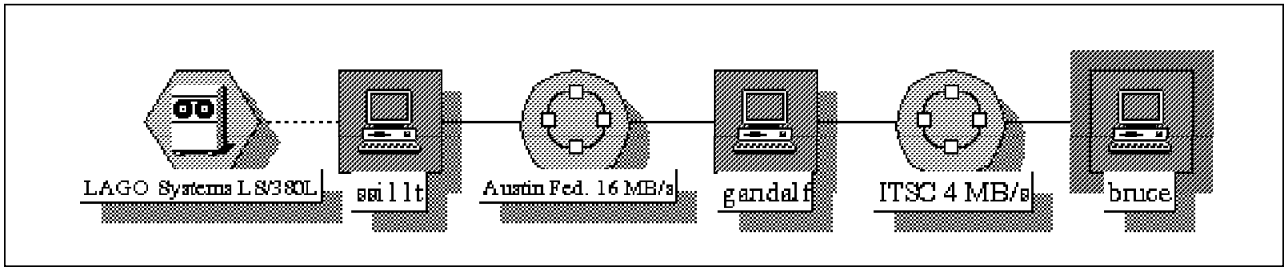

Figure 46. Network Architecture of the Test, bruce is the Client, ssi11t is the Server.

The test is to backup the /home filesystem of the machine bruce to the Legato NetWorker server named ssi11t. These machines are interconnected through the TCP/IP network of the IBM Austin site. ssi11t is directly plugged into the 16Mb/second tokenring. The client bruce is plugged in a 4Mb/second tokenring. A RISC System/6000 model 930 named gandalf is used as a TCP/IP gateway between the two tokenrings. /home filesystem is about 41MB and is 83% used, which means we have 34MB to backup. The Legato NetWorker server has a LAGO Systems LS/380L DataWheel 8mm tapes carrousel.

```
Devices:
/dev/rmt1.1(J)      8mm 5GB    netlab1.007 writing at 43KB/s, 5.0MB
/dev/rmt2.1(J)      8mm 5GB netlab1.008 saving, done

Sessions:
bruce.itsc.austin.ibm.com:/home/ saving to netlab1.007 5.0MB

Messages:
Wed 17:44:00 bruce.itsc.austin.ibm.com:/home/ saving to netlab1.007
```

Figure 47. Legato NetWorker Message Window

The backup started at 17:44:00. You can see in Figure 47 the messages the user (or any user connected to Legato NetWorker at this time) sees on his Motif interface. The device message says, that the tape drive named /dev/rmt1.1 is working, writing at 43KB/second on the tape. At this time 5MB had already been written. The session message says that, at this point in time, only the client bruce is using the server and what this client is doing: saving /home. We also see how much data had been sent by this particular client (5MB). The Messages window is a logging of what happens during the time.

```
Messages:
Wed 17:44:00 bruce.itsc.austin.ibm.com:/home/ saving to netlab1.007
Wed 17:48:15 bruce.itsc.austin.ibm.com:/home/ done saving to netlab1.007 14MB
```

Figure 48. Legato NetWorker Message Window

In Figure 48 you see that the backup took 4 minutes and 15 seconds. 14MB had been physically written to the tapes. That means that the compression factor had been 2.43 (34MB in /home, 34MB/14MB=2.43).

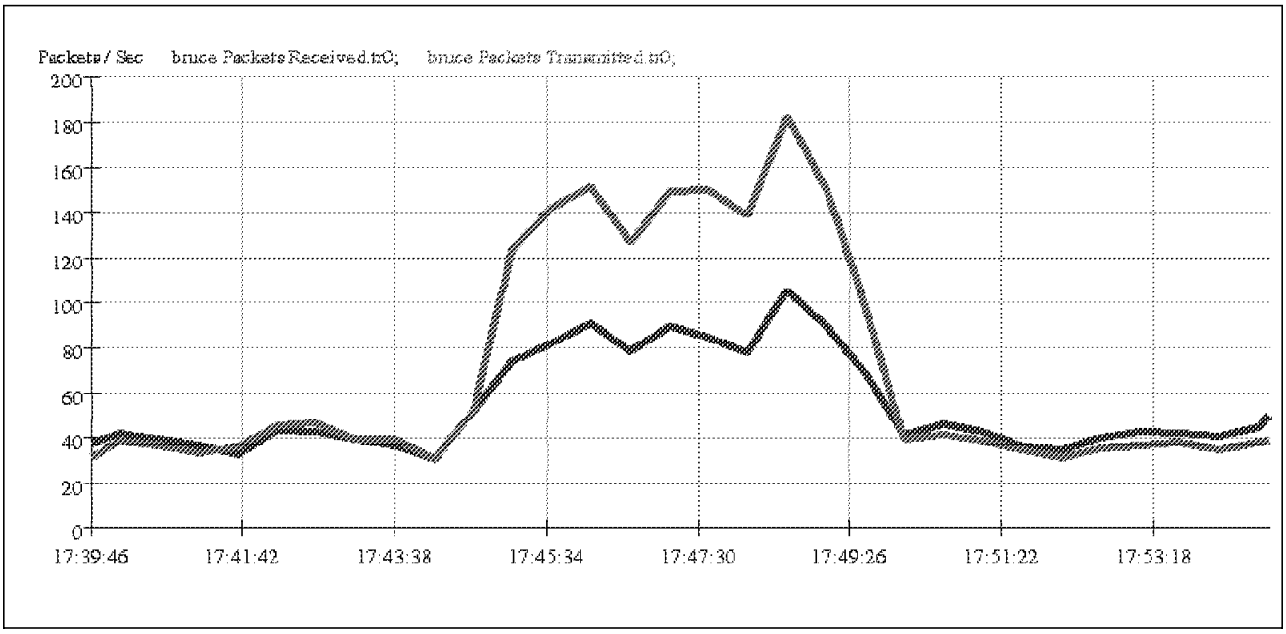

Figure 49. TCP/IP Traffic on bruce in Packets/Second

We also monitor the resources used during this backup operation on the client machine and on the network. In Figure 49, you can see the TCP/IP traffic on the machine bruce (the client) during the test. The throughput is about 150 packets per second on the network.

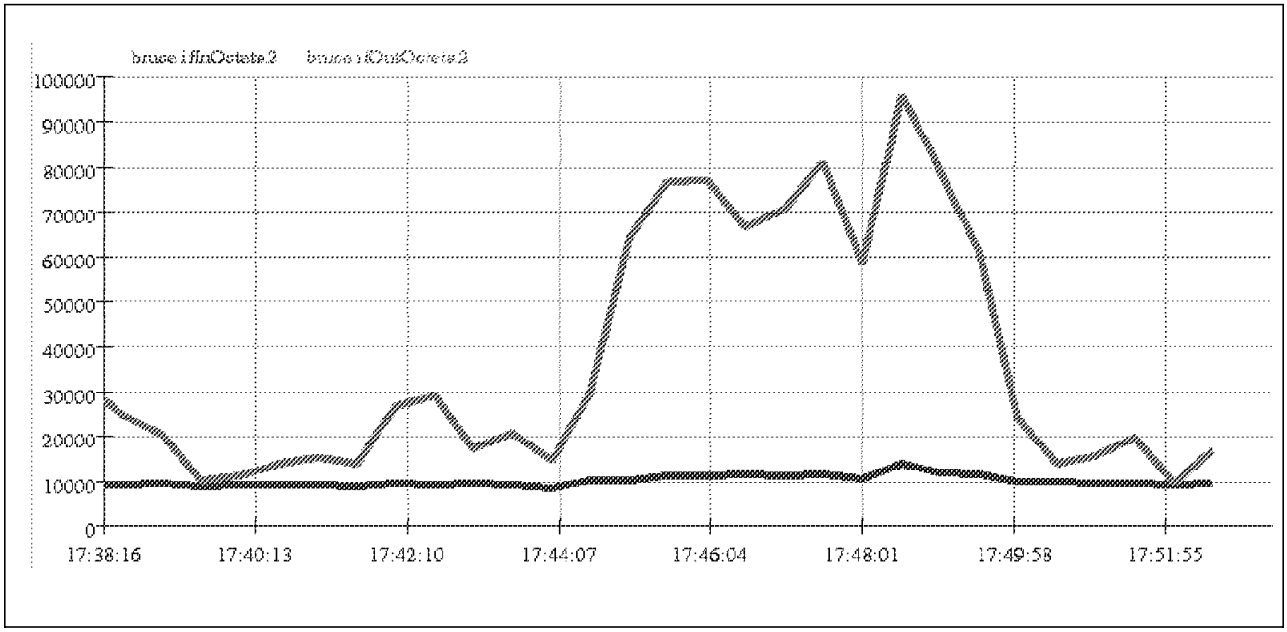

Figure 50. TCP/IP Traffic on bruce in Bytes/Second

In Figure 50, this traffic is given in bytes per seconds. We can see that the throughput is about 60KB/s in average (70KB/s minus the 10KB/s that was already here before any backup operations).

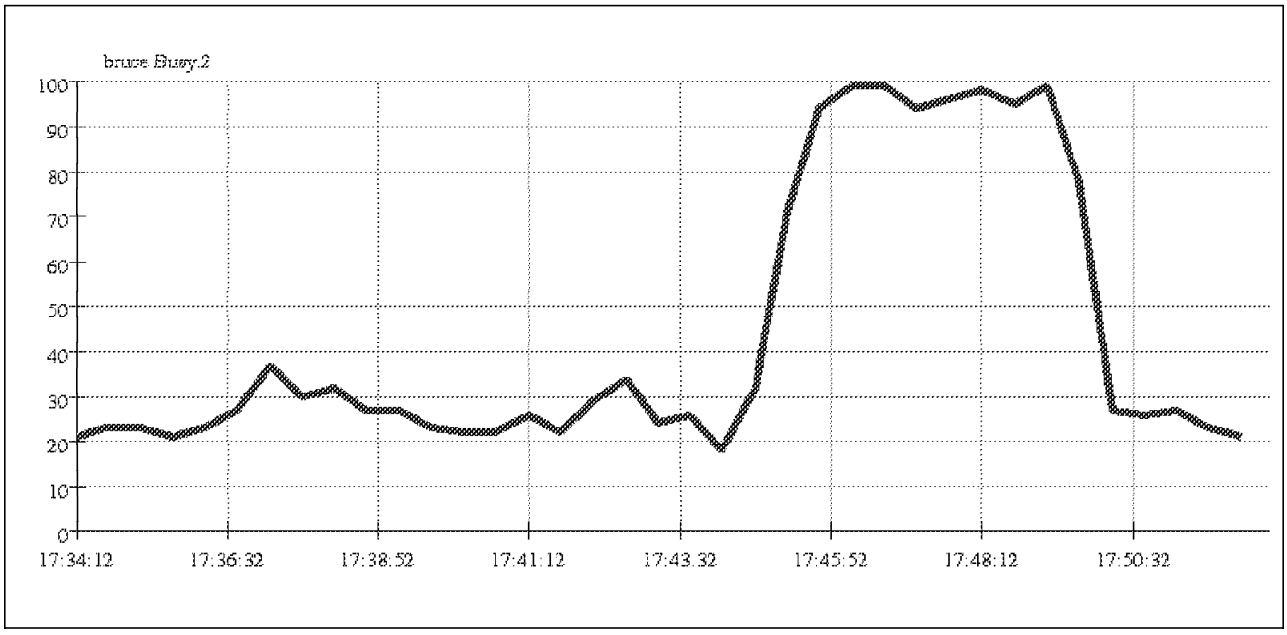

Figure 51. Total CPU Utilization on bruce (user+kernel)

Legato NetWorker allows you to backup with compression. The compression is done on the client machine, before any transmission through the network. This reduces the network traffic but requests CPU in the client machine. For instance, in Figure 51, you have the total CPU usage (CPU user + CPU kernel) on bruce during the test (this machine is a RISC System/6000 model 52H operating with 25 MHz).

As shown with the above test, the time needed for your backup depends on the client, the server, the backup media, the networks and also the strategy which are used. You also have to check your tool for the way it operates, for instance, where and when the compression is done.

Chapter 6. Distributed Batch Job Management

Batch job management is not a natural operation under the UNIX operating system. Batch job management came from the commercial type machines and UNIX was not primary designed for commercial uses. Therefore, UNIX has no batch job management included.

6.1 Considerations

The goal of running jobs as batch jobs is to get better utilization of the CPU. Basically this means, to run jobs after office hours or to distribute them in a network to other machines. Here are some examples, on how this can be done:

- Starting jobs at a later time
- Running jobs during the night
- Executing jobs depending on CPU load
- Use other CPUs for calculating

This can be done either with standard commands, with a network queueing sytem or other applications.

6.2 Standard Batch Job Commands

Some basic job management exists in UNIX or AIX commands. They are not part of an integrated job management product but just base operating system commands like:

- The `at` command: starts a job at a certain point in time
- The `batch` command: runs a job at a later time when the system loadlevel permits
- The `cron` daemon: starts a job at a defined frequency, depending on on the entries in the crontab file (like every day at 3am)
- The standard queueing system (qdaemon)

The standard queueing system can be used, when you define a queue with a shell as a backend program. This is a way to make sure that CPU intensive jobs are serialized and not running at the same time. None of the other commands are very powerful. For them, a job is just an executable. If you need logic between many jobs to check for completion or errors, then you need to write scripts using your favorite shell language (ksh or bsh). Also none of them allow you to take advantage of a cluster architecture.

6.3 Network Queueing Systems

Network queueing systems use the standard queueing system to distribute jobs to other machines. Additionally they are notifing the user via mail about their jobs. Without load balancing, it is a static configuration. This means a user submits a job to a dedicated queue, which is a remote queue. The job is then executed on the remote machine and the user gets the results back. Some of the network queueing systems have a load balancing feature. There is a

© Copyright IBM Corp. 1993,1994

program running on each machine which reports the CPU load periodically to all machines which have balanced job queues defined. When a user now submits a job to a loadbalanced, remote queue, the backend program decides depending on the load information, to which machine the job should be queued. This gives more flexibility and better utilization to the whole system.

6.4 Job Scheduling Application

Besides the standard commands and the network queueing systems there is a job scheduler application from IBM, called IBM LoadLeveler. LoadLeveler′s approach to network job scheduling for UNIX batch jobs differs from the network queueing system approach. While the network queueing system based approach involves managing pipe and batch queues, LoadLeveler lets you view the total computational capabilities as a single source. Therefore, job requirements are dynamically matched with the best available resources based on your preferences, globally defined scheduling policies and priorities. You will get an overview of the IBM LoadLeveler on the following pages.

6.5 IBM LoadLeveler

IBM LoadLeveler is a job scheduler application that runs as a set of daemons on each node in your RISC System/6000 network or on IBM Scalable POWERparallel Systems 9076 SP2 and 9076 SP1. It also runs on Sun SPARCstation** Systems, HP Apollo** 9000/700 Systems and Silicon Graphics IRIS Systems**. It provides a facility for building, submitting, and processing batch jobs quickly and efficiently in a dynamic environment.

LoadLeveler has been primary designed for Numeric Intensive Calculation jobs.

LoadLeveler has both a command line interface and a Motif based GUI. The user can submit and cancel jobs, monitor their status, set and change job priority. The system administrator can perform the same tasks as end users plus checking and changing machine status, assigning job classes, starting and stopping jobs and scheduling jobs on all nodes.

LoadLeveler provides accounting information on completed jobs and stores this information in a file. This job information includes a listing of the job parameters, the amount of resources consumed by the job, as well as other job related data. Administrators can collect data on one job or on all running jobs and merge this information into one central file. This feature can help track the different types of jobs being processed, as well as how many resources each type of job requires for processing.

LoadLeveler runs in environments where the Network File System** (NFS**) or Andrew File System** (AFS**) is installed. NFS or AFS is used to giving to the running job a single image of the disk resources. For instance a job is using a file xy which is physically on machine A. Machines A,B and C are running LoadLeveler. You submit a job for execution on machine A, based on the CPU load conditions of the machines A and B at this time, LoadLeveler may decide to start this job on machine C. Then all the disks I/O from the job to the file xy will be done through the TCP/IP network using NFS or AFS.

LoadLeveler supports machines outside the LoadLeveler cluster that run a network queueing system. You can run your LoadLeveler jobs or network

queueing system scripts either on machines in the LoadLeveler cluster or on machines outside the cluster. If you are submitting a network queueing script to LoadLeveler processing, LoadLeveler handles it in one of the following ways:

- LoadLeveler translates the network queueing system options in the script to equivalent LoadLeveler commands and submits the job to a machine in the LoadLeveler cluster for processing.
- LoadLeveler routes the network queueing system script to a network queueing system machine outside of the LoadLeveler cluster for processing.

You can also have jobs that access relational databases using their remote transparent access libraries. These jobs should not use checkpointing (see 6.5.3, “Using Checkpointing” on page 69 and 6.5.5.1, “Limitations of Checkpointing and Job Migration” on page 78).

6.5.1 What Types of Machines are in the Cluster?

Your LoadLeveler cluster may contain many machines. Each of these machines can play one or more of the following roles:

- Allow users logged in to this machine to submit LoadLeveler jobs. This type of machine is referred to as a submit-only machine.
- Participate in the scheduling of LoadLeveler jobs. This type of machine is referred to as a public submission machine.
- Execute LoadLeveler jobs. This means that if your machine is a submit-only machine, it contacts a public submission machine that in turn, contacts a machine that will execute the job. Keep in mind, however, that one machine can play all three of these roles. Therefore, you may be able to submit a job on a machine and have it executed on the same machine.

6.5.2 Central Manager

In each LoadLeveler pool there is a special machine known as the Central Manager. In addition to the roles listed previously, the Central Manager′s principal function is to coordinate LoadLeveler related activities of all the machines in the pool. He maintains the status of all machines and jobs, makes decisions on where jobs should be run and responds to user queries.

The Figure 52 on page 68. illustrates the relationship between the central manager and the other machines in the cluster.

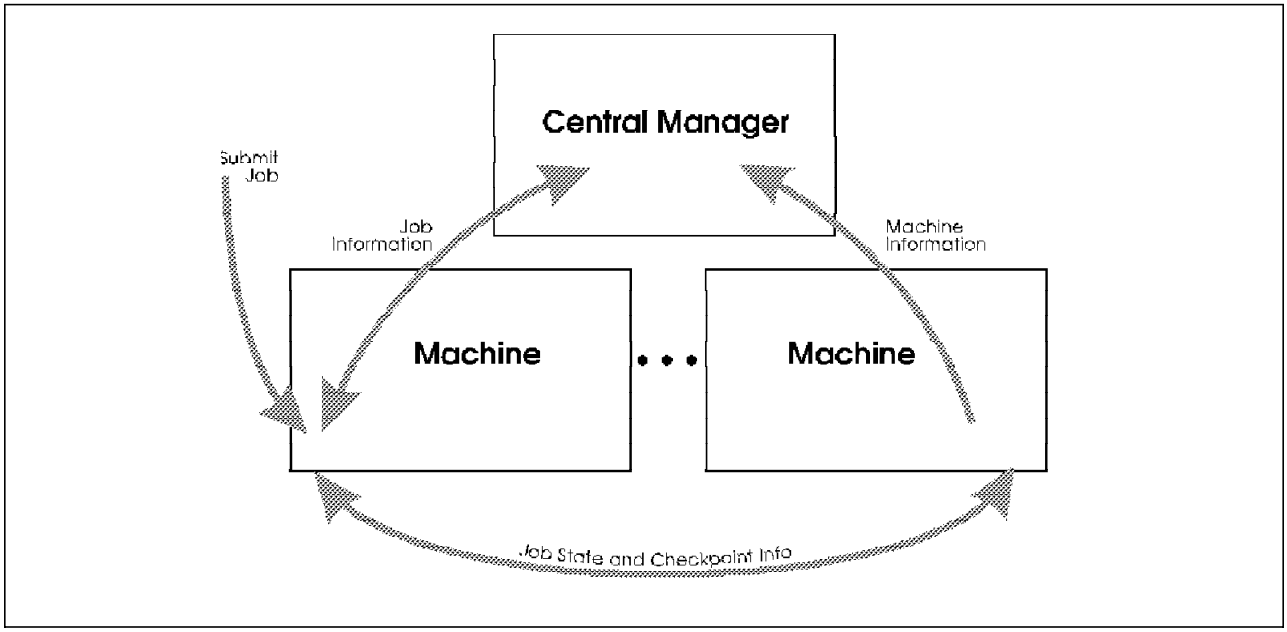

Figure 52. IBM LoadLeveler Central Manager

After a job has been selected by the Central Manager for execution, the job can be in one of several states as shown in Figure 53.

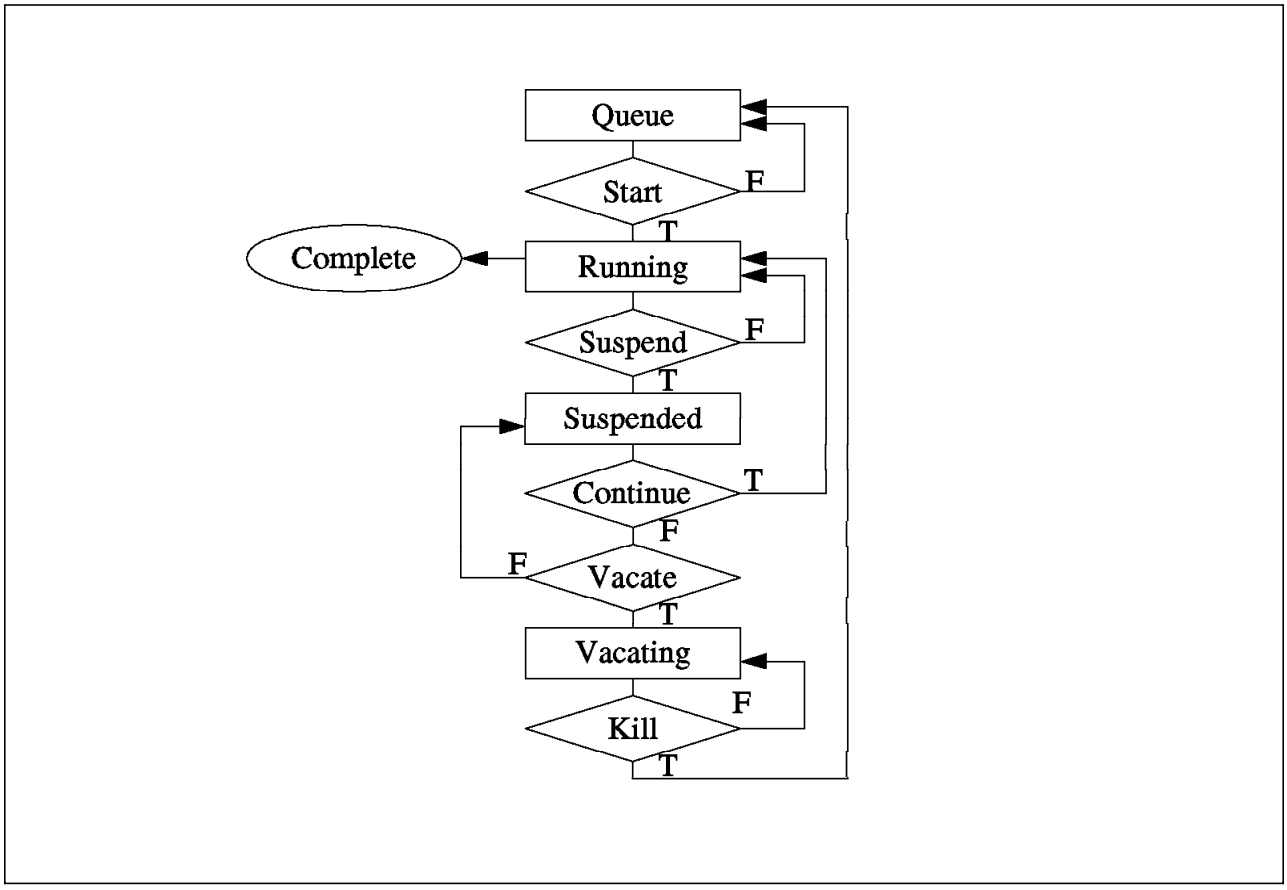

Figure 53. IBM LoadLeveler Job Management

Start The job is running on a machine that meets all the resource requirements requested by the job.

Complete The job has terminated processing.

Suspend The job has been suspended. The machine hosting this job may have become busy with many interactive foreground activities and is no longer willing to support a CPU-intensive batch job. Suspended LoadLeveler jobs will either be continued or vacated.

Continue The job has resumed execution and will run to completion unless it is suspended again.

Vacate The job is being vacated from the machine that has been executing the job. Jobs suspended for an amount of time greater than some threshold are usually vacated. If the job is a LoadLeveler checkpoint job, the checkpoint files are sent to the machine that has been monitoring the execution of this job. Under some circumstances the job will go from the Vacate state to the Kill state. Usually, the job is simply returned to the queue for dispatching. When it re-enters the Start state at some future point in time, a non-checkpoint job will restart from the beginning and a checkpoint job will resume from the last checkpoint.

Kill A job taking too much time to vacate may be moved to the Kill state. The job being vacated may have a very large checkpoint file and the time spent in vacating the job has exceeded some threshold. When the job enters the Kill state, the attempt to send checkpoint files to the machine that has been monitoring the execution of the job will be aborted. The job is returned to the queue for dispatching. When it re-enters the Start state at some future point in time, a non-checkpoint job will start from the beginning and a checkpoint job will resume from the last successful checkpoint.

Criteria used to determine when a LoadLeveler job will enter Start, Suspend, Continue, Vacate, and Kill states are defined in the LoadLeveler configuration files and may be different for each machine in the pool. They may be modified to meet local requirements.

6.5.3 Using Checkpointing

Checkpointing is a method of periodically saving the state of a job so that, if for some reason, the job does not complete, it can be restarted from the saved point. Only serial jobs can be checkpointed, not parallel jobs. Checkpointing also allows the migration of a long running job to another machine when the first machine is required for other work. The machines have to be of the same architecture: you can submit jobs from a RISC/6000 to a RISC/6000 or from an HP to another HP machine. Migration can be initiated automatically or upon LoadLeveler commands from the system administrator.

Any executable program could be used as a LoadLeveler job, but to enable checkpointing you must link special LoadLeveler libraries. These are available for C and FORTRAN. At startup and at checkpoint time, a checkpoint file is created on the executing machine. The checkpoint file contains the text data, stack segments, register contents, and the status of the open files at the time of the checkpoint. See section 6.5.5.1, “Limitations of Checkpointing and Job Migration” on page 78 for more information.

6.5.4 Example of Submitting a Job with IBM LoadLeveler

This is a very simple example of using the IBM LoadLeveler Motif Interface, which can be started with the `xloadl` command, to submit and monitor a job in a cluster of two RISC System/6000s connected through a token-ring LAN. This example assumes that IBM LoadLeveler is already installed, configured and started. You can find more information about installing and customizing LoadLeveler in the *IBM LoadLeveler User's Guide* and *IBM LoadLeveler Administration Guide*.

6.5.4.1 Building a Job by using the GUI

In our example, the cluster is composed of only two machines, named andrea.itsc.austin.ibm.com and francois.austin.ibm.com.

Figure 54. LoadLeveler Cluster is Started with Two Machines, Now Both are Idle

In Figure 54, you can see from the Machines control panel that both machines are idle. There are no job currently running in the cluster.

Using the Build Job option in the File menu of the Jobs control panel (see Figure 54 on page 70), you can start another window where al l the parameters needed to submit a job could be entered (see Figure 55). You just have to enter the shown fields and then you can submit your job.

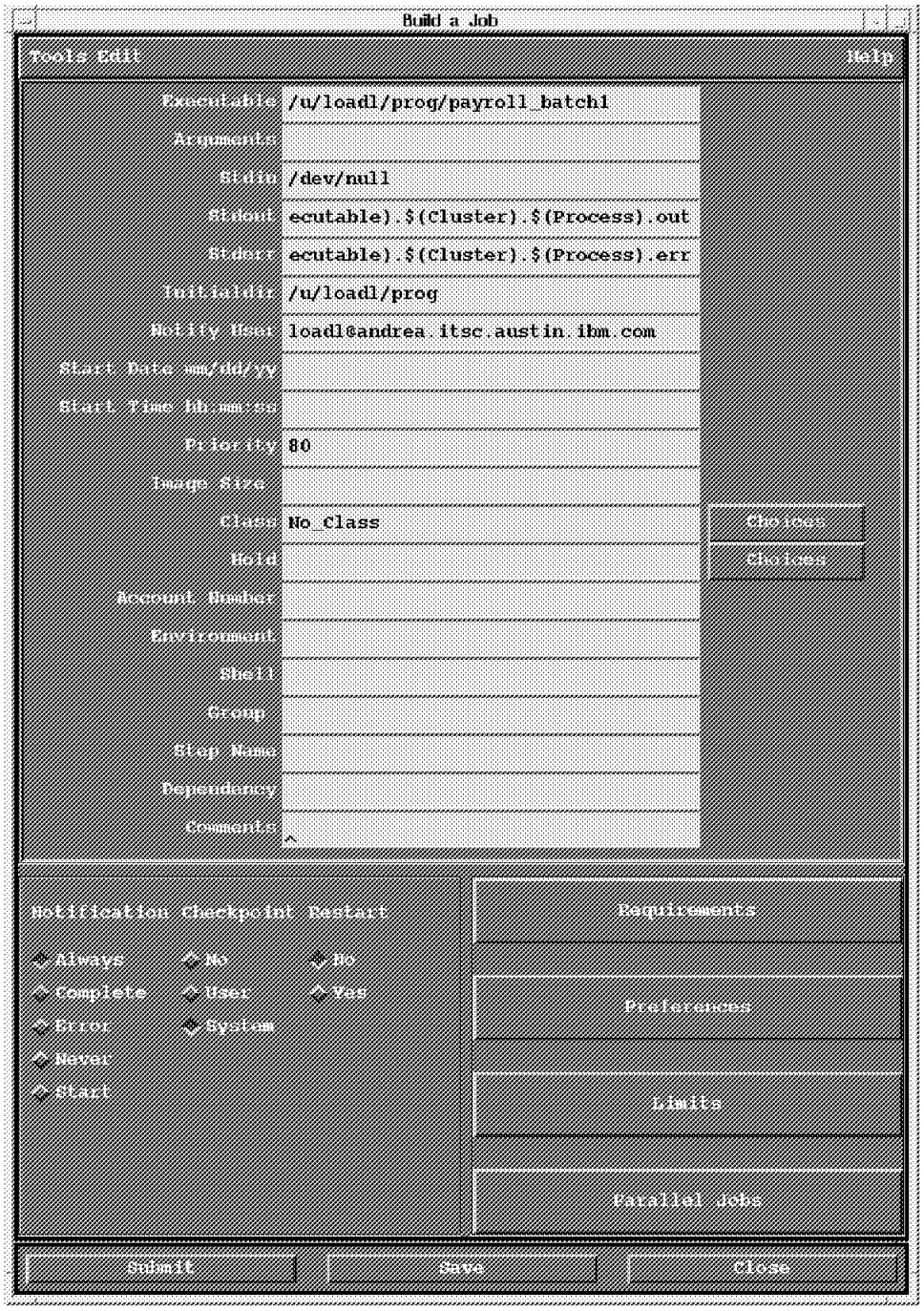

Figure 55. Building a Job Control File with the Motif Interface

For LoadLeveler, a job is just an AIX executable program. Following is a description of the fields from Figure 55:

Executable The name of the AIX executable file which is the job.

Arguments The arguments you want to give to the executable.

Stdin/Stdout/Stderr Can be redirected from/to any file or special devices

Initialdir LoadLeveler changes to this directory before running the job.

Notify User The mail address where messages (notifications) are sent for various job's status changes.

Start Date/Time Used if you want the job to be scheduled at a certain time.

Priority The AIX priority that will be used to run the executable.

Image Size Number of KBs that reflects the maximum size you expect your program to grow to as it runs.

Class A job could be assigned to a class, the administrator could use job classes as a parameter in the scheduling algorithm.

Hold A job could be submitted to LoadLeveler but not started. It is then in the hold status and the user could release it, which makes the job ready to be started.

Account Number Number identifying the job, used with LoadLeveler job accounting commands (`llacctmrg`, `llsummary`).

Environment Specifies you initial environment variables at the job start-up.

Shell Specifies which shell you want to call for the job (the default shell is /bin/sh).

Group The LoadLeveler group name to which the user belongs.

Notification switches These switches (Always, Complete, Error, Never, Start) give the user the possibility to configure on which event he wants to get a notification mail.

Checkpoint Used to indicate whether this job should use checkpointing or not. (see 6.5.3, "Using Checkpointing" on page 69). Available choices for checkpointing are User, if the user coded the LoadLeveler `ckpt()` function in his program, or System if you want LoadLeveler to automatically checkpoint the program.

Restart To restart or not the job. Default option is Yes.

Requirements To set the needed conditions for this job.

Preferences To set your preferences for this job.

Limits To give some resource limits to this job.

Parallel Jobs To set your requirements for parallel jobs scheduling.

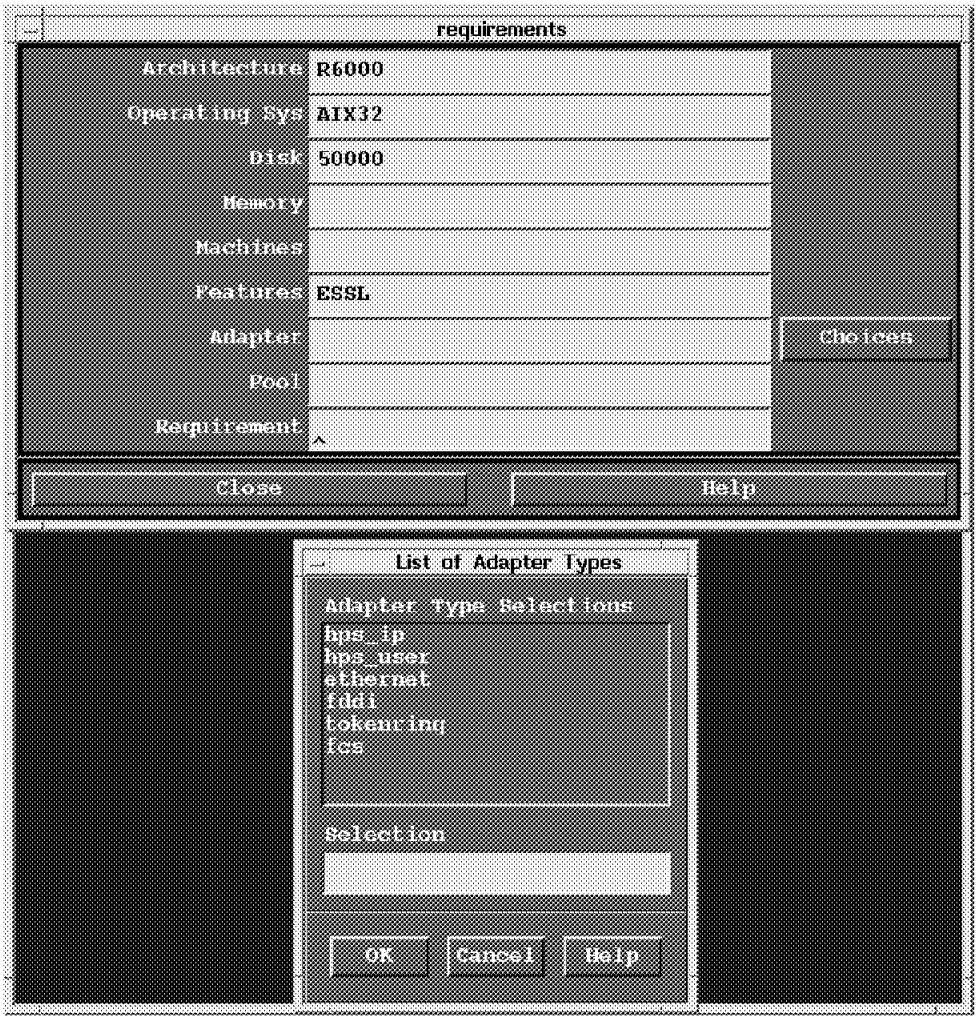

Figure 56. Setting Requirements to the Job

In Figure 56 we are setting the requirements for the machine that is executing our job:

- The job must be executed on a RISC System/6000 (LoadLeveler is designed to support several architectures in the same cluster).
- The machine must run AIX Version 3.2.
- The free disk space for the job must be greater or equal to 50000KB.
- The machine must have the ESSL scientific libraries.
- You can also give the quantity of memory the job requires at minimum for running, and specified machines where to run the job. Adapter and Pool parameters are used when processing a parallel job on the 9076 system. You can specify an adapter among Ethernet, Token Ring, FDDI or FCS. The Pool parameter is the number associated with the 9076 system parallel pool to use.

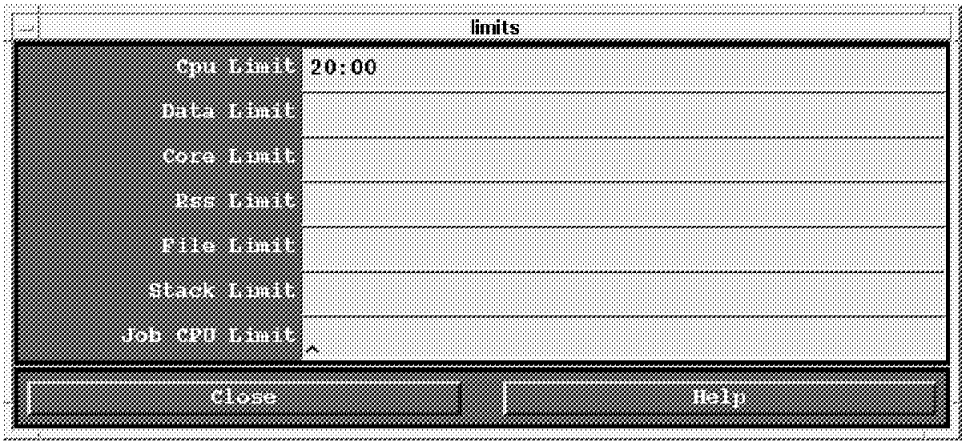

Figure 57. Setting Limits to the Job

In Figure 57, we are setting the limits for the resources our job will be allowed to consume:

- The CPU time must not exceed 20 minutes
- The job CPU limit is the maximum amount of CPU a single job step can use per processor

```
#!/bin/ksh
# @ executable = /u/loadl/prog/payroll_batch1
# @ input = /dev/null
# @ output = $(Executable).$(Cluster).$(Process).out
# @ error = $(Executable).$(Cluster).$(Process).err
# @ initialdir = /u/loadl/prog
# @ notify_user = loadl@andrea.itsc.austin.ibm.com
# @ user_priority = 80
# @ class = No_Class
# @ notification = always
# @ checkpoint = system_initiated
# @ restart = no
# @ requirements = (Arch == "R6000") && (OpSys == "AIX32") &&  (Disk >= 50000) && (Fe
ature == "ESSL")
# @ Cpu_Limit = 20:00
# @ queue
```

Figure 58. Job Control File Created using the Motif Interface

When you have filled all, this information job control file is created. This file will be submitted to the LoadLeveler as a job. It contains all the information needed by the scheduler to run the job. In Figure 58 you can see the job control file that have been generated from our example.

6.5.4.2 Submitting Parallel Jobs

With LoadLeveler, you can schedule parallel batch jobs that have been written using Parallel Virtual Machine (PVM) 3.3 (a public domain package), or any general parallel programming language which uses the LoadLeveler parallel application programming interfaces, such as the IBM Parallel Environment Library 1.2.1 or the Parallel Virtual Machine Extended (PVMe) 1.3.1.

LoadLeveler allows you to submit parallel jobs to both the 9076 system and a cluster of heterogeneous workstations. When building a parallel job, you have to specify requirements on the Build a Job panel (see Figure 55 on page 71):

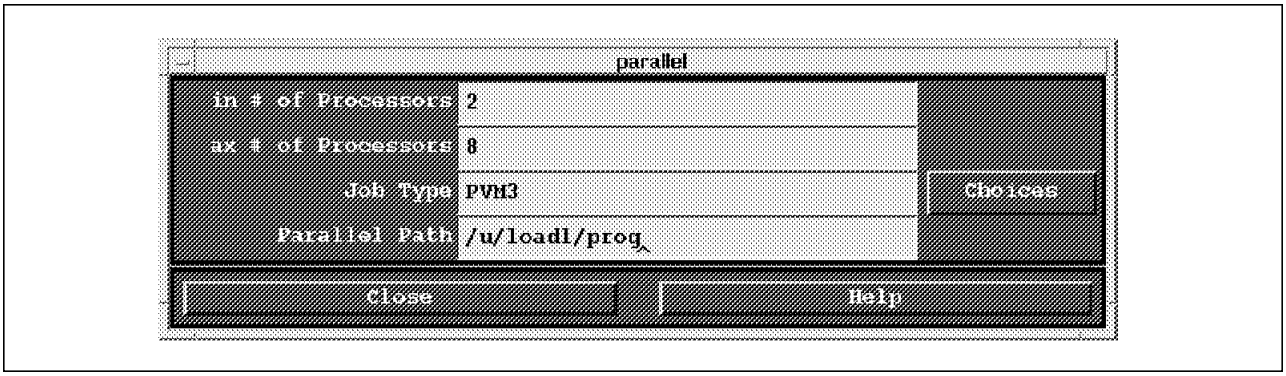

Figure 59. LoadLeveler Parallel Job Requirements

Min,Max # of Processors Specifies the minimum and maximum number of processors required to run the parallel job.

Job Type Gives the type of job step to process: Serial, PVM3 or Parallel.

Parallel Path Specifies the directory that defines where the PVM3 executables are located (see Figure 59).

6.5.4.3 Submitting a Job to LoadLeveler

You can submit the job using the `llsubmit` command or the Submit Job option of the File menu in the Jobs control panel (see Figure 54 on page 70).

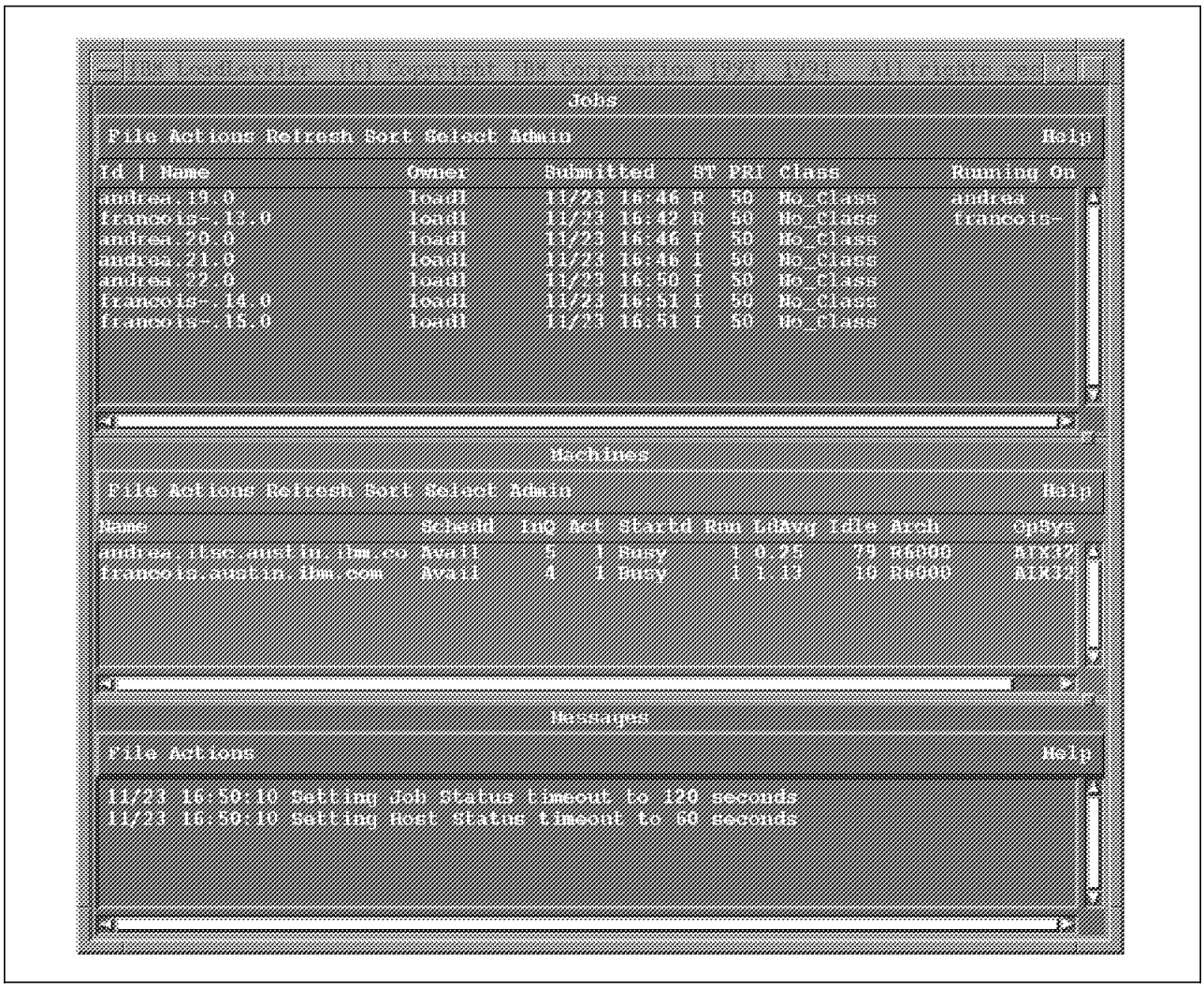

Figure 60. LoadLeveler Motif Interface Main Window

In Figure 60, you can see that seven jobs have been submitted. In the Machines control panel, you see that now both machines in the cluster are busy. In the Jobs control panel, you see that there are two jobs running (one on andrea and one on francois) and five are idle. By default, LoadLeveler is able to run two jobs simultaneously on one machine. The remaining jobs stay idle till one machine is ready to execute it. The policies LoadLeveler is using for scheduling jobs is fully configurable. See *IBM LoadLeveler Administration Guide* and *IBM LoadLeveler User's Guide* for more information.

6.5.4.4 Job Notification

LoadLeveler sends UNIX mail messages to the users when something happens to their jobs.

```
From: LoadLeveler@andrea.itsc.austin.ibm.com
To: loadl@andrea.itsc.austin.ibm.com

Subject: andrea.itsc.austin.ibm.com.19
Your job step, " andrea.itsc.austin.ibm.com.19.0" has started.

Starter information:

         Submitted: Thu Nov 24 19:46:01 1994

        Executable: /u/loadl/prog/payroll_batch1
     Job Step Type: NonParallel
       MachineName: andrea.itsc.austin.ibm.com
       Architecure: R6000
  Operating System: AIX32
```

Figure 61. Mail Sent to Notify the User that One of his Jobs has been Started

In Figure 61, you can see the mail sent to notify the user loadl on machine andrea, that his job /home/loadl/prog/payroll_batch1 has been started.

```
From: LoadLeveler
Subject: andrea.itsc.austin.ibm.com.19

Your LoadLeveler job step
         andrea.itsc.austin.ibm.com.19.0 (/u/loadl/prog/payroll_batch1 )
has exited.

Status for machine andrea.itsc.austin.ibm.com:
        The job step exited normally with code 0

This job step was dispatched to run 1 time.
This job step was rejected by a starter 0 times..

Submitted at: Thu Nov 24 19:46:01 1994
Exited    at: Thu Nov 24 19:48:04 1994

               Real Time:   0 00:04:03
      Job Step User Time:   0 00:00:00
    Job Step System Time:   0 00:00:00
     Total Job Step Time:   0 00:00:00

       Starter User Time:   0 00:00:00
     Starter System Time:   0 00:00:00
      Total Starter Time:   0 00:00:00
```

Figure 62. Mail Sent to Notify the User that one of his Jobs has Finished

When a job is finished, the user receives a new mail message notification with the summary of his job execution including CPU utilization and errors. For instance, if the job program stopped with a core dump error, this will be stated in the mail message.

6.5.5 LoadLeveler′s Limitations

LoadLeveler is coming from the scientific world. It′s primary design is for CPU intensive long jobs and not for critical business commercial applications.

6.5.5.1 Limitations of Checkpointing and Job Migration

When you link a C or Fortran program with the LoadLeveler libraries in order to use checkpointing, some system calls are changed to allow LoadLeveler to save state related informations. For example, the `open()` system call records all the files used by the process, so that at checkpoint time, the current file pointer can be saved and then restored at restart time.

For some processes, it is impossible to obtain or recreate the state of the process. For this reason, you should only checkpoint jobs whose state is simple enough to checkpoint and recreate. A job that is long-running, computation-intensive, and runs a single process, is an example of a job well suited for checkpointing.

Any checkpointing job should not use the following system calls in order to prevent unpredictable results from occurring:

- Administrative (for example audit or swapqry)
- Signals
- Fork
- Dynamic libraries loading
- Shared memory
- Semaphores
- Messages
- Internal timers
- Set user/group ID

Chapter 7. Configuration Management

The configuration of a machine makes it unique in your network. In fact, it is virtually impossible to make two machines exactly identical. At least some values like your Ethernet or token-ring address will be different. If you have only one RISC System/6000 the minimum tasks to do are:

- Configure the system the first time
- Back up the configuration
- Maybe restore the configuration in case of crash

But, if you have many RISC System/6000s, the tasks can become more complex. This is because, on the contrary of AIX and applications versions, most of the time you will have to have one configuration per machine. You may have two or three used versions of AIX and one or two versions of your applications currently in your organization but 200 or 300 different configurations!

Then comes another question: what is a configuration? A configuration is made of data (like application data), but this data is deeply linked to the operating system. This means that, very often, the data itself is not self sufficient. You need also to have the ″know-how″ corresponding to this data. This ″know-how″ could be named ″methods″. An example of that is the network configuration. You may have the data that you need (your TCP/IP address, your network mask), but if you do not know the commands and special files to use, you will never be able to make your network work.

This rapidly growing complexity of configuration management makes it one of the most difficult tasks for the system administrator of a large distributed network of UNIX machines. The complexity of the configuration of large UNIX servers could be equivalent to what we are used to for the mainframes but with several tens of hundreds of systems.

Configuration management also becomes very complex when you have a multi-vendor UNIX network. Unix has been standardize to be faster for applications than for administration purposes. Most of the leading UNIX machines manufacturers have completely different configuration mechanisms. This leads to technical and human difficulties when you want to operate large multi-vendor distributed systems.

7.1 The Future

The configuration problem has been recognized by the industry for several years. The main multi-vendors effort is OSF/DME. Today (1994), it is not ready to be use in a real production environment. OSF/DME is not limited to configuration management, but we think that configuration management will be one of his most important functions in a multi-vendor distributed systems.

© Copyright IBM Corp. 1993,1994

7.1.1 OSF/DME

The Open Software Foundation was created in 1988 by a consortium of vendors: Apollo, Bull**, DEC**, HP, IBM. This non-profit organization wanted to develop a set of products that would be usable by many vendors, not only for sponsors and members. OSF was an alternative to UNIX International sponsored by AT&T** and SUN. OSF publishes a Request For Technology (RFT) and any manufacturer may propose a technology for acceptance by the foundation. OSF has selected a number of products from RFT′s such as Motif, DCE**, OSF/1**, DME, ANDF**. Motif and DCE are now well known standard products. OSF/1 is an operating system used by DEC on Alpha** platforms and by IBM on AIX/ESA* platforms. Application Neutral Distributed Format (ANDF) is in standby state. Distributed Management Environment (DME) is a very complex product because there are not any real standard products in this market, but many individual products and partial solutions to address different parts of distributed management. In addition, DME takes a long time to define an RFT and then to evaluate and select products. Some product examples are:

- Palladium** from project Athena** of MIT, which is a print manager.
- System Resource Controller (SRC) from AIX of IBM, to manage system resources such as network protocols.

7.2 AIX Configuration Management Today.

From its initial design, IBM has tried to improve the configuration mechanism of it′s UNIX line of product with AIX Version 3. This has been sometimes done by using none-UNIX type of techniques coming from the traditional business oriented machines. From the beginning UNIX was made from different pieces. Every piece has its own configuration files and methods. What AIX Version 3 was trying to do is to have a central repository for configuration data and methods: the Object Database Manager (ODM).

7.2.1 ODM Configuration Database versus UNIX Flat Files

ODM is a light Object Oriented Database Manager built-in AIX. There are many independent databases managed by ODM. Those databases are stored in the directory /etc/objrepos. The administrator could use ODM for its own usage, creating its own database. He could even directly modified the contents of the system databases. This is strongly not recommended. The systems ODM databases should be modified only when you use the AIX high level commands like `mkdev` to create a new device. Those commands will ensure that all the modifications in ODM objects are made in a consistent manner.

Also, many commands exist to extract the information stored in the ODM system database. A command like `lsdev` will allows you to different queries to ODM and get back detailed information about device configuration.

In summary, ODM configuration databases may look unfamiliar to a traditional UNIX system administrator but it gives to AIX a unique internal object oriented base to build system management tools.

7.2.2 SMIT

The System Management Interface Tools (SMIT) is part of the base AIX Version 3. SMIT has two main functions:

- Be an intuitive way of doing most of the configuration and administration work on a RISC System/6000. SMIT has simple English speaking menus where an unexperienced user can browse until finding the configuration or administration panel he needs. SMIT menus are also translated for many non-English speaking countries.
- Insure high level methods and consistency for configuration management. Very often, configuration and administration tasks are complex tasks made up of several smaller tasks. SMIT ensures for you that those elementary tasks are made for you in the right order, with the proper error checking. For instance, changing a TCP/IP parameter may be composed of:
  - Make the change in the ODM databases using the chdev command
  - Update several flat files
  - Refresh several TCP/IP daemons

  With SMIT, those three elementary actions are seen from the user as only one operation.

```
                                System Management

Move cursor to desired item and press Enter.

  Installation and Maintenance
  Devices
  Physical & Logical Storage
  Security & Users
  Diskless Workstation Management
  Communications Applications and Services
  Spooler (Print Jobs)
  Problem Determination
  Performance & Resource Scheduling
  System Environments
  Processes & Subsystems
  Applications
  Using SMIT (information only)




F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 63. ASCII Mode User Interface for SMIT

SMIT could be used on an ASCII full-screen terminal like in Figure 63 or in its Motif version as shown in Figure 64 on page 82.

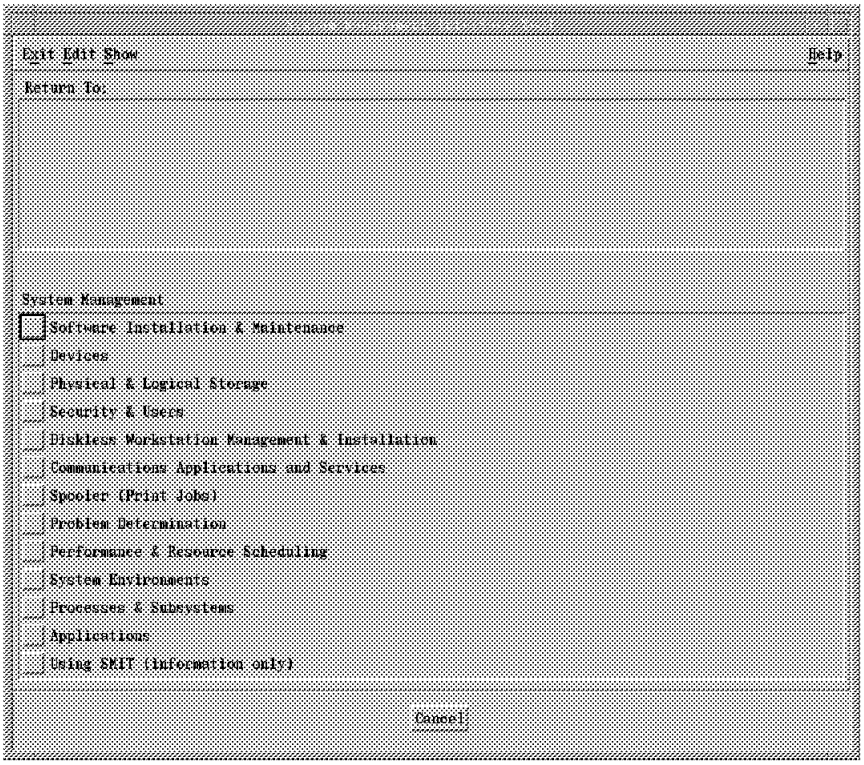

Figure 64. Motif Graphical User Interface for SMIT

SMIT does not have built-in distributed capability. If you need to make distributed configuration tasks, SMIT could help you by logging the shell script commands it has used on one machine and let you use to reproduce this same action elsewhere. This script may need to be tailored or enlarged to suit multiple replay. For instance, if you need to add a user on 10 machines, do it on the first one. Get the command that SMIT has used in the smit.script file and rexec on the nine remaining machines.

In some situations, this may be more complicated. For instance, if you want to add an asynchronous line device (tty) and then enable this terminal to be used for login connections, you will have to add the script shell lines around those generated by SMIT. Why ? Because the script file generated by SMIT is suited for the a particular configuration of the machine where you make it run. If on this machine there is no other tty defined, the new one will be named `tty0` and the next command in the smit script file will be `penable tty0`. If you run this script on an other machine that has already configured an other asynchronous line for a printer, for instance, then the new tty will be named `tty1` and the `penable tty0` command will fail.

7.2.3 AIX Visual Systems Management Graphical User Interface

IBM has introduced a new Graphical User Interface (GUI) to configure and administrate key AIX functions: the AIX Visual Systems Management GUIs. This is a completely new GUI design using Motif Version 1.2 and a drag and drop style. AIX Visual Systems Management GUIs are four independent programs for:

- Users and Groups Management, Figure 67 on page 86
- Logical Volumes Management (disk management), Figure 68 on page 87

- Devices Management, Figure 73 on page 91
- Printing Management, Figure 76 on page 93

These interfaces are design to suite the configuration and administration needs of a user managing its own workstation. This user does not have a clear understanding of the underneath AIX architecture but can still administrate it because of the intuitive graphical user interface. The functions covered in each user interface are basicaly the same as what you can do with SMIT but the drag and drop interface make it easiest.

Today, AIX Visual Systems Management GUI does not have distributed facilities.

7.2.3.1 AIX Visual Systems Management GUI Usability

AIX Visual System has a consistent look and feel interface for all the application (users and groups, logical volumes, devices and printing). For instance when you drag a light bulb to a user icon, this means enable a user to login for the Users and Groups Management application. If you drag the same light bulb icon to a filesystem icon in the Logical Volumes Management application, this will mount the filesystem. This light bulb icon always refers to an enable type of action.

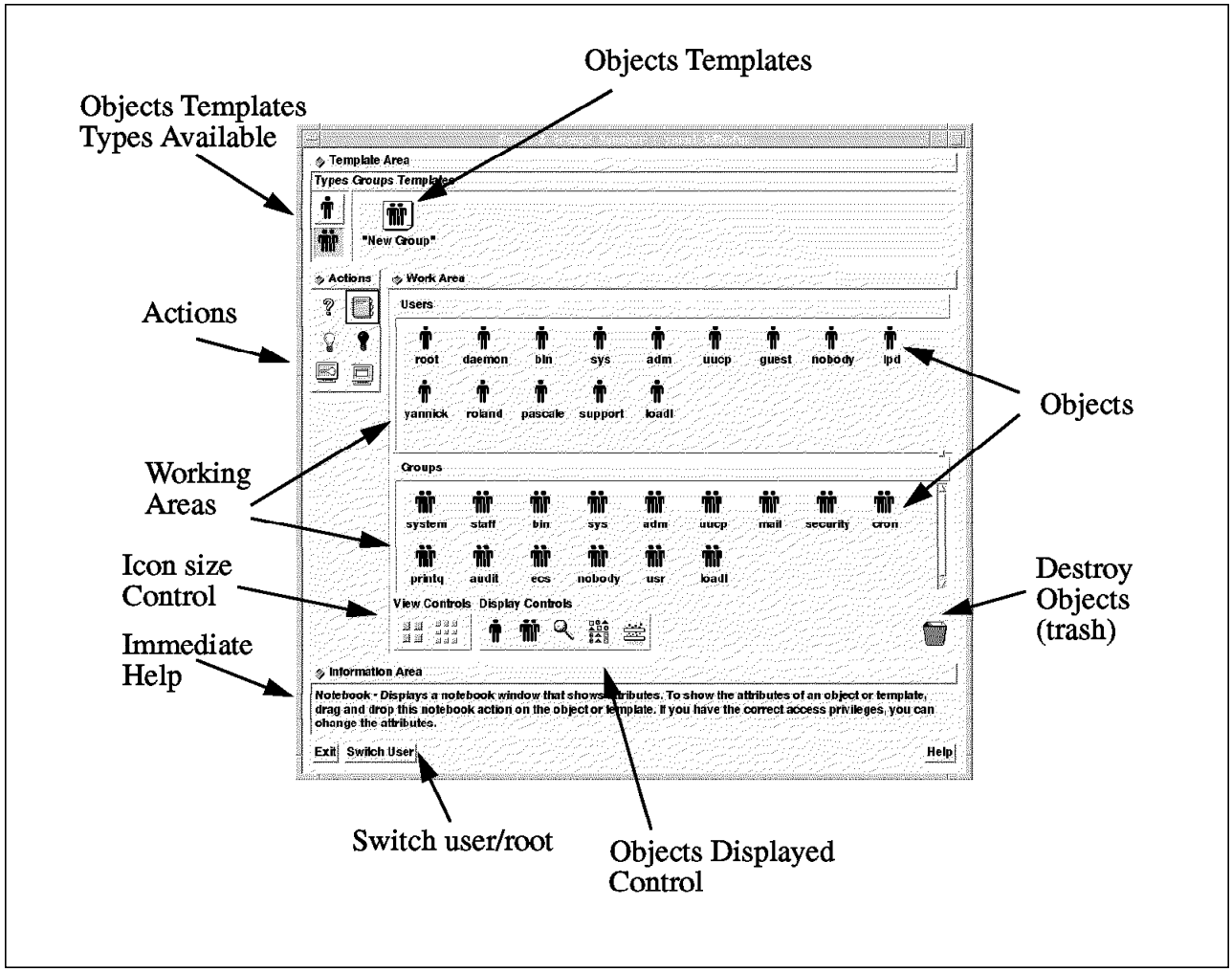

Figure 65. AIX Visual Systems Management Graphical User Interface

In Figure 65, you have a typical AIX Visual Systems Management Graphical Interface main window:

- An Object template is a pre-defined or user-defined set of attributes that you can use to create an object (a user or a group). For instance, you may have two types of users: the secretaries and the developers. A secretary is a member of the staff group, a developer is a member of staff and loadl groups. You can create a secr template, which will have the default group for a secretary, and a dev. template which will have the default groups for a developers. Then, when you have to create a secretary, you take with the mouse the secr user template, drag it to the user work area. The created user is automatically in the right group. The template mechanism works also for creating groups.
- You can click on a Template Type to select within which set of templates you want to work. For instance in Figure 65 on page 83 you can choose between Users templates and Groups Templates. The Group Template type is selected.
- The Actions Icons can be dragged and dropped to an object icon to perform a particular action. For instance in Figure 65 on page 83 you can drag the light bulb icon from the action work area to a user object icon to enable this user to login.
- Working areas is a window which contains a related set of icons.
- The Icon size Control let you choose between small and large icons size for a given work area. This is useful when you have many objects icons.
- When you move your mouse on the AIX Visual Systems Management GUI, you get Immediate help text dynamically updated in the information area at the bottom of the main window. For instance in Figure 65 on page 83 the mouse was on the notebook icon in the action work area, so the help text in the information area is referring to this icon.
- At anytime you can use the Switch User button to change your effective userid from your original userID to root or from root to your original userid. When you go from user to root, you are prompted for the root password. This is useful because you do not need to always run AIX Visual Systems Management GUI from the root user.
- The Display Control lets you setup some filtering and customizing about what objects you want to see. For instance if you have 200 users, you may want to see only the users with a name like ″cicsuser″.
- You can destroy objects by dragging and dropping them into the trash.

You can also see the hierarchy between objects by ″exploding″ the object icon like in Figure 66 on page 85.

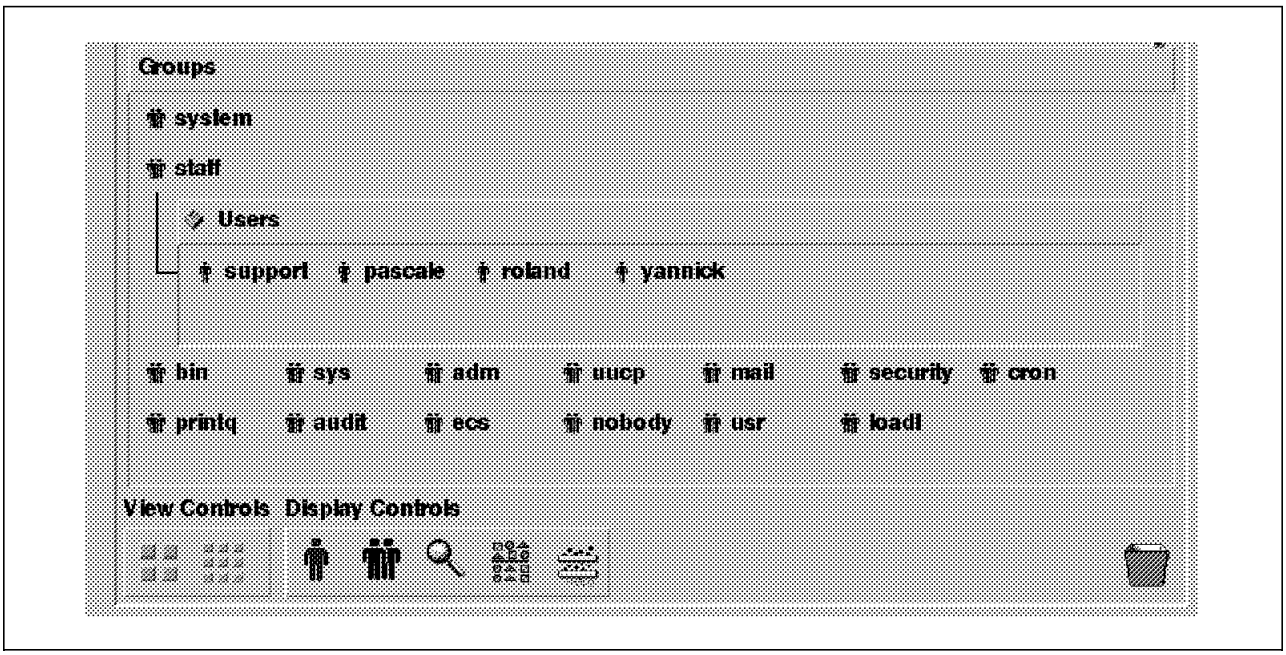

Figure 66. Exploding a Group Icon to See the Users Belonging to this Group

7.2.3.2 AIX Visual Systems Management GUI: Users and Groups Management

This application is started by the `xuserm` command. It contains all the needed operations to manage the users and groups on a local machine. The main functions for Users and Groups Management GUI are:

- Create users and groups using templates.
- Get information about a user or a group.
- Enable a user to log in or disable a user to log in.
- Change the password of a user.
- Change the initial user interface.
- Set the language for messages displayed to a user.

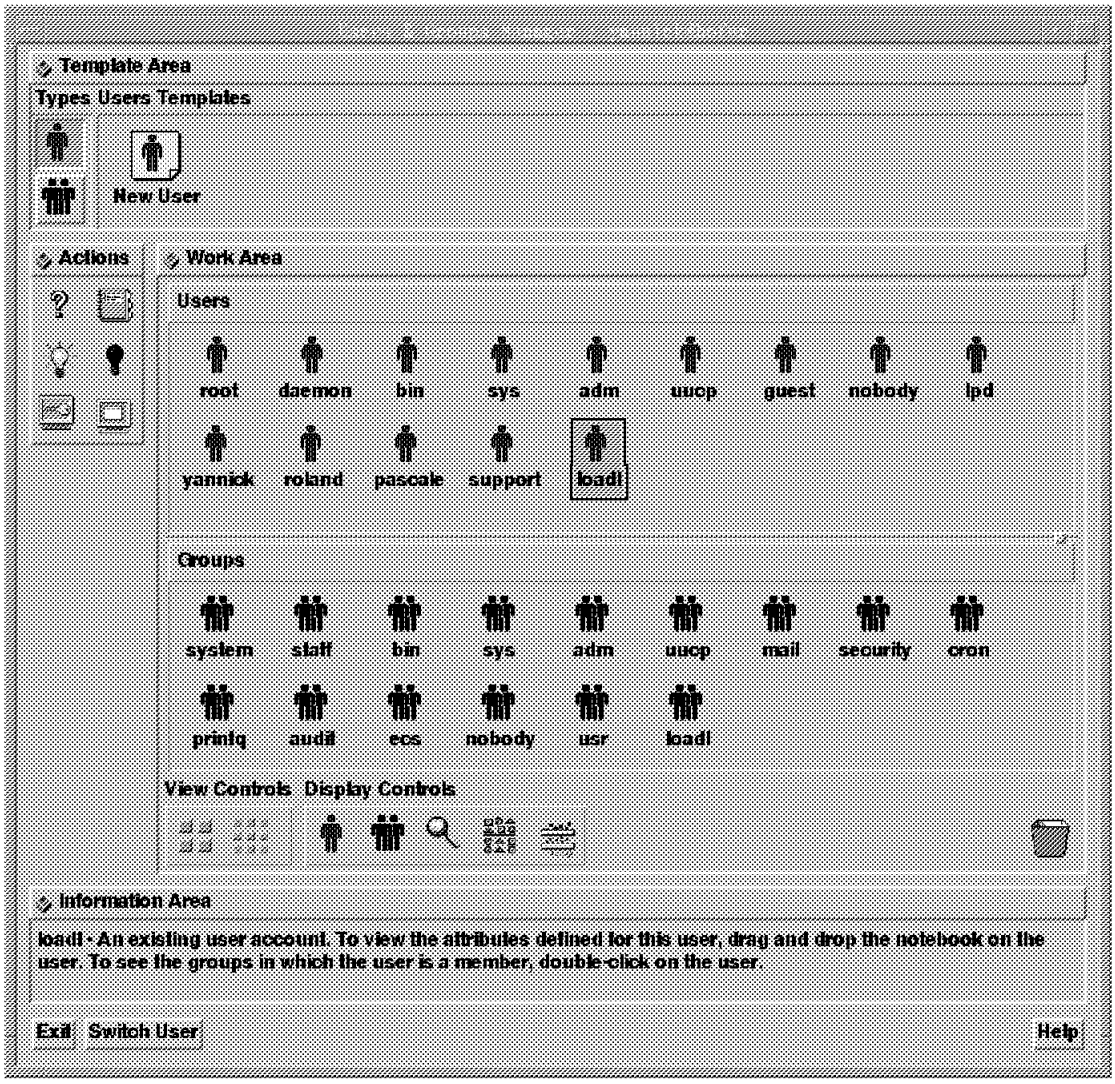

Figure 67. AIX Visual Systems Management GUI for Users and Groups

7.2.3.3 AIX Visual Systems Management GUI: Storage Management

The Storage Management application is started by the `xlvm` command (`lvm` for Logical Volume Manager). The number of operations you can do with this interface is very important. Most of the Logical Volume Manager storage management commands are available here in a drag and drop format (see Figure 68 on page 87).

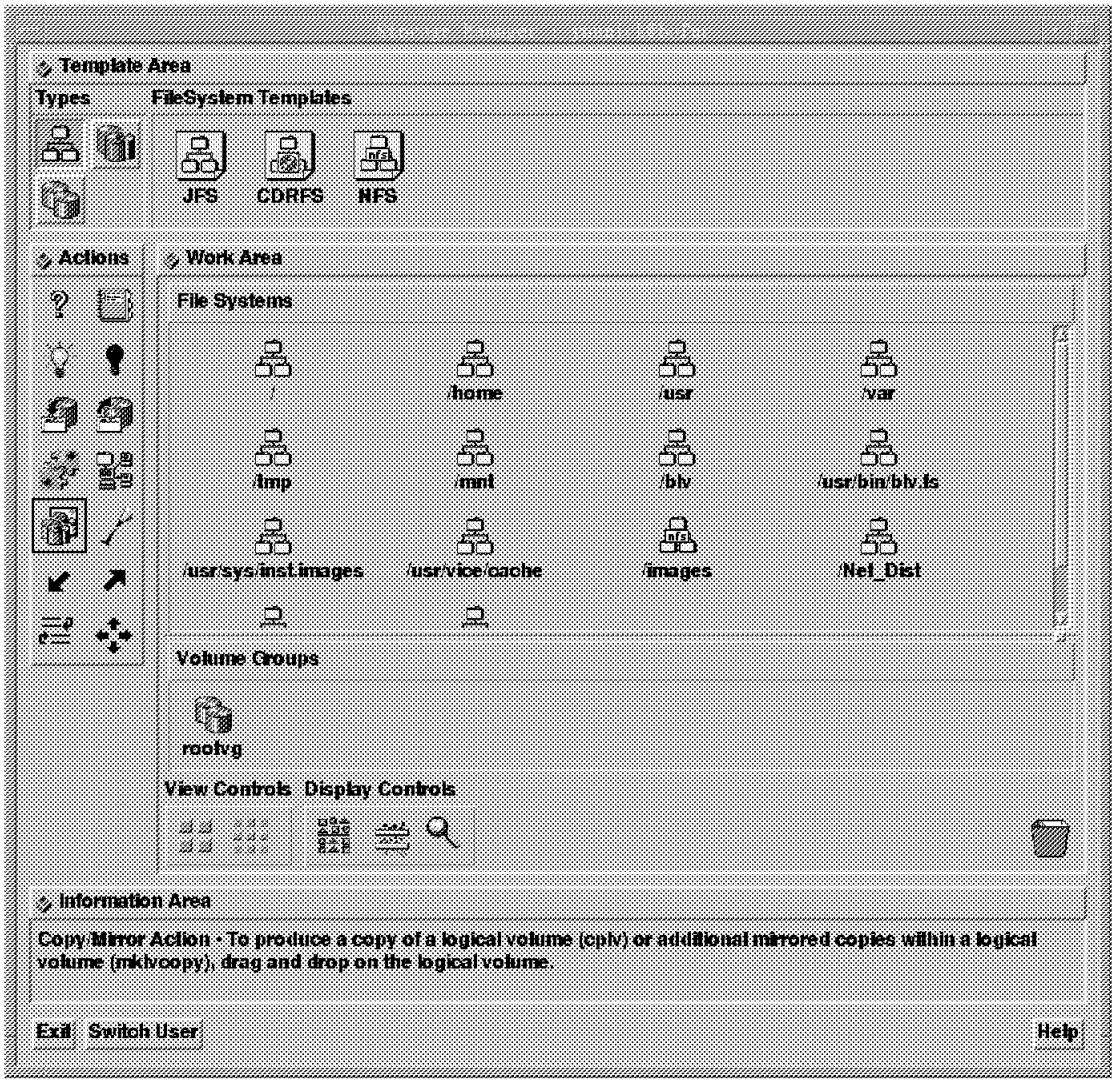

Figure 68. AIX Visual Systems Management GUI for Storage Management

The objects managed by `xlvm` are the classical LVM objects:

- Physical volumes (disks)
- Logical volumes (disks partitions)
- Filesystems (UNIX files in a disk partition)
- Volume groups (set of physical disks)

The hierarchy between those objects could be seen graphically (Figure 69 on page 88).

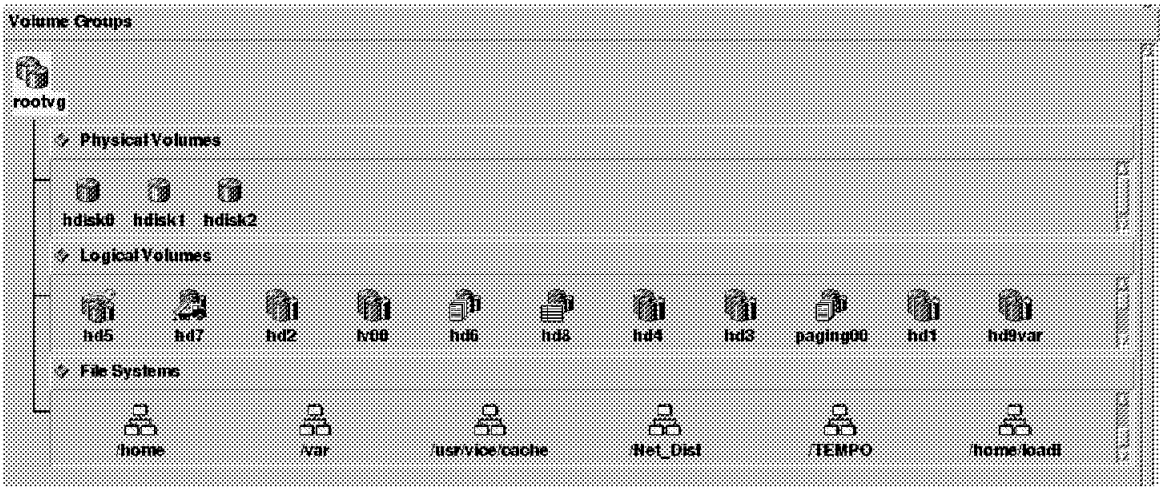

Figure 69. Hierarchy into a Volume Group: Physical volumes, Logical Volume and File

You can also get detail information about detail physical volume usage graphically like in Figure 70.

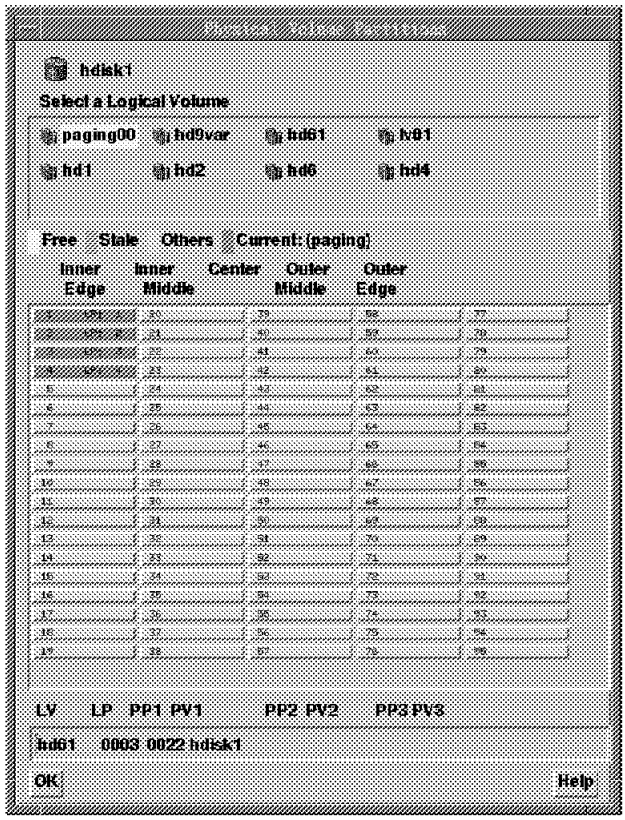

Figure 70. Detail Information about Physical Partition in a Physical Volume

The management actions are made from dragging an action icon from the action work area to one of those objects. In Figure 71 on page 89, you have the available action's icons with a short description of each.

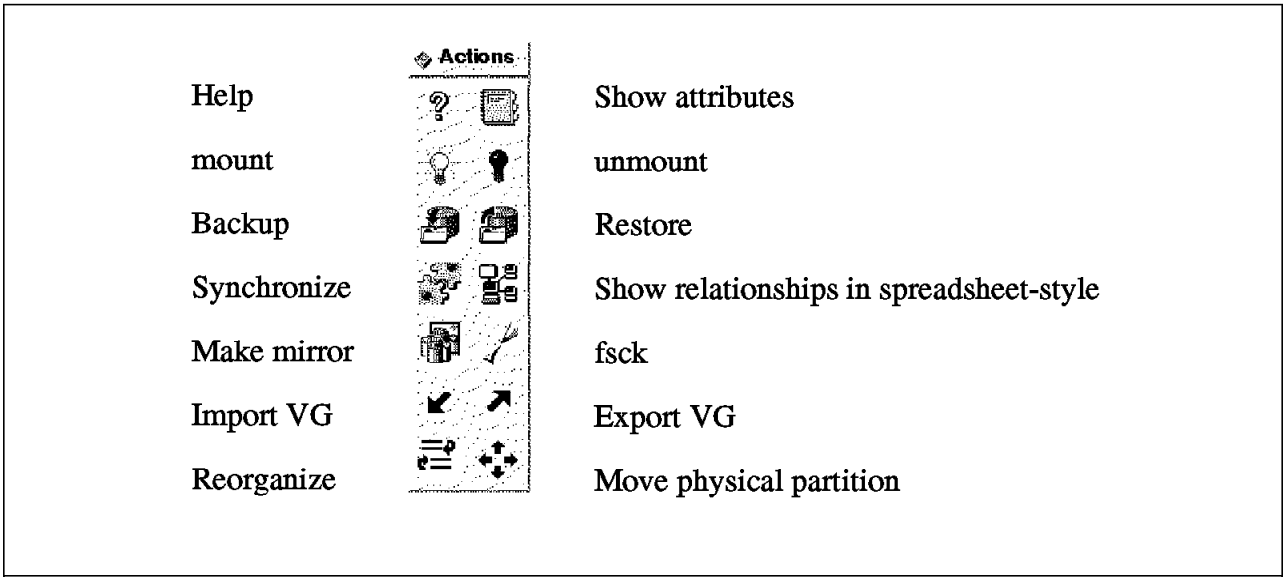

Figure 71. Actions Icons

You notice that some action's icons like backup/restore or synchronize are not LVM operations. For instance, if you drag and drop the backup action icon to a filesystem object icon, you get a panel to start a backup command of this filesystem. If you drag and drop this same backup action icon to the rootvg volume group object icon, then you get a panel to start `mksysb`.

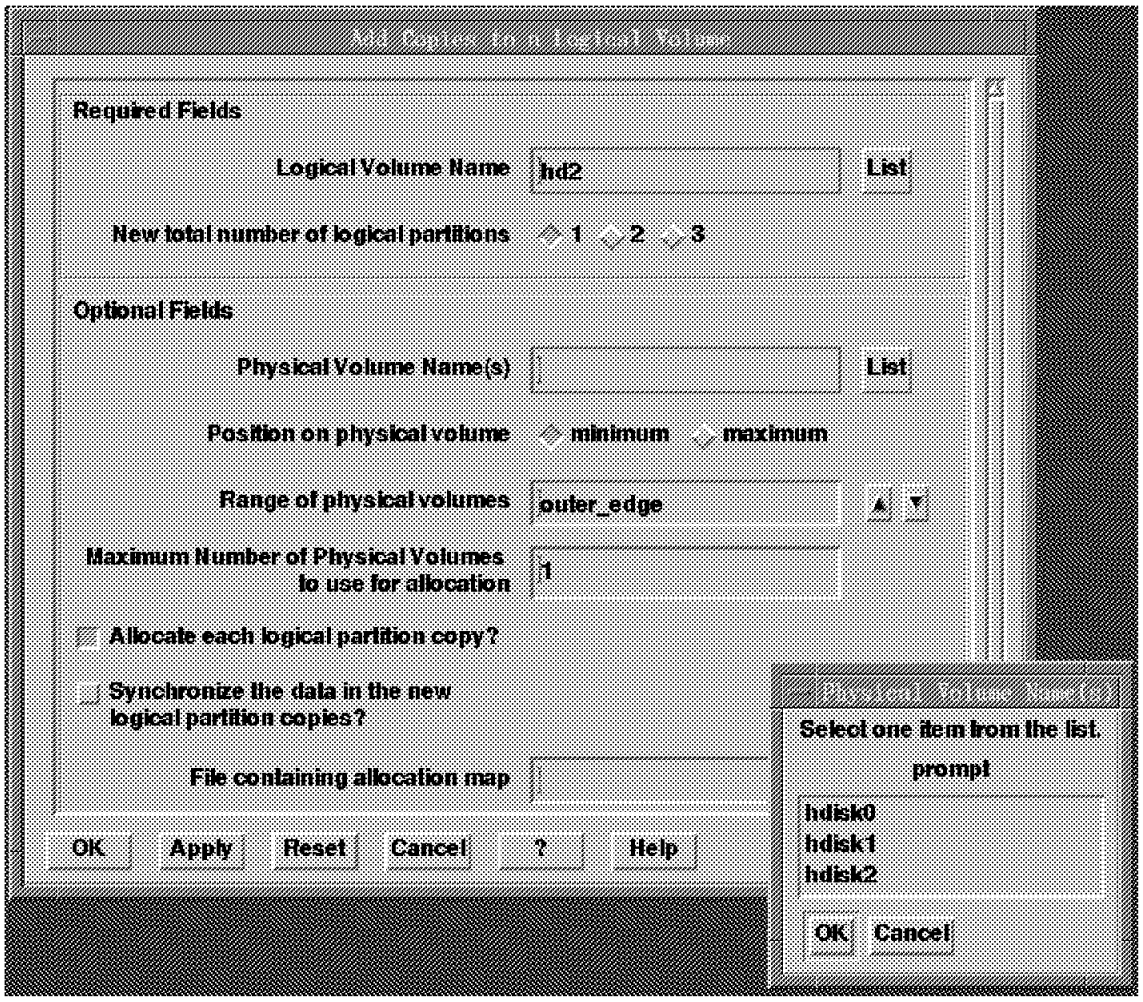

Figure 72. Make a Mirror Copy of a Logical Volume

By dragging and dropping the Make mirror action icon to a logical volume, you easily make an LV mirrored. In Figure 72, you have the panel you get to do that.

7.2.3.4 AIX Visual Systems Management GUI: Devices Management

The devices management application is started with the command (Figure 73 on page 91). The manipulated objects could be physical devices like SCSI devices or logical devices like pseudo-terminals. The look and feel is exactly the same as the previous ones.

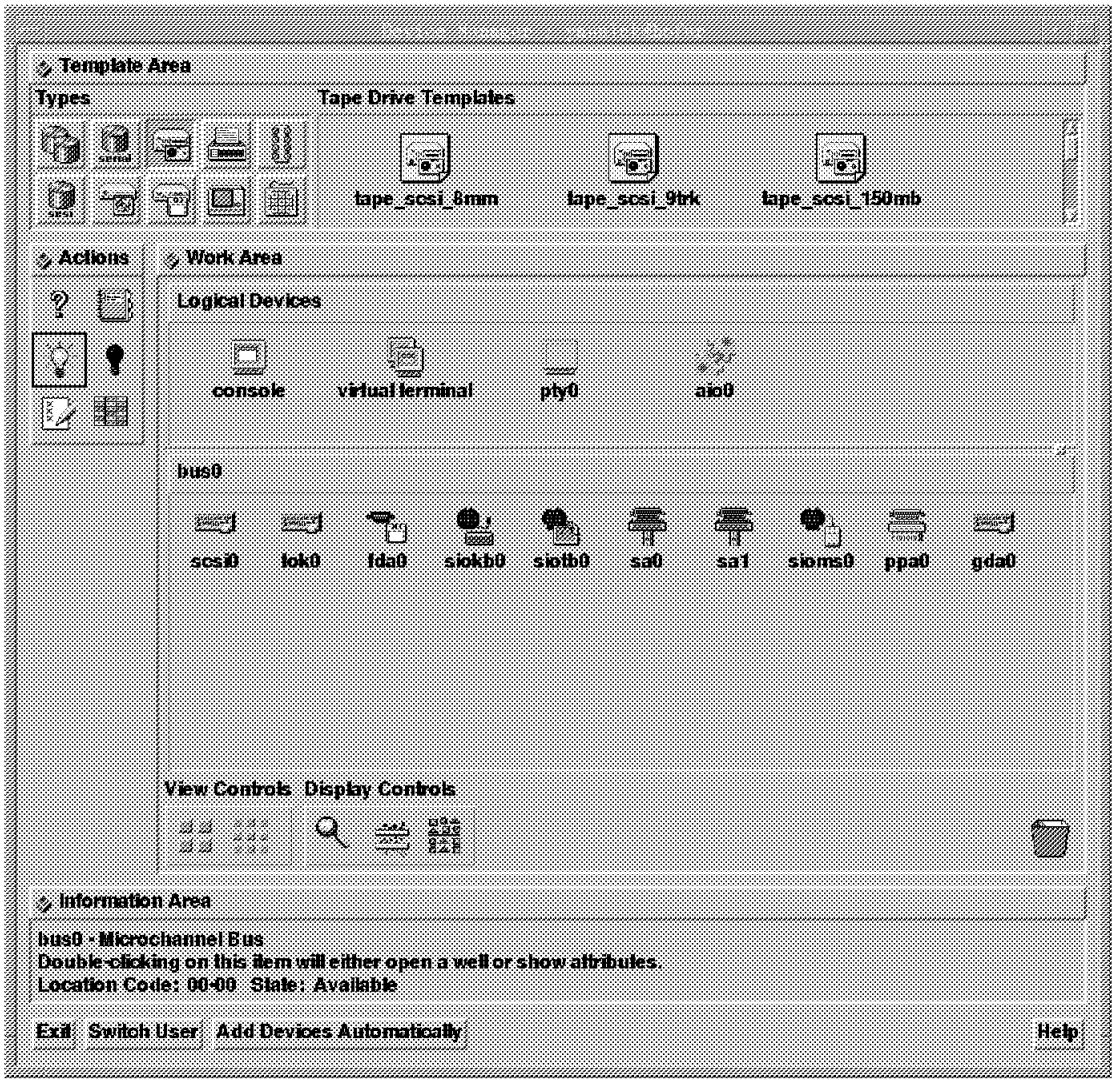

Figure 73. AIX Visual Systems Management GUI for Devices

You can also have a graphical view of the devices hierarchy like the devices on an SCSI bus in Figure 74 on page 92.

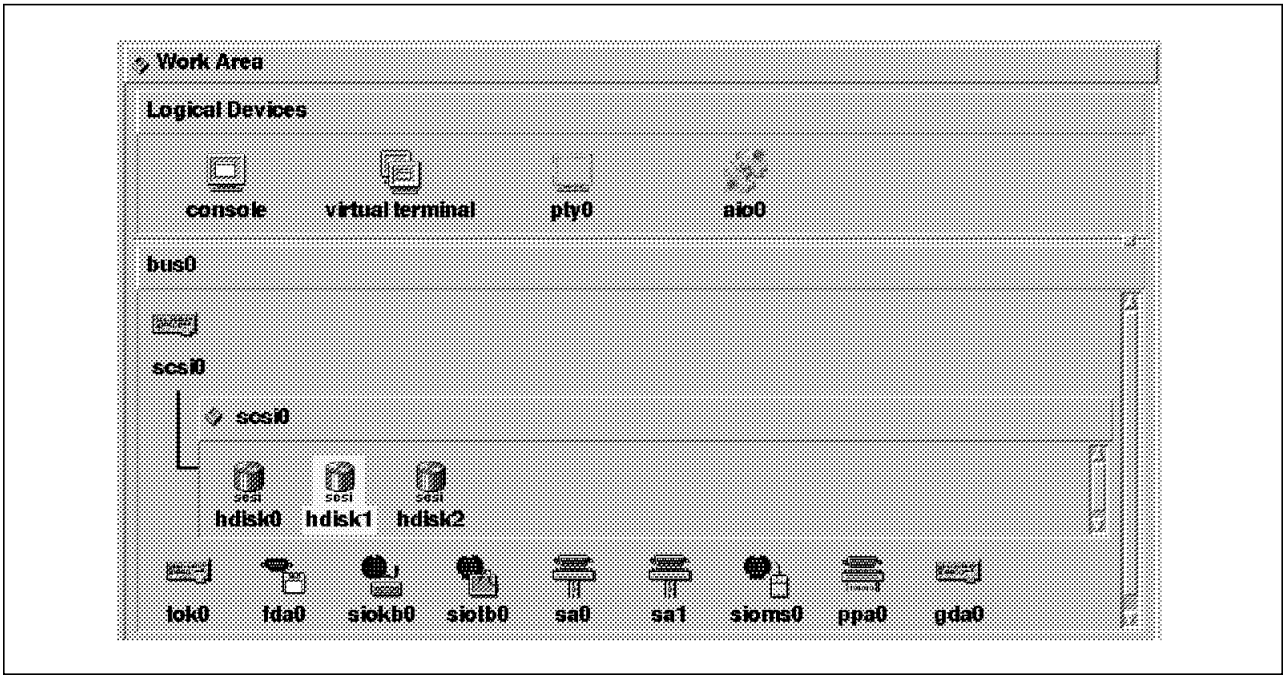

Figure 74. Hierarchy of Devices on the SCSI Bus

You can modify the devices parameters by dragging and dropping the notebook action icon on a device object icon. You can also add new devices using devices templates like in Figure 75.

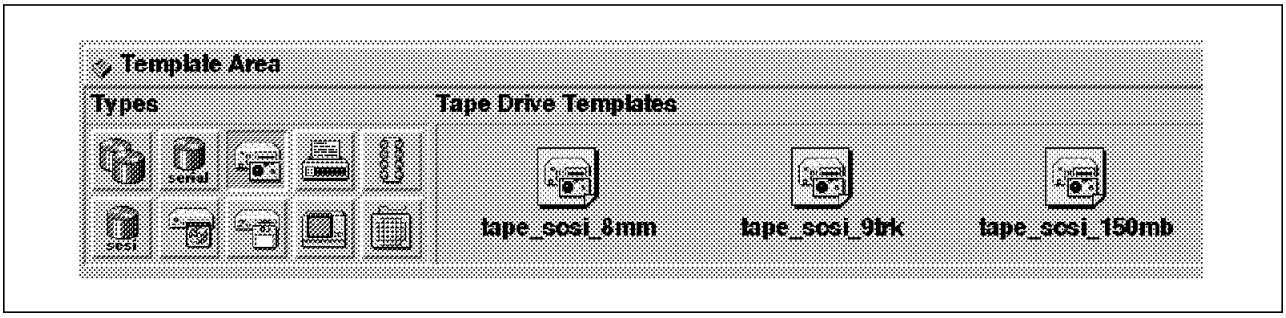

Figure 75. Devices Templates to Add Tape Drive Devices

7.2.3.5 AIX Visual Systems Management GUI: Printing Management

The last application is the printing management, started with the `xprintm` command. This interface let you manage everything about the local printers and the local and remote queues. (see Figure 76 on page 93).

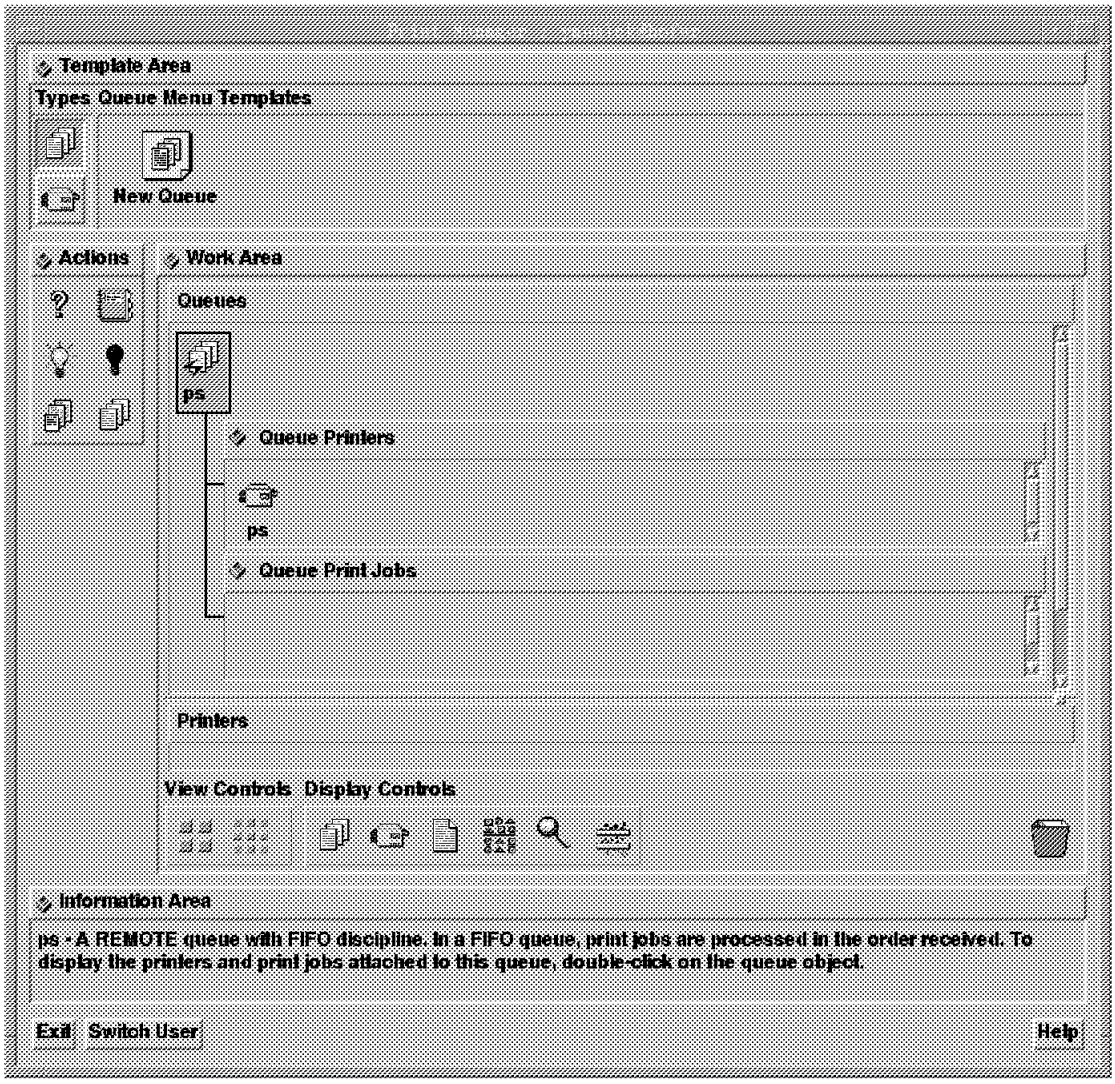

Figure 76. AIX Visual Systems Management GUI for Printing

The functions are equivalent to the printers and queues services available under SMIT, but in a more intuitive drag and drop format

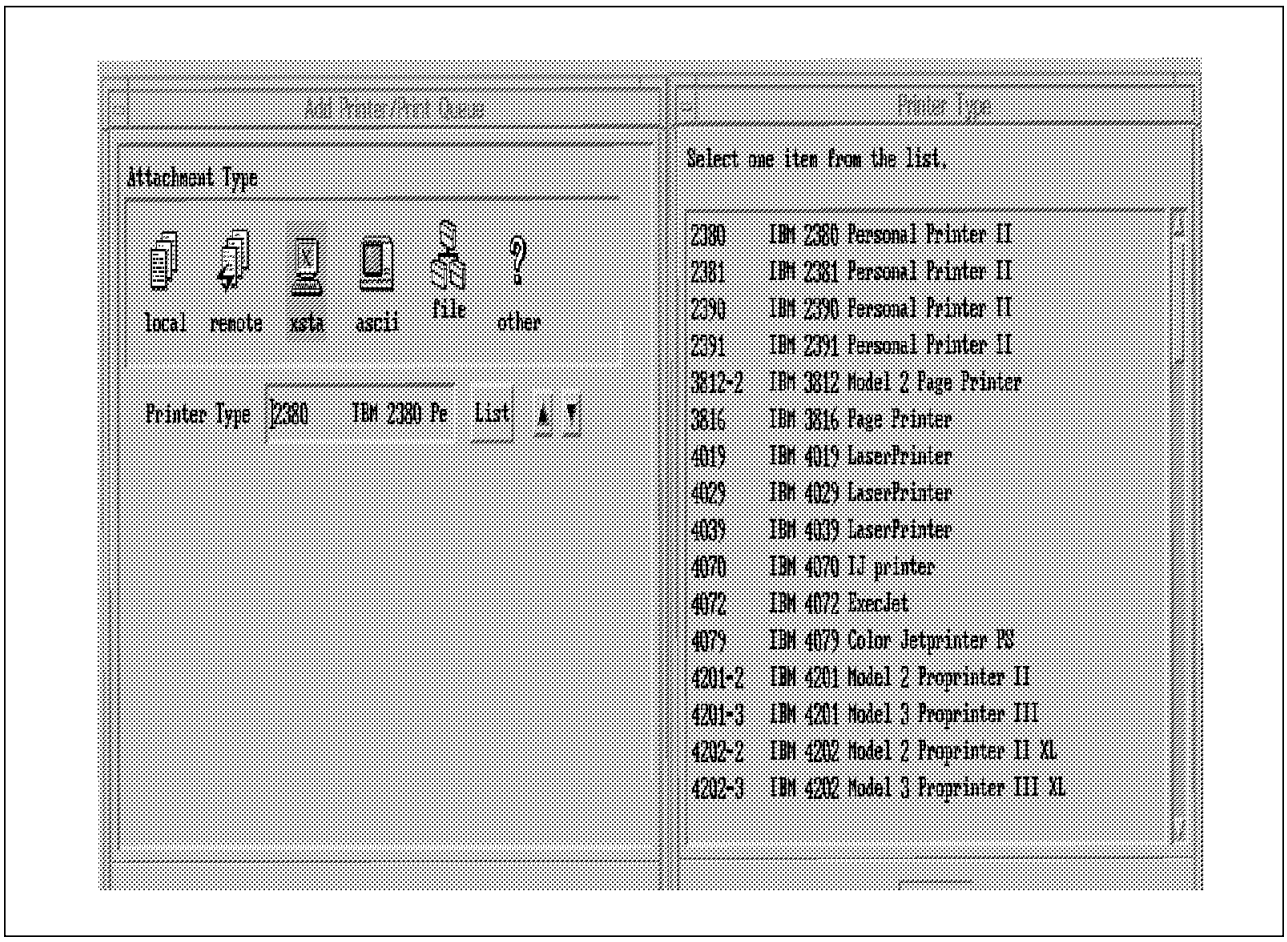

Figure 77. Adding a New Printer

The administrator is assisted in setting up his printers and queues by an easy to understand graphical hierarchy view of what he is doing. For instance in Figure 77 and Figure 78 on page 95 you can see the panel you get to add a new local printer.

This is easy to use because the steps needed to add this printer are logical: you drag and drop the new printer template to the printer work area, you select your attachment type ((local, remote, Xstation, ASCII terminal connected, to a file or other), you select the printer type, then (for local printers) you choose your printer interface (parallel, rs232, rs422), the ports and finally configure the physical device itself (baud rates) all this in one logical suite of operations.

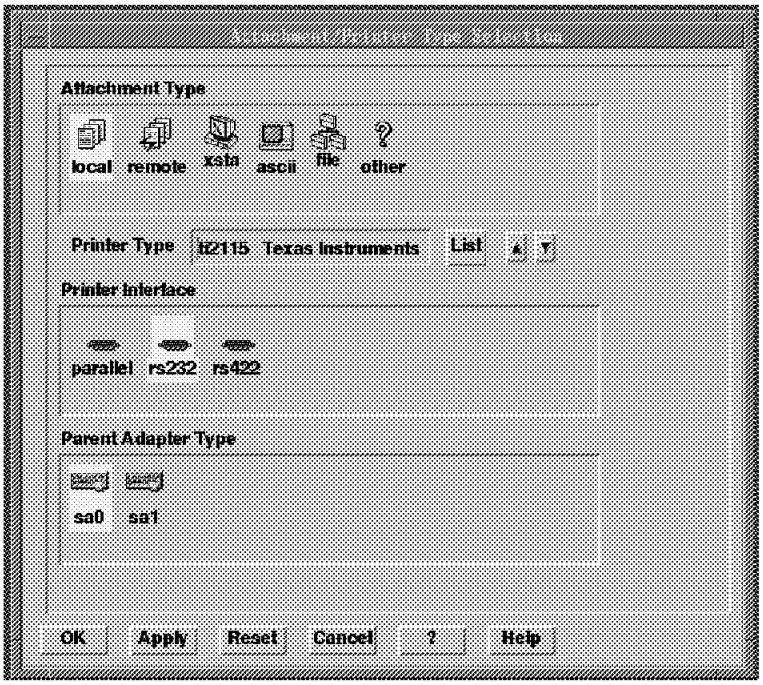

Figure 78. Choosing the Interface for a Local Printer

The `xprintm` command can help you for more than configuration. You can also use it for monitoring and managing living queues (Figure 79 on page 96). For instance, if you want to cancel a printing job from a queue, you just need to drag and drop this job icon from the queue to the trash icon.

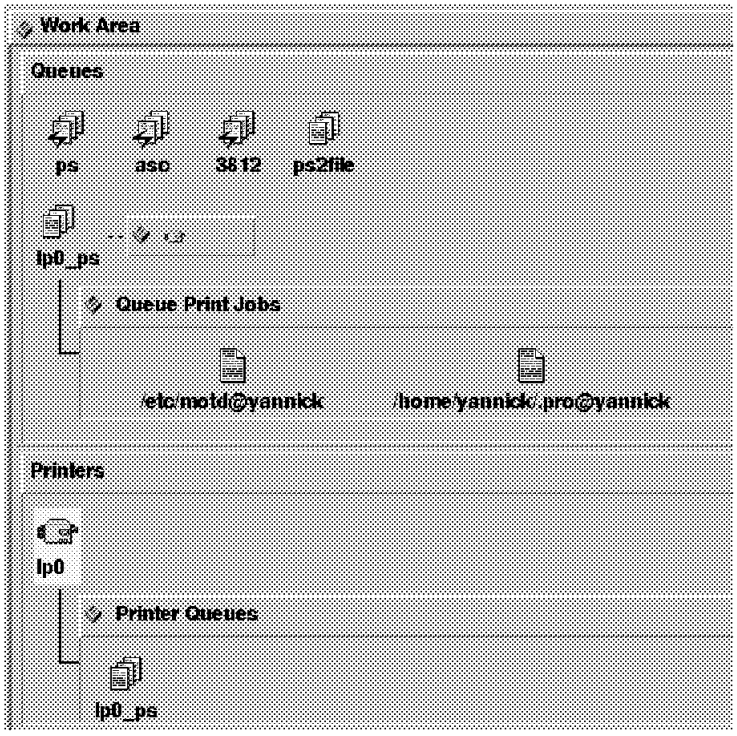

Figure 79. Managing Printer Queues and Printer Jobs in those Queues

7.3 Distributed SMIT

Distributed System Management Interface Tool (DSMIT) is a new product available as an LPP since January 1994. DSMIT gives you extensions to the SMIT management interface for distributed and heterogeneous support.

With DSMIT, you can perform a traditional SMIT operation like add a user, not only on the local machine but to a set of several RISC System/6000, SUN or HP machines. This operation could be performed on all the machines simultaneously or one machine at a time. For instance, you can add the user with the same userid to 50 machines in three operations: one for 30 RISC System/6000s, one for 10 SUNs and one for 10 HPs.

Today, DSMIT only supports an ASCII interface. There is no Motif DSMIT (see Figure 80 on page 97).

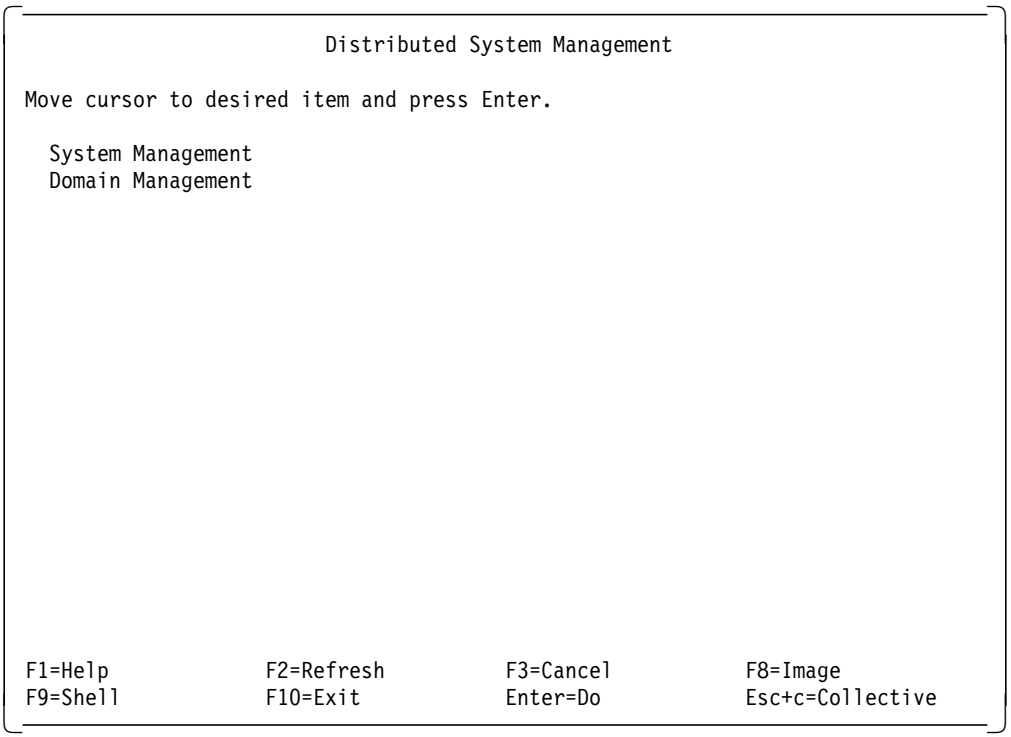

Figure 80. DSMIT User Interface

7.3.1 DSMIT Domains and Working Collective

DSMIT works as a client/server. For DSMIT, the server is the user interface where you prepare a DSMIT task and the clients the code on the target machines that will finally locally execute the task. Both client and server code can be installed by `smit installp`. A machine could be client or server, or client and server. A client only machine can just execute tasks. A server only machine can just prepare and send task requests.

Warning: DSMIT uses the same level of security as that provided by TCP/IP `rsh` commands. Usage of DSMIT client support will subject the system to the potential risk of unauthorized access. DSMIT is not a trusted environment because the DSMIT client must specify the DSMIT server host name in the /.rhosts and /etc/hosts.equiv files.

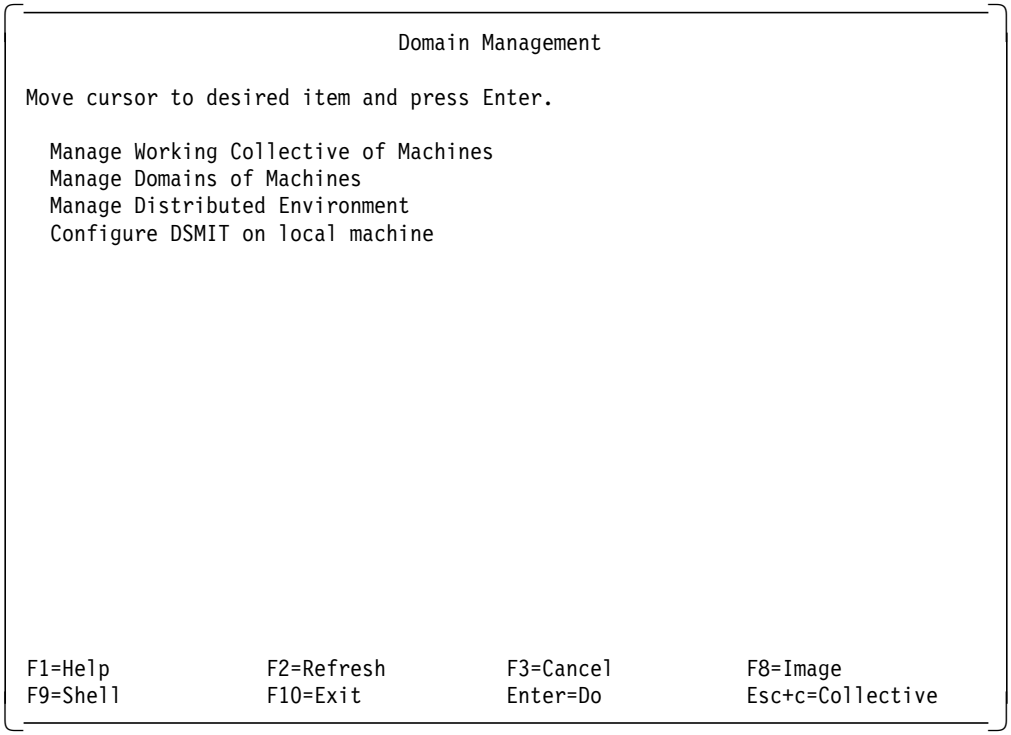

Figure 81. Domain Management Panel of DSMIT

Before using DSMIT for a distributed systems management task, you need to define on what Working Collective you are working. A Working Collective is the sub-set of machines that is currently selected to the one where the task you are requesting will be executed. The default Working Collective is the local machine.

You can also define domains of machines. A domain is a list of machines identified by a name. A given machine can belong to several domains.

```
                                COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

Machine         Op. system  Domains
-------         ----------  -------
bob1            AIX_3.2     domain2
bofur           AIX_3.2     domain2,sysmag
bruce           AIX_3.2     domain1,domain2,sysmag









F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit
```

Figure 82. List of the Machines with their Domains and Operating Systems

From most of the DSMIT panels, you can call a Working Collective Include/Exclude panel with the Escape+C sequence. This let you include or exclude machines from your current Working Collective with search criteria like machine, domain or operating systems names (see Figure 83).

```
              Include/Exclude Machines from the Working Collective


Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                  [Entry Fields]
  Current Inclusion/Exclusion Status                                  +
  Current Inclusion/Exclusion Status                                  +

* INCLUDE or EXCLUDE machines from the Collective?    include        +
* INCLUDE or EXCLUDE machines from the Collective?    include        +
  Specify ma┌
  Specify ma│          Current Inclusion/Exclusion Status          │
    MACHINE │                                                      │  +
    MACHINE │ The List is for Display Only.                        │  +
    OPERATIN│                                                      │  +
    OPERATIN│   Machine         Inc/Ex  Op. system                 │  +
    DOMAIN N│   -------         ------  ----------                 │  +
    DOMAIN N│   bofur             +     AIX_3.2                    │  +
[MORE...2]  │   bruce             +     AIX_3.2                    │
            │                                                      │
F1=Help     │                                                      │
F5=Undo     │ F3=Cancel                                            │
F9=Shell    └
```

Figure 83. Managing your Current Working Collective with the ESC+C Panel

7.3.2 Example of Using DSMIT

In this example we will cancel two printer jobs from two different queues on two machines from DSMIT. The two machines must be your Working Collective (see Figure 83 on page 99).

```
                              Spooler (Print Jobs)

Move cursor to desired item and press Enter.

  Start a Print Job
  Cancel a Print Job
  Show the Status of Print Jobs
  Prioritize a Print Job
  Schedule Jobs
  Manage Local Printer Subsystem
  Manage Remote Printer Subsystem

F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do            Esc+c=Collective
```

Figure 84. DSMIT Panel for Managing the Print Jobs (idem to the same SMI T panel)

The System Management screens tree is identical to the traditional SMIT screens tree (see Figure 63 on page 81 and Figure 80 on page 97). From the first DSMIT menu, select:

- **System Management**, Figure 80 on page 97
- **Spooler (Print Jobs)**, Figure 84
- **Cancel a Print Job**, Figure 85 on page 101

```
                     Common Dialogue for Cancel a Print Job

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                           [Entry Fields]
* PRINT QUEUE where job resides                         []             +
* JOB NUMBER                                            []             +
    (Note:  There may be a slight delay in
     displaying a list of remote jobs.)











F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Undo             F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do            Esc+m=By Machine
Esc+f=By Field      Esc+c=Collective
```

Figure 85. DSMIT Panel to Cancel Print Jobs

If you were in an usual SMIT panel, you will select a print queue and a print job from the local machine, maybe using the F4=List function key. If you want to select queues and jobs from all the machines in your current working collective, you use the ESC+f sequence for every field.

```
                         PRINT QUEUE where job resides

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                           [Entry Fields]
* bofur                                                 []             +
* bruce                                                 []             +


                         PRINT QUEUE where job resides

    Move cursor to desired item and press Enter.

      asc
      ps
      3812
      test
      remq

    F1=Help                 F2=Refresh              F3=Cancel
F1  F8=Image                F10=Exit                Enter=Do
F5  /=Find                  n=Find Next
F9
```

Figure 86. Choosing Queue Names on the Machine bofur and the Machine bruce

For instance, if your cursor is on the PRINT QUEUE where job resides field of Figure 85 then you get the panel Figure 86. In this panel, you can choose, using the F4=List function key, a queue on the machine bofur and a queue on the machine bruce.

```
                                JOB NUMBER
                      (Note:  There may be a slight delay in
   ┌
   │                                JOB NUMBER                                │
Ty │                                                                          │
Pr │ Move cursor to desired item and press Enter.                             │
   │                                                                          │
   │   Queue     Job Files              User                                  │
*  │   -------   --- ------------------ ----------                            │  +
*  │   ps                                                                     │  +
   │   ps                                                                     │
   │   asc                                                                    │
   │   asc                                                                    │
   │   3812                                                                   │
   │   3812                                                                   │
   │   ps2file                                                                │
   │   lp0_ps                                                                 │
   │   lp0_ps     79 /etc/motd          yannick                               │
   │   lp0_ps     81 /home/yannick/.pro yannick                               │
   │   lp0_ps     82 /home/yannick/xuse yannick                               │
   │                                                                          │
   │ F1=Help                F2=Refresh             F3=Cancel                  │
F1 │ F8=Image               F10=Exit               Enter=Do                   │
F5 │ /=Find                 n=Find Next                                       │
F9 └
```

Figure 87. Choosing Jobs Numbers in the Two Queues on Machine bofur and Machine bruce

You also do the same operation for the JOB NUMBER field of Figure 85 on page 101.

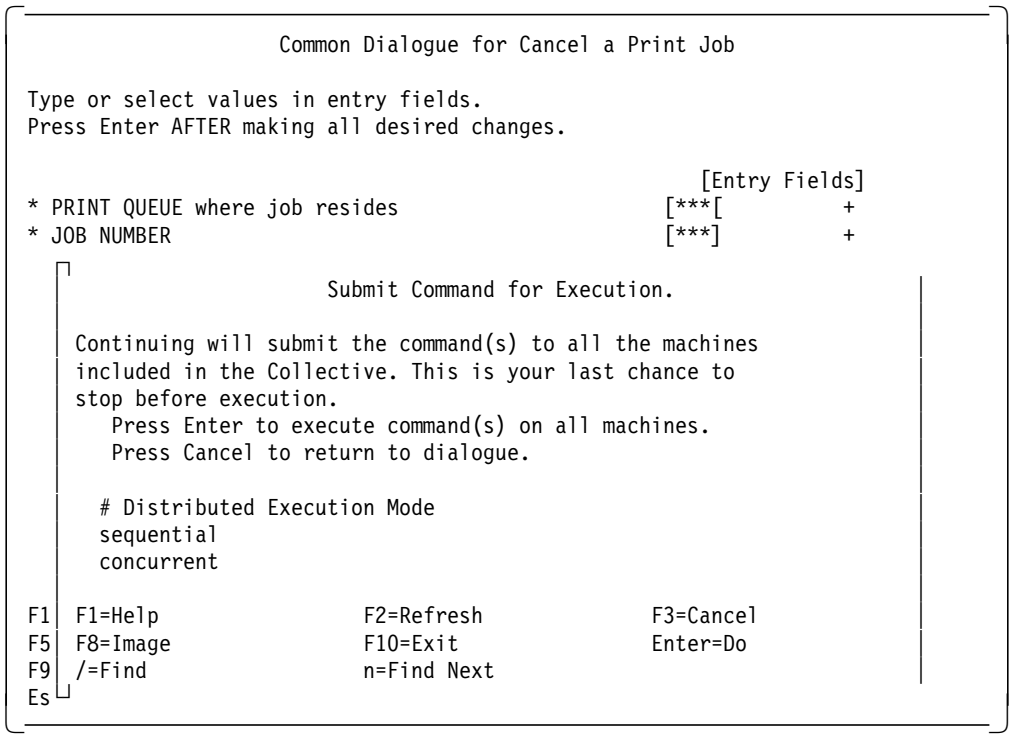

Figure 88. Running the Final Cancel Task

When both distributed selection had been made for the two fields, you run the Cancel a Print Jobs task pressing Enter. DSMIT asks you if you want to run the task in sequential order (one machine at a time) or concurrent order (all the machines at the same time).

Notice in Figure 88 that a field that has a distributed selection in it is displayed with asterisks.

7.4 Installation Assistant

On the AIX Version 4.1, the Installation Assistant tool, simplifies customizing and managing your system by guiding you through post-installation tasks and, in some cases, automatically installing software packages for you. In addition, the Installation Assistant introduces you to various interfaces, providing only the help you need, when you need it.

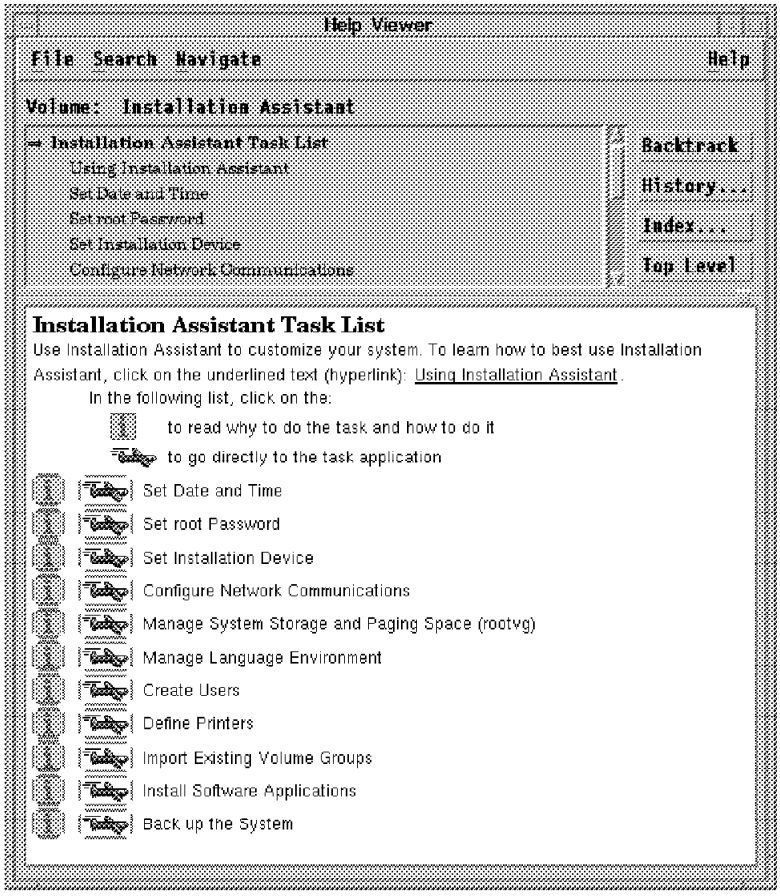

Figure 89. Installation Assistant Main Panel

Figure 89 shows you the Installation Assistant Graphical User Interface. Two paths are provided :

- ***Learning Path***: Clicking on the i icon gives you information on why you would want to do a task, what the prerequisites are, and how to complete a task.
- ***Fast Path***: If you are already familiar with the tasks and know how to use SMIT and Visual Systems Management, then click on the airplane icon to skip the instructions and go directly to the task application.

In the Installation Assistant Task List of Figure 89, clicking on a fast path runs AIX commands, SMIT or Visual Systems Management on the task application. For example, if you activate the Configure Network Communications fast path button, you will get the SMIT panel on TCP/IP and NFS. Information on how to configure TCP/IP guides you after you have activated the learning path button on the same task (see Figure 90 on page 105).

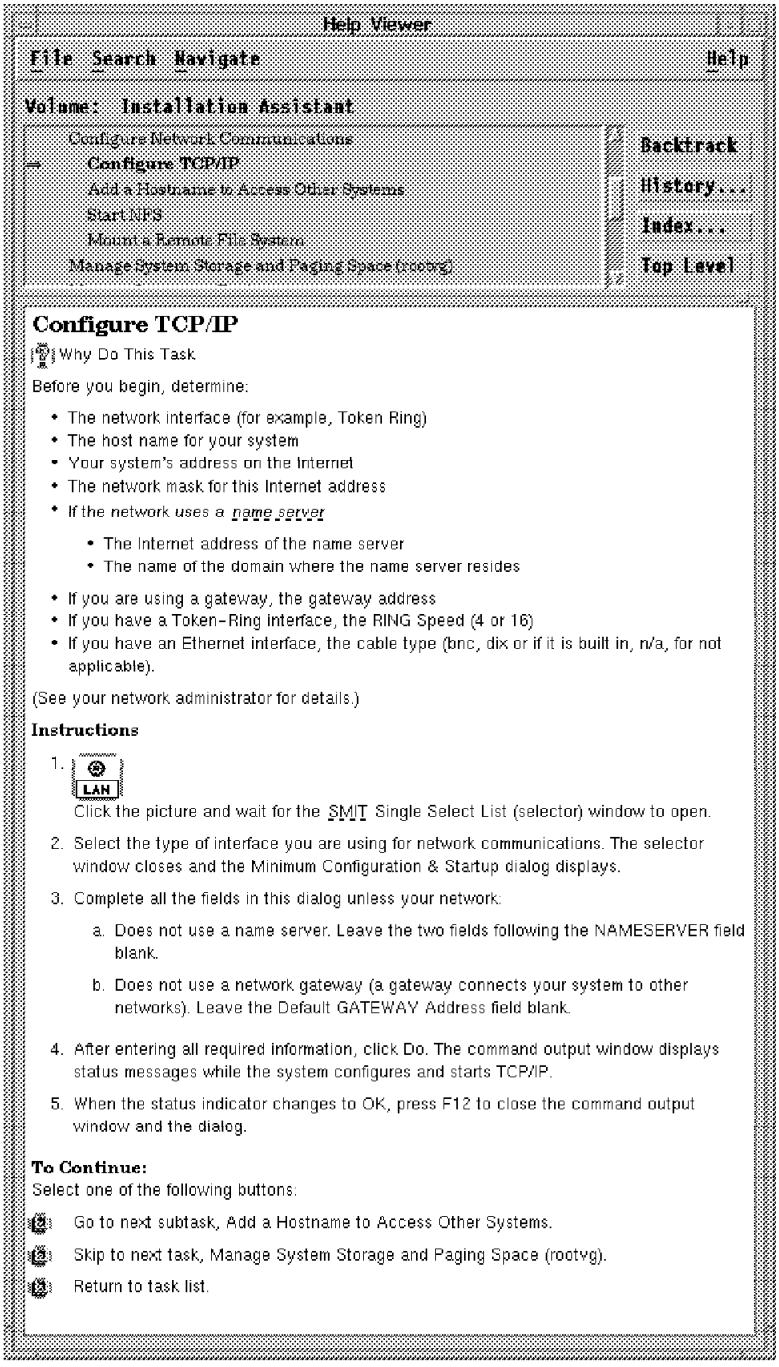

Figure 90. Configuring TCP/IP with Installation Assistant

Another example of using Installation Assistant is the Define Printers task: the fast path runs the Visual Systems Management Print Manager, which was discussed in the section 7.2.3.5, "AIX Visual Systems Management GUI: Printing Management" on page 92. Figure 91 on page 106 shows you the panel you get when you activate the fast path icon on the task.

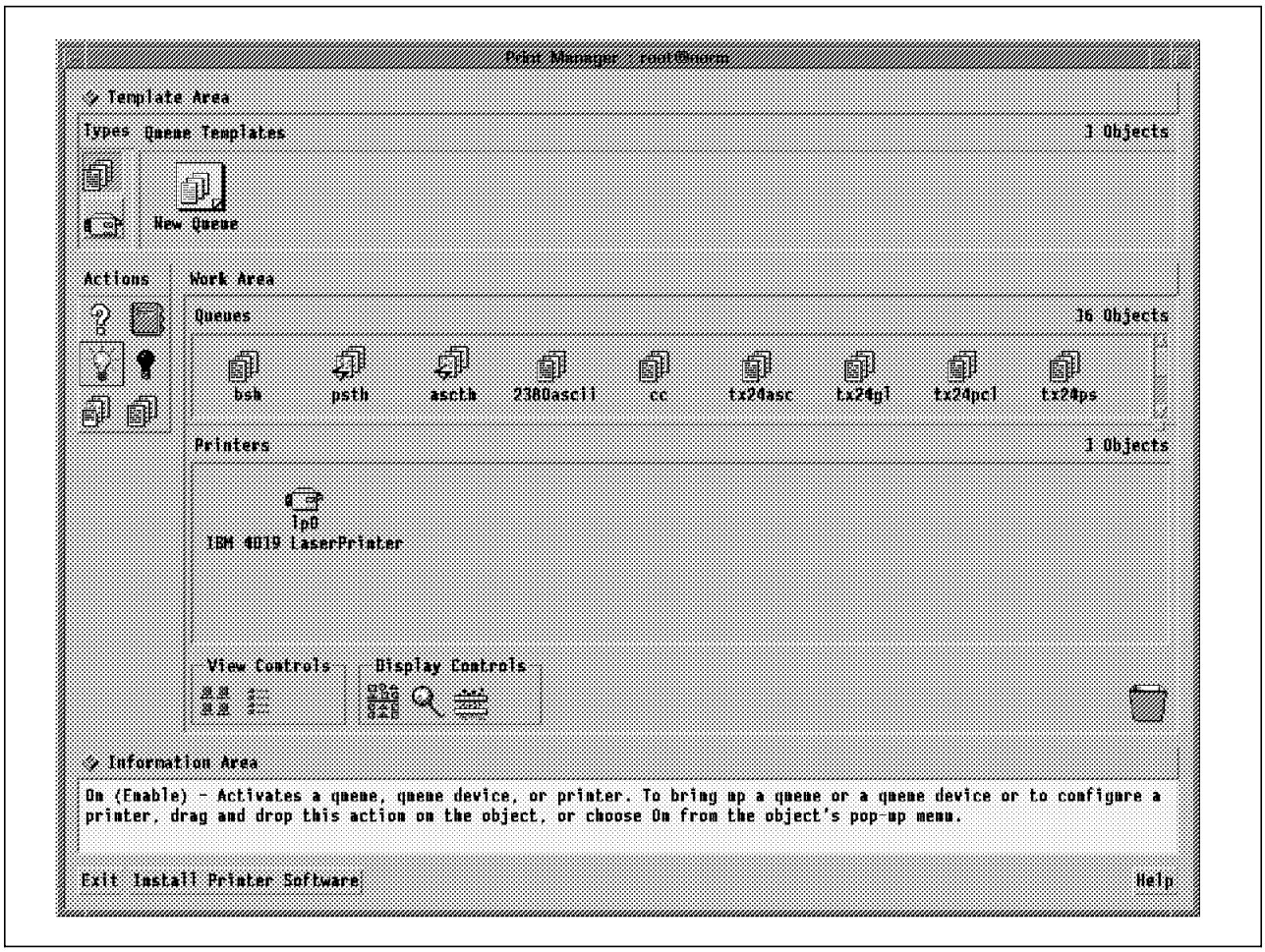

Figure 91. Visual Systems Management Print Manager Run by Installation Assistant

Chapter 8. Performance Measurement

Since the satisfaction of users is highly dependent on response time, it is necessary to periodically track system performance. This activity provides information about the current state of your system, and allows you to respond proactively to potential performance problems. In this chapter you will get an overview of various commands and tools that are available for tracking system performance.

8.1 Considerations

To be prepared to handle performance problems, you have to measure and understand the workload of your system at a time when all users are satisfied.

Basically you need to collect two types of information:

- Which user's processes use the most system resources?
- What is the normal workload of the system?

System accounting allows you to record the system usage by users, processes, and system devices. This gives you an understanding about the behavior of these processes and shows you their normal usage. When performance problems arise, you can tell whether a process is behaving correctly or not. You can find detailed information about the customization of system accounting in the *InfoExplorer* database.

By using various commands and tools, which are described later on, you can get information about system resources such as CPU, memory, paging space, I/O and so on. By periodically monitoring these values, you will be able to recognize performance problems before they turn into user complaints.

In the event of performance problems, you must first identify where the bottleneck is and then determine the reason for it.

The overall performance of the system for all users is dependent on three fundamental resources:

- CPU
- Memory subsystem
- I/O subsystem (disks, network, terminals)

In general, performance problems arise from two types of situation:

- The system is simply overloaded (users, applications). The only solution in this case is to add hardware (faster CPU, more memory, disk, and so on).
- A process, a user, or an application is making abnormal use of system resources. This can sometimes be fixed very easily, but can also require a redesign of your application.

© Copyright IBM Corp. 1993,1994

8.2 Standard Commands

There are several standard UNIX commands available for performance measurement. We will just list them here; for a detailed description of their usage, see *InfoExplorer*, or the *AIX Version 3.2 Commands Reference*.

- `ps` - shows the current status of processes
- `sar` - collects and reports system activity information
- `vmstat` - reports virtual memory statistics
- `iostat` - reports CPU and I/O statistics
- `timex` - reports the user, elapsed and system time for an executable

8.3 AIX Tools

In addition to the standard commands, AIX provides other tools to simplify performance measurement.

- `filemon` - file system activity monitor
- `fileplace` - displays the placements of file blocks
- `lvedit` - provides control over the creation and placement of logical volumes on physical volumes
- `netpmon` - for monitoring of network traffic
- `trace` - system and program event trace
- `trcrpt` - displays listings of the trace log
- `rmap` - extended analysis for trace log data
- `rmss` - reduced memory system simulator
- `svmon` - displays a snapshot of real and virtual memory
- `tprof` - provides a detailed profile of CPU usage by an application

These commands are provided with the performance monitoring tools enhancement. For detailed information check the README file in the /usr/lpp/bosperf directory.

8.4 Performance Toolbox - Performance Aide for AIX

The products, *Performance Toolbox Version 1.2 for AIX* and *Performance Aide Version 1.2 for AIX*, provide graphical views of system performance, and operate in concert with other performance tools (such as trace facility commands, and the `nfsstat`, `netstat`, `iostat`, `vmstat`, `pstat`, `no`, `sar` and `timex` commands). The two products are implemented through a client/server architecture, allowing centralized performance monitoring of networked workstations.

In addition, Hewlett-Packard′s HP 9000/700 and HP 9000/800 machines or Sun Microsystem′s SunSPARC (SunOS Version 4.1.3) machines, connected via TCP/IP, can be monitored from a single PTX console. For that, you have to install the xmservd agent (PAIDE) on these machines.

8.4.1 General Organization

Performance Toolbox for AIX (PTX), the manager component, provides a graphical view of system performance statistics, and access to a set of performance tools via a system of menus.

PTX consists of several performance programs packaged together. It provides a toolbox framework for performance management tools that can be used on a single system or in an Internet (IP) Network environment. This facility can augment the information gathered from a higher level Simple Network Management Protocol (SNMP) network manager.

PTX includes X and Motif-based applications that provide realtime color graphic performance monitors for local and remote systems, performance analysis tools, and performance tuning controls.

Performance Aide for AIX (PAIDE), the agent, includes the server daemon that feeds the manager component with the performance statistics of your local or remote system.

PAIDE has facilities for concurrently servicing multiple data requests from local or remote applications. It can do local data filtering and alert processing. Additionally, it can provide data to the SNMP agent on the local node.

PAIDE for AIX is a prerequisite for PTX for AIX. It is included as part of PTX, but can also be ordered separately. You must order and install one copy of PAIDE for AIX for each node being monitored.

In Figure 92 you can see the general organization of Performance Toolbox and Performance Aide for AIX.

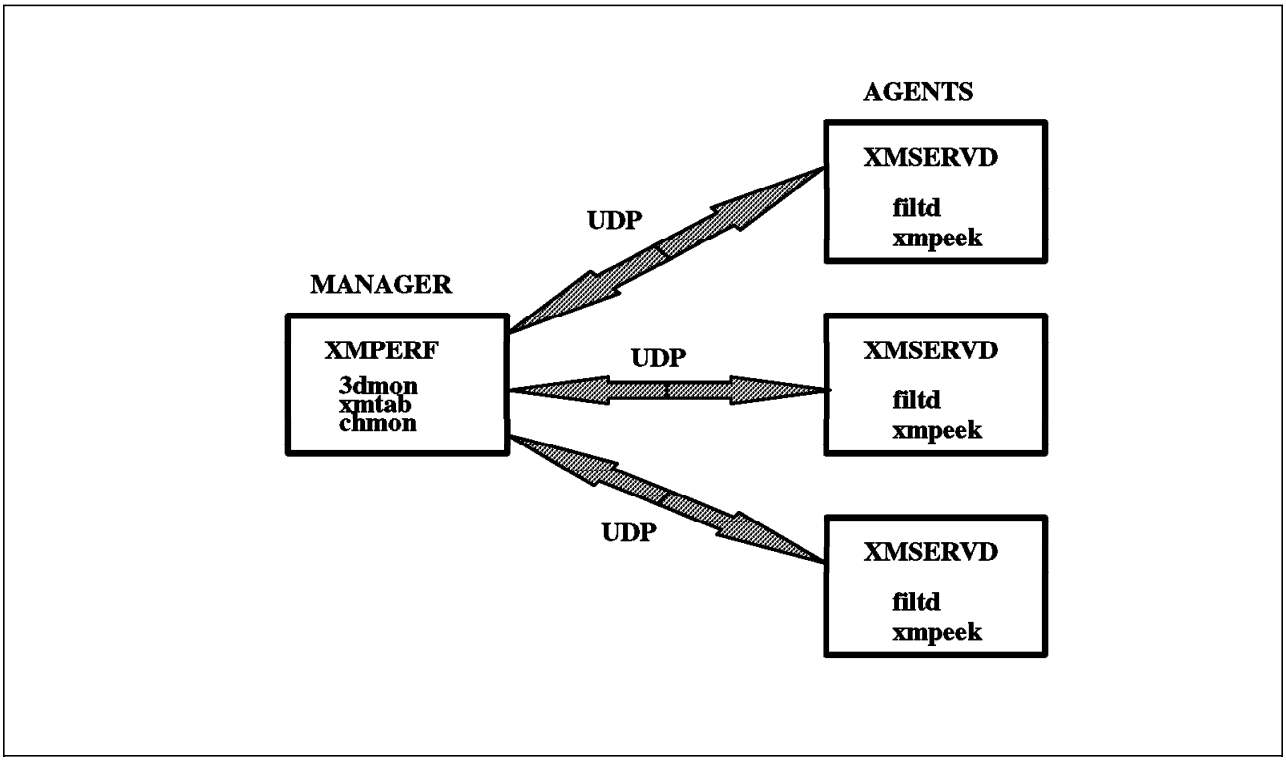

Figure 92. General Organization

PTX consists of four executables:

xmperf The main program of PTX, showing statistics

3dmon A program giving a 3-D view of performance statistics

xmtab Postprocesses recordings of monitoring sessions

chmon A program giving a simple ASCII-like activity monitor

It also provides an application programming interface (API), called Remote Statistics Interface (RSi).

PAIDE consists of three executables:

xmservd The main program of PAIDE, supplying statistics

filtd Data reduction and alarm daemon to create customized statistics

xmpeek A program that can query statistics and status of `xmservd`

It provides an application programming interface (API), called System Performance Measurement Interface (Spmi).

As with NetView for AIX, there is a client/server model with the manager being the client and the agent being the server. The communication protocol between the two components when the agent and the manager are not running on the same machine is User Datagram Protocol (UDP).

8.5 Performance Toolbox for AIX

After the installation, Performance Toolbox for AIX can be started on the command line by entering the `xmperf` command. There are a few options with this command, which can be ignored initially. Two windows, as shown in Figure 93 on page 111, appear on your screen.

One is the PTX main window, and the other is a console window started automatically because it was specified in the xmperf.cf configuration file of PTX. When starting out, you can work with the default configuration file.

In PTX the information window is called a console. In this console there are some subwindows called instruments. Each instrument has a graphical representation (such as pie, state bar, speedometer) for displaying multiple statistics concurrently. Each statistic is a variable which can have attributes such as a label, a color, or a threshold value.

For instance, in Figure 93 on page 111, you can see a console called `andrea: Mini Monitor`, which contains one instrument displaying sixteen variables as graphical bars. These variables are, by default, the CPU consumption (kernel, wait, user), the network view with the tokenring activity, TCP and UDP activity, the page space utilization, storage instruction called, the run and swap queues, and how busy the disks are. They are defined in the configuration file xmperf.cf. You can add or delete the variables easily with the `bar` option of your console window. This menu allows you to manipulate the console (copy, open, create and delete an instrument or the console) and modify the values (add, delete, change a value or their attributes). You can record the statistics of an instrument or a console with the recording option.

The PTX main window has a menu bar with the following menus:

File	For general action and exit
Monitor	To open, close or create consoles
Analysis	Tools for performance measurement
Controls	Tuning and control
Utilities	Additional useful tools
Help	Online Help

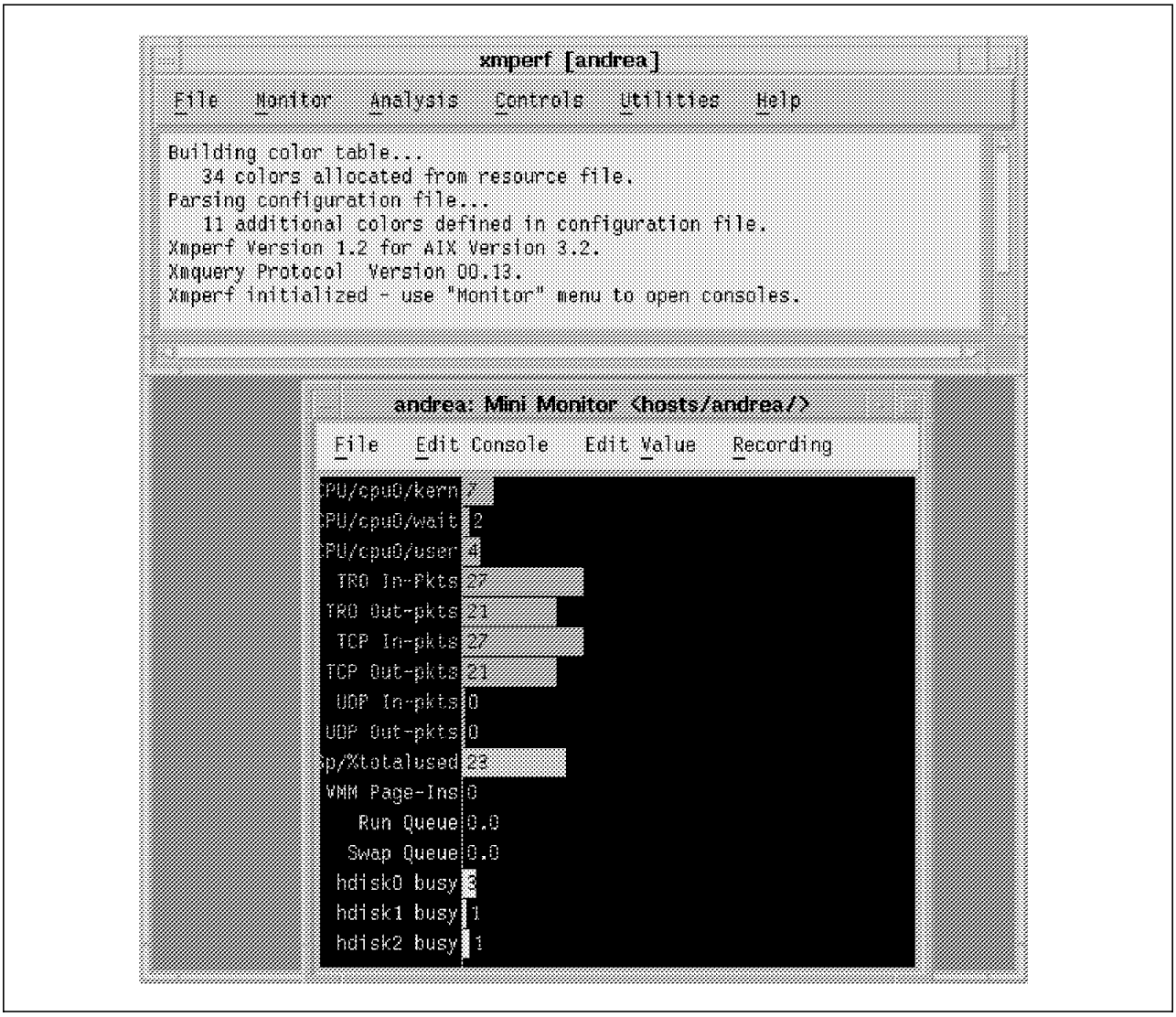

Figure 93. PTX Main Window

8.5.1 File Menu

Figure 94 shows the File menu.

Save All Changes When you use this option, all the changes to all consoles will be saved. These changes can affect a console, an instrument in a console or a variable in an instrument.

Playback This starts the playback feature, which shows you a session you have previously recorded from a Recording submenu console. You can choose which playback file you want to see and look at your recording at the speed you prefer (slower or faster).

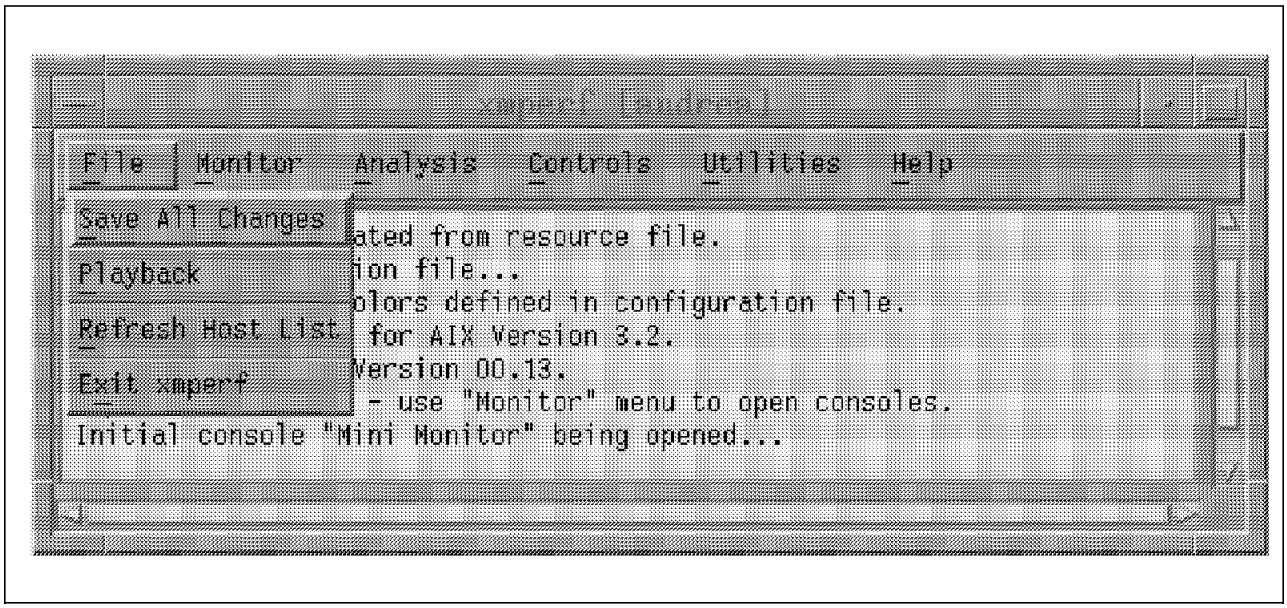

Figure 94. File Menu

Refresh Host List This refreshes the list of hosts currently available on the network. There is an automatic update, which is done not more than once every five minutes. This option allows you to refresh whenever you like.

Exit xmperf This terminates the entire application and warns you if there are unsaved changes.

8.5.2 Monitor Menu

With this menu we can see all the power and ease of use of PTX. As you can see in Figure 95, the `Mini Monitor` is marked with an asterisk(*). This means it is a running console. If you want to stop the console just click on it, the asterisk(*) disappears, followed by your console. This console was started at the beginning of `xmperf` but you can also select one of the five predefined consoles:

Local System Monitor To display all your major local values in eight different area instruments

Combo Style Sample Seven different instruments to show you the different styles available

Dashboard Style Sample It is just like an automobile dashboard

Cockpit Style Sample It is similar to an aircraft cockpit layout

Local CPU Sample Very light CPU view with some thresholds

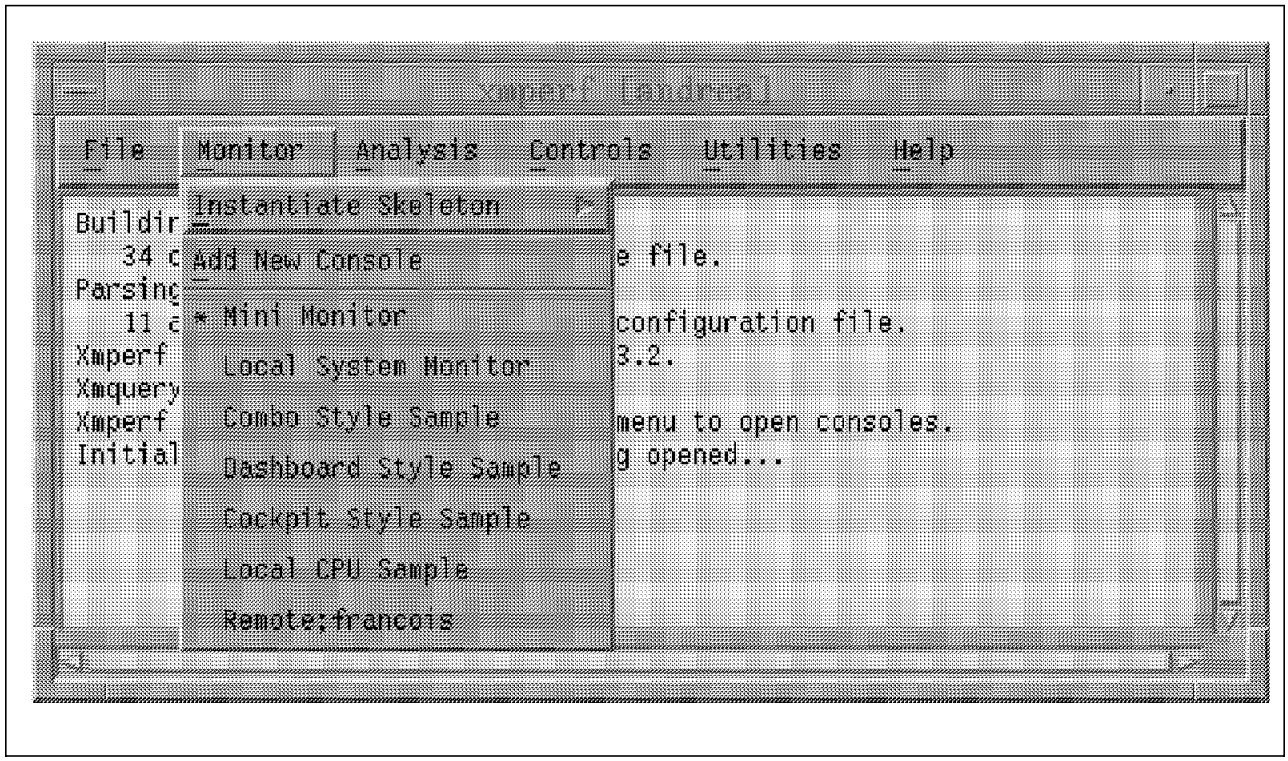

Figure 95. Your Performance Consoles

As shown in Figure 95, all user defined consoles are also available in this pull down menu, for instance `Remote;francois`.

8.5.2.1 Delete a Console

If you want to delete a console, you just call it with your push selection (clicking on the selected console). You display the console File menu and select the Erase Console option, as you can see in Figure 96, which shows the Local System Monitor console.

As soon as this change occurs, your console disappears from your screen and from your Monitor main menu. If you want to recover this deleted console simply exit xmperf without saving any changes.

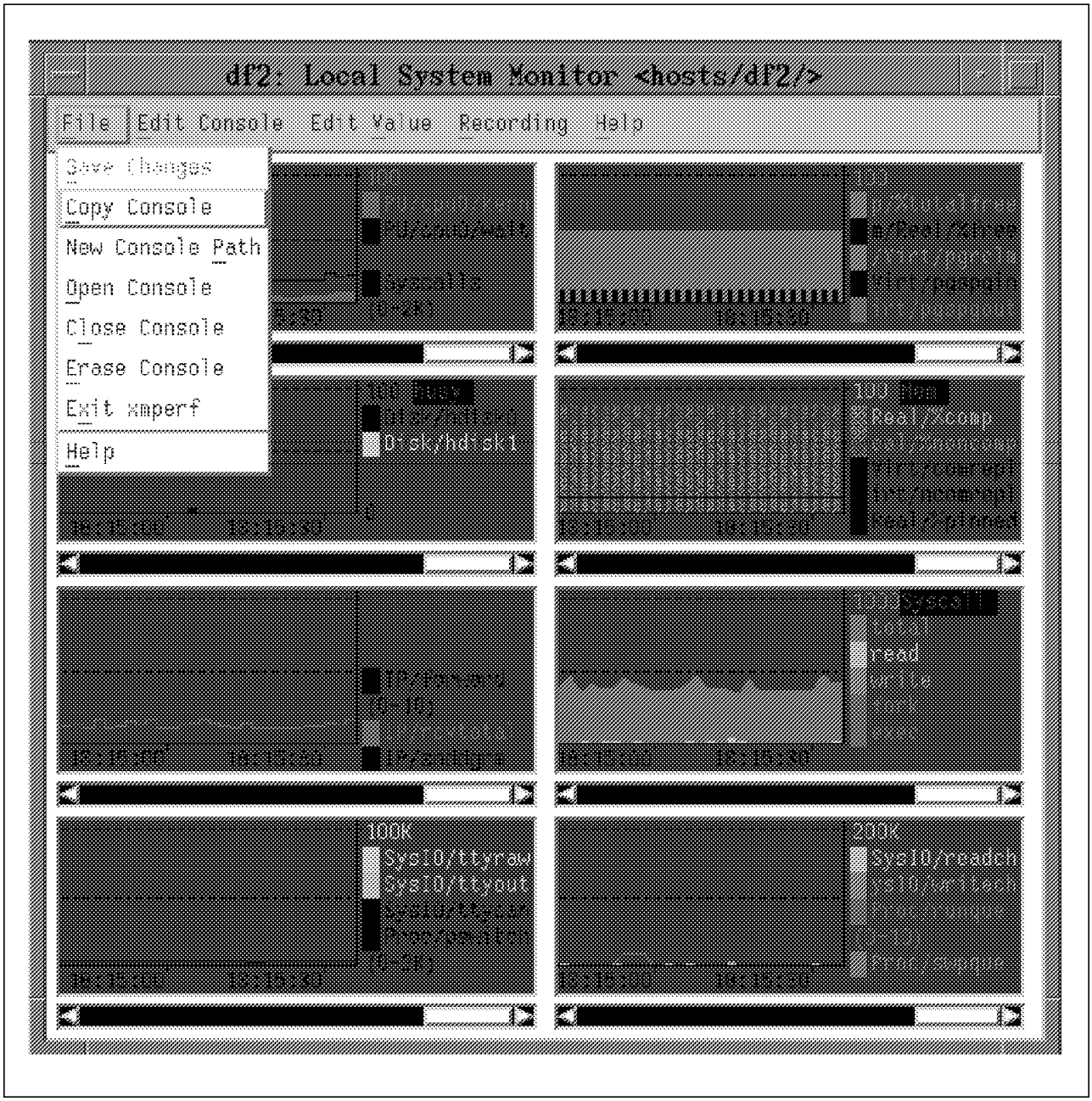

Figure 96. Console File SubMenu

This useful console displays eight different instruments about CPU consumption: Paging space usage, Disk I/O, Storage, Traffic Network, Syscall, Sysio/tty and Run queue.

From the displayed sub-menu, you can also copy and close the console, or open a new existing one.

8.5.2.2 Change a Console

If you want to customize a console, you have to display it and use the Edit Console or Edit Value choices which can be seen in the menu bar of Figure 97, which shows the Dashboard Style Sample console. This shows basically the CPU, SysIO and Storage utilization displayed in speedometer style.

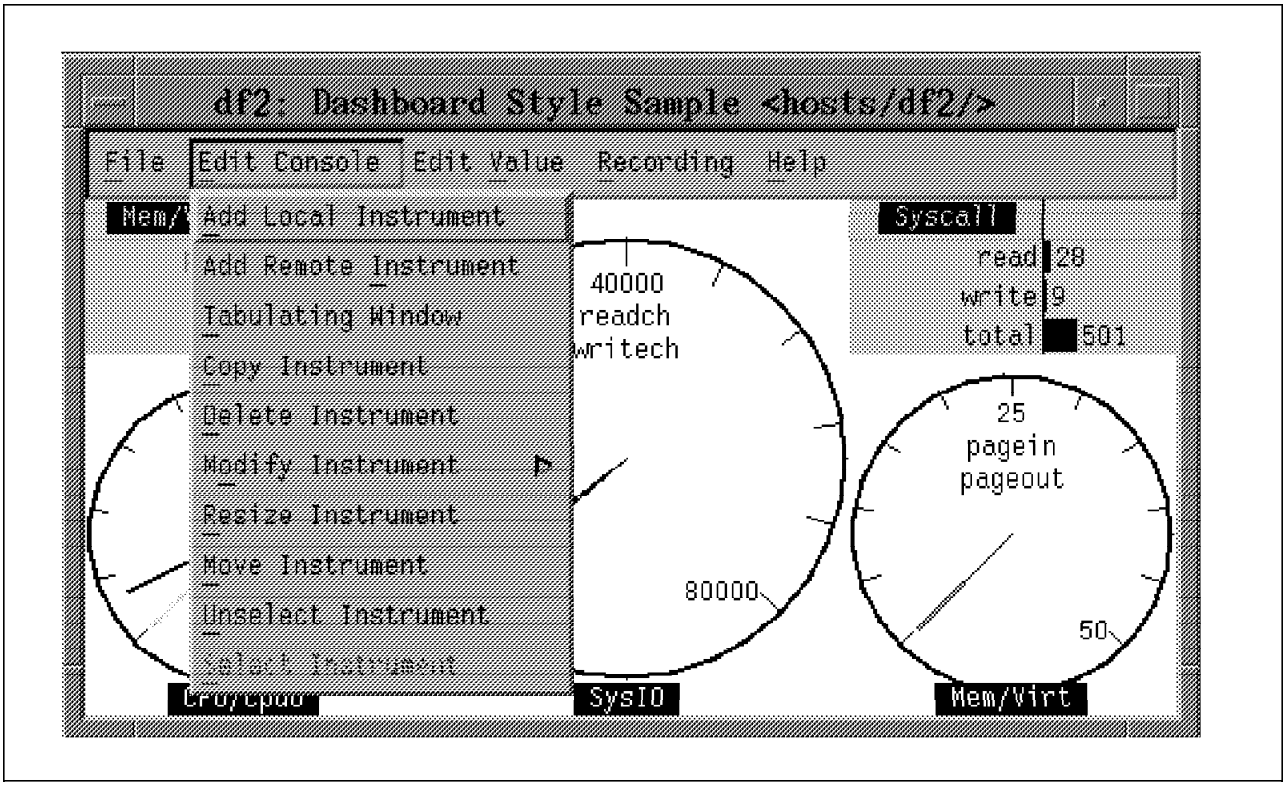

Figure 97. Edit Console Submenu

First you need to click on one instrument of your console. Now, you can activate the Edit Console menu to change your instrumentation. You can add, delete, or move any instrument. You can also use the Edit Value menu to add, delete or change the value or the attribute of any value of your instrument (a color for instance). All modifications are immediately executed, so you can see the change. If you are satisfied with your changes, you can save them with the Save option of the File submenu.

When you select a value to add, you get a set of default properties for that data value. These properties can be changed to reflect your specific needs. The properties are: style, color, scale, threshold and label. Each value is given a unique name, and this name is composed of single-level names separated by slashes, like a fully qualified UNIX file name. For instance, the statistics for the user component of CPU utilization on machine *andrea* would be called: /hosts/andrea/CPU/cpu0/user. This fully qualified name is called the path name of the value.

When you add a value you first see a dialog window called a Value Selection Window with a list of the top layer of values that you can select from. The value selection is done from a series of cascading lists, each level of which is

represented by a part of the value path name. In this way, you can move down to the ultimate level of statistics from which you wish to select.

8.5.2.3 Add a Console

There are several methods for creating a new console. The simplest way is to use one of the existing predefined fixed consoles shown in Figure 95 on page 113. As a quick start, you can modify those consoles with the method already explained in section 8.5.2.2, “Change a Console” on page 115. If none of the predefined consoles is satisfactory, you can use the Instantiate Skeleton menu shown in Figure 98, where you select one of twelve basic consoles to display your statistics. As its name indicates, the skeleton window allows the user to dynamically select specific elements from a class of objects such as disks, processes, LAN interfaces, and network nodes. You can then create dynamic views of statistics from the selected elements. Suppose you have ten machines running the xmservd program. When you select the Remote Mini Monitor skeleton option, you can select from a submenu list the machines you want to see. After selecting one or more machines, xmperf creates a fixed console with a set of instruments for each selected machine.

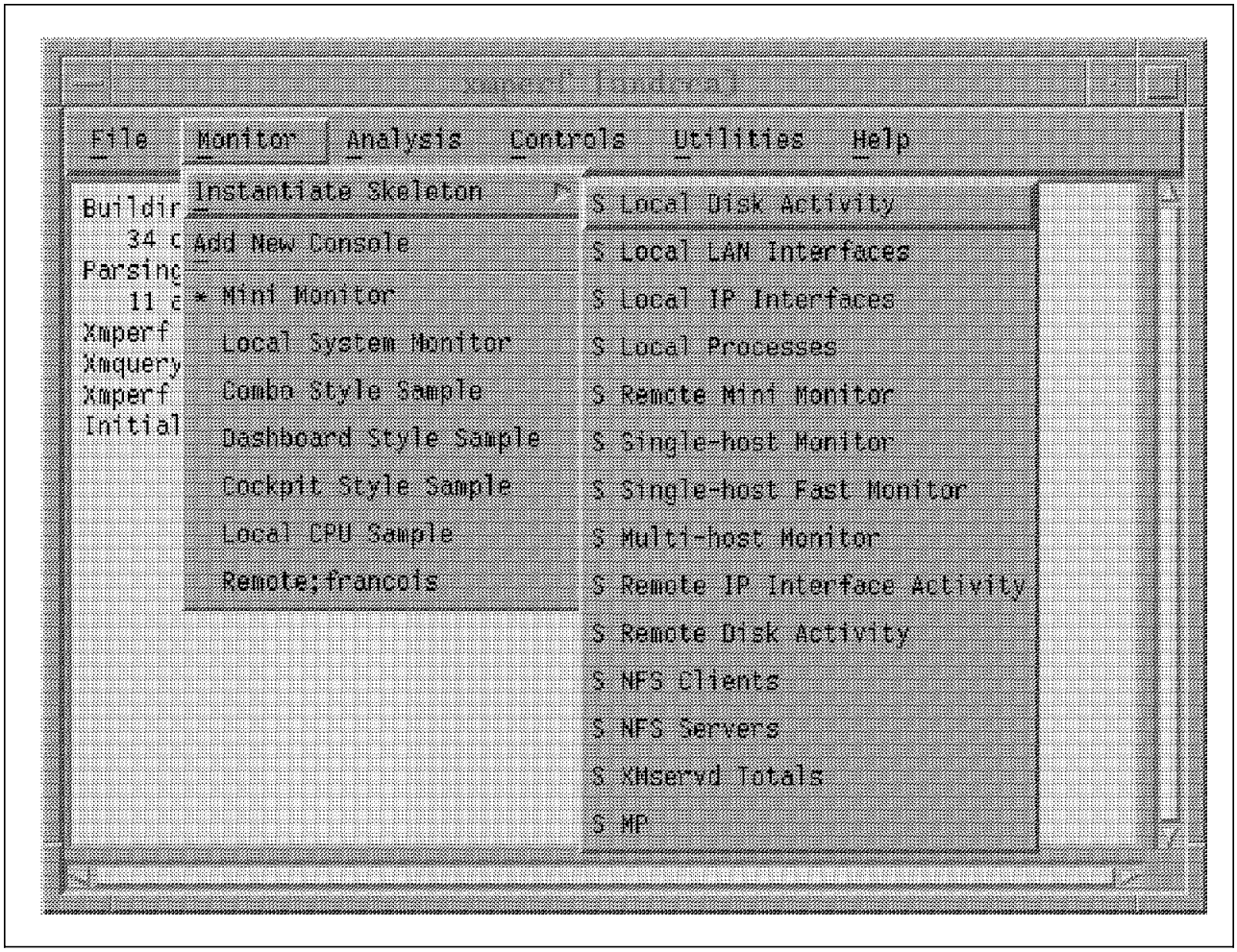

Figure 98. Skeleton Submenu

The strength of this skeleton method is that you choose the console which most closely fits your requirements. After defining any missing elements, you create your fixed console. You do not need to start from scratch, since you already have some instruments and variables which closely match your needs.

Another method, for the advanced user, involves starting from scratch with the Add New Console option, that you can see in Figure 98. You need to define every instrument and statistic value, but even with this option, xmperf helps you to create and define your elements.

8.5.3 Analysis Menu

In this menu, xmperf provides several performance tools, which are discussed in greater detail in the *AIX System Management Tips and Techniques* manual.

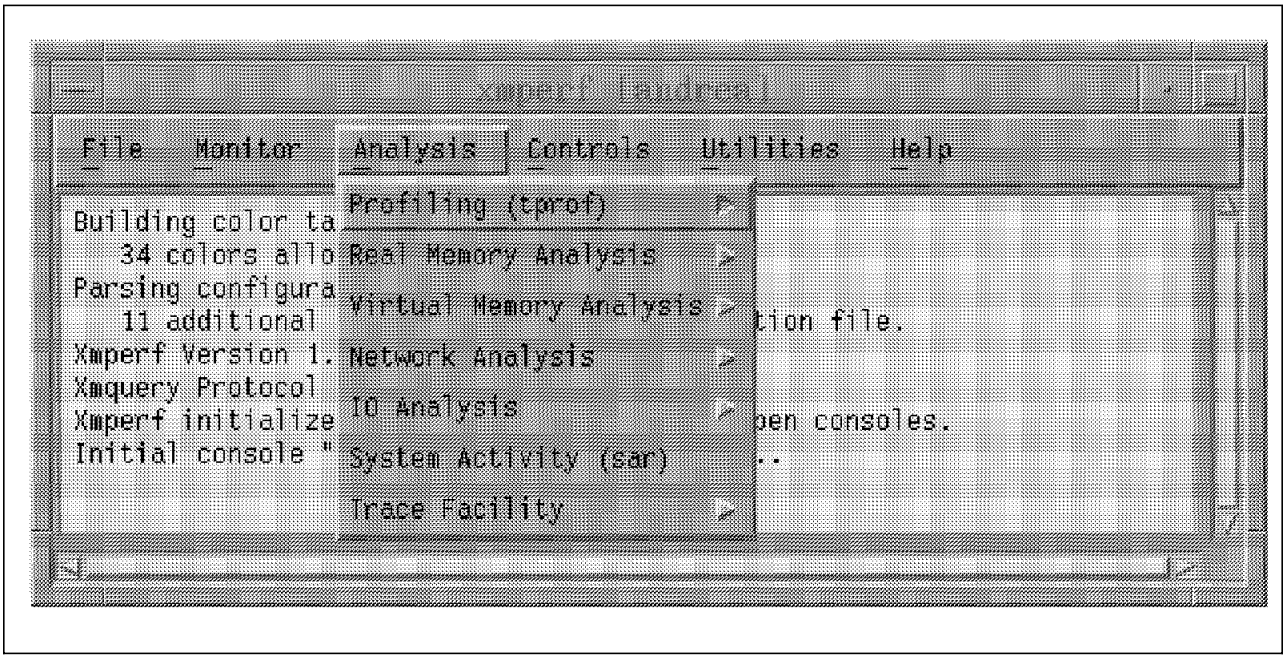

Figure 99. Analysis Menu

Reduced Memory Simulator
Calls the rmss command to simulate reduced real memory, prior to analyzing a particular workload on your machine.

Profiling
Starts the tprof command, which helps you to evaluate programming performance.

Virtual Memory Analysis
You can start the svmon or vmstat commands to focus on storage analysis.

Network Analysis
Helps you to understand your network load by using the nfsstat, netstat and netpmon commands.

IO Analysis
This option calls the I/O related commands and allows you to study the I/O usage of your machine. The commands are filemon,fileplace and iostat.

System Activity (sar)
You can start the well known sar analysis.

Trace Facility
Allows you to interact with the standard trace facility.

For performance analysis purposes, this menu provides a convenient way to access all the main tools from a single location.

8.5.4 Controls Menu

This pulldown menu provides three options for performance tuning, as shown in Figure 100.

Process Controls Selecting this option opens a window which permits you to view and sort local processes. You can select one or more processes, to which various commands may be applied. These commands include: `kill`, `renice` or `svmon`. This window is an example of a console which could have been defined using the skeleton menu `S Local Processes`, shown in Figure 98 on page 116.

Network Tuning You get specific information about the network by using the `no` command.

Disk Tuning This starts the the LVM editor, called with the `lvedit` command, to modify the specifications of your LVM definitions.

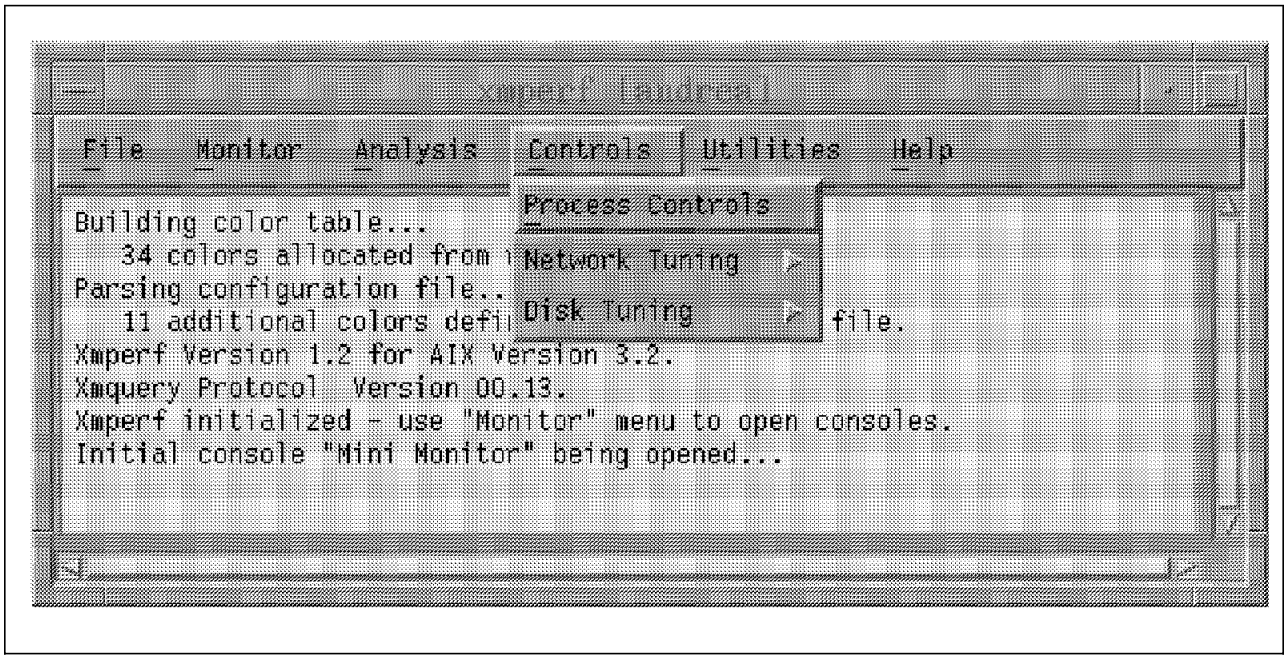

Figure 100. Controls Menu

If desired, other tuning programs can be added to this menu.

8.5.5 Utilities Menu

This menu contains several useful tools.

Remote Processes

Permits the display of the processes of the selected remote machines (sorted by highest CPU utilization). This window is not as functionally rich as the Local Processes window at this time, but it is very useful to know which is the most CPU intensive process on a remote node.

System Monitor This option gives you a Monitor ASCII screen, with the most important parameters and the *x* largest processes (where you specify *x*) of your machine or any remote machine. Because of the ASCII interface, this tool does not consume much resource and can be run on several nodes for an extended period of time.

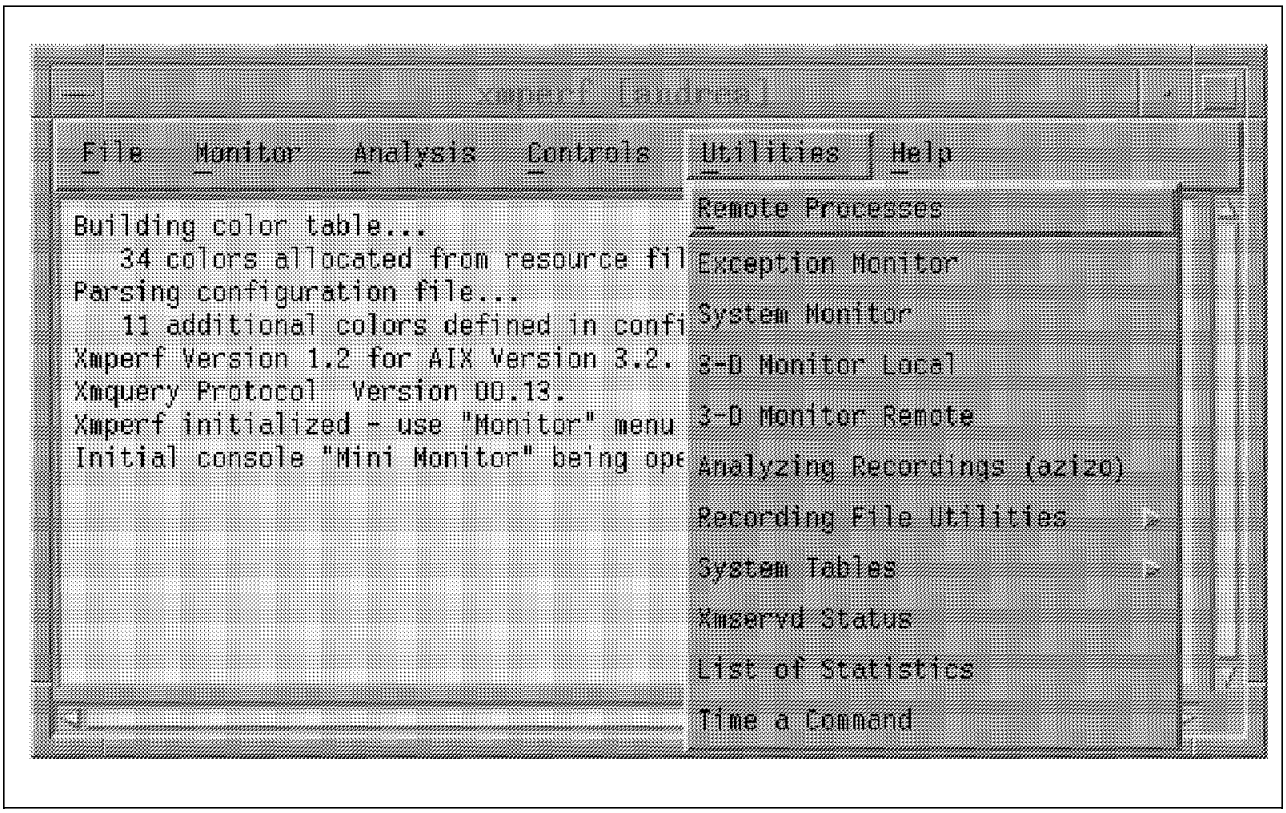

Figure 101. Utilities Menu

3-D Monitor (Local and Remote)

Provides an interesting presentation of your performance statistics in a 3-dimensional view. There are five preconfigured samples for processes, LAN interfaces, IP interfaces, disks and file systems. This graphical view can be very useful, for instance, when you are comparing the differences between several servers.

System Tables Shows the three internal tables (Process, File, Inode) generated by the `pstat` command.

Xmservd Status Allows you to understand the xmservd activity. This could prove to be very important if your queries result in too much information from different remote nodes. To avoid overloading your agent, you can use a limitation file to customize your request.

List of Statistics Gives you all the available statistics of the remote or local node.

Time a command Gives you the results of a timex -s command.

8.5.6 Help Menu

The Help menu, shown in Figure 102, is rarely used because PTX is so easy to use and self-explanatory.

However if you need some information about a specific subject, the Help Index option lets you quickly find your information through a series of cascading pull-down menus.

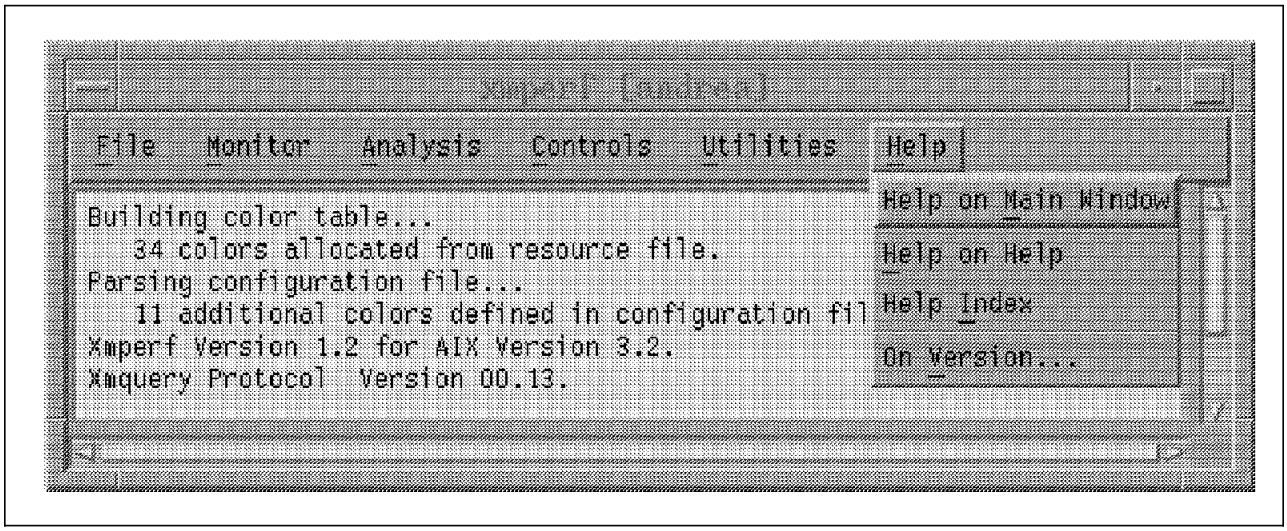

Figure 102. Help Menu

If you want to have more specific information about the interfaces or other details of this product, review *Performance Toolbox 1.2 and 2.1 for AIX Guide and Reference*.

8.6 Performance Aide for AIX

The Performance Toolbox for AIX works using a client/server model. The server side of that relationship is constituted by the xmservd program, which is provided by the PAIDE product. It is the data supplier giving the statistics to the data consumer manager. You need to install xmservd on each monitored node.

You can reduce the statistics and create alarms with the filtd program, which is another component of PAIDE. The primary purpose of this program is to reduce large amounts of collected data into fewer and more meaningful statistics. In addition, filtd allows you to define *alarms* by setting thresholds and specifying the actions to initiate when you reach them. There are different kinds of actions that can be specified:

- Execution of any command on the node where filtd is running
- Exception to forward a special message to a data consumer like xmperf
- Emission of an SNMP trap to an SNMP manager like NetView for AIX

You can easily customize the filtd program through a configuration file called filtd.cf.

The xmservd program can also be configured as a subagent for Simple Network Management Protocol (SNMP) agents. That means the statistic variables of xmservd can be supplied through SNMP to NetView for AIX, which is an SNMP manager. Figure 103 shows you the relationship between PTX and NetView.

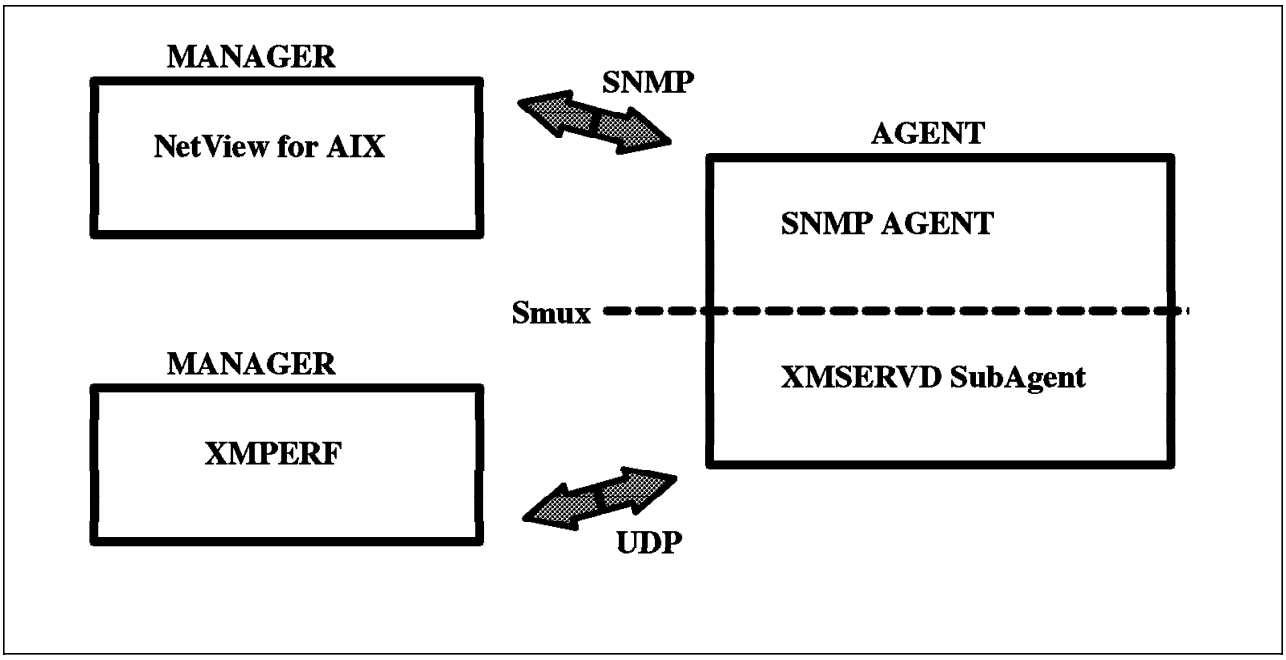

Figure 103. Xmservd as an SNMP SubAgent

In the real world, the two managers will run on the same machine and the connection between the NetView-Xmperf manager and SNMP-Agent/Xmservd (subagent) will be through SNMP and UDP.

To activate that relationship, you need to set up your SNMP configuration by adding (or verifying) the smux definition statements in the two definitions files:

```
In /etc/snmpd.conf, add the following line:
smux      1.3.6.1.4.1.2.3.1.2.1.3        xmservd_pw  # xmservd

In /etc/snmpd.peers, add the following three lines:
xmservd   1.3.6.1.4.1.2.3.1.2.1.3        xmservd_pw

  do a refresh of snmpd with:
         refresh -s snmpd
```

You need to activate the xmservd program as an SNMP subagent. Add the following statement to the xmservd configuration file:

```
In /usr/lpp/perfagent/xmservd.res, add the following line:
       dosmux
Do a refresh of xmservd with:
       kill -1 PID_of_xmservd
```

You must create your xmservd values as SNMP variables with the creation of source MIB in ASN.1 notation with just a SIGINT (kill -2) on xmservd:

```
       kill -2 PID_of_xmservd
  to create a /etc/perf/xmservd.mib
```

You load this file on your SNMP manager. NetView for AIX has an option to do this. Now, from NetView, you can collect the SNMP variables concerning xmservd.

You can set a NetView menu in /usr/OV/registration/C directory to start xmperf from your Network manager:

```
/*
        Registration for NetView for AIX API XMPERF application
        @(#)$Revision: 1.0 $ $Date: 94/11/15 15:00:00  $
*/
Application    "Monitor_XMPERF"      {

        MenuBar    " Monitor "        _M {
                   "Xmperf"    _X      f.action     "xmperf";
                                }

        Action   "xmperf"     { Command '/usr/bin/xmperf' ; }
      }
```

8.7 Conclusion to PTX

Performance Toolbox and Performance Aide for AIX allow you to monitor in realtime your system resource consumption. You can monitor a large number of variables on one or more servers (using, as an example, the Mini Monitor console from Figure 93 on page 111), at a fairly small cost in system overhead (around 1 to 2 percent of CPU for the xmservd daemon). Of course, the more information you try to gather, the more CPU resource will be required.

For a critical server machine you can set the filtd program to be alerted at a threshold of, say, 80 percent CPU busy. This is a new variable defined in the filtd configuration file (filtd.cf). With this simple capability, you can monitor an important server without spending a lot of time and resource to setup a special environment.

On the other hand, if you want to merge xmservd with your network manager NetView for AIX, you can easily transform your xmservd to a *proxy-agent*, which provides the values of xmservd variables in SNMP variables to your manager.

In addition, if you want to use your own routines to analyze the results on the manager side, or add your own routines to filter the values on the agent side, you can do this with the help of the two APIs, one on the manager (RSi - Remote Statistics Interface) the other on the agent (Spmi - System Performance Measurement Interface).

Performance Toolbox and Performance Aide for AIX are powerful applications to assist you in your daily performance management tasks.

Chapter 9. Distributed Security

You could find many meaningfull definitions about security in books or AIX documentation. Herein, the purpose is to give you the elements for choosing the best tools for your security network.

9.1 Prerequisites

If you want to implement security, you must first check that certain prerequisites have been met.

9.1.1 Information System Safety

Your machines and your network must be safe before they can be secured.

Safety consists of:

- Correctness of your installation
- Stability of the software
- Quality of the administration
- Control of time synchronization
- Administrators′ ethic

9.1.1.1 Safety Prerequisites

A correct installation implies that your machine is in a safe site. A safe site means, for instance, that the machines are protected against power supply outages. It is pointless to worry about security on a machine that is constantly being rebooted.

The stability of the software means your software has few bugs. To ensure that this is the case, you must avoid using new and untested versions of software. You should also be sure that any bugs that threaten security are quickly fixed. For more information, see 9.1.1.2, “ How to Fix Well-Known Security Holes” on page 126). Stability also involves avoiding the use of software obtained from questionable sources. There are no viruses on UNIX machines, but some undesireable results may occur if you execute new software with *superuser* authority. AIX has tools to analyze new software (see 9.4.6, “Software Audit” on page 135).

The quality of the administration means you have a good way to quickly restart from a safe and tested system. If an intrusion is detected, you must have a safe backup preceding the intrusion date. Also, a safe backup must not be too old. A regular backup strategy is a key to quick restarts with minimal losses (see Chapter 5, “Application Data Backup Strategies” on page 51).

Time synchronization across machines is essential for all trace and audit functions on the system. It becomes indispensable for distributed software tools based on Kerberos** because authentication uses a timestamp to avoid replay.

Of course, none of the preceding elements can help without a good administrator′s ethic. Once the wolf gets among the sheep, it is too late to build a security strategy. In this situation, there is no solution.

© Copyright IBM Corp. 1993,1994

9.1.1.2 How to Fix Well-Known Security Holes

There is no way to have security if your software has well known bugs and if you do nothing to correct them. To avoid the well known bugs, you can follow the advisories published by the Computer Emergency Response Team. CERT is a USA organization which publishes through email the most recently discovered security holes. A CERT advisory explains:

- What is the danger posed by this security hole.
- How to quickly close this hole.
- Identify the name of the fix to request from the vendor in order to repair the security hole.

Everybody can request a subscription to the CERT advisories by sending an email to *cert-advisory@cert.org*. Also, through anonymous `ftp` at cert.org (192.88.209.5), you can get many software tools to improve security, such as *TCP Wrappers*.

9.1.2 Physical Security Prerequisites

Physical security consists of items such as:

- Ethernet wires would not be accessible by someone wishing to plug an unauthorized machine for unapproved purposes. In other words, you should be very conscious of how LAN or communications cabling is laid out.
- Machine key must not be left in the keylock.
- User must shield his keyboard when typing his password.
- User must not write down his password (see 9.4.2, “ Password Configuration and Checking” on page 133).

This is obviously not an exhaustive list.

9.1.3 Security Strategy

Security is not just the use of some available tools on your machine. Security involves a compromise between the cost in time and resources to implement security and the risk of losing critical information. When you purchase insurance, you weigh the cost of insurance against the risks you cover with this insurance. The use of the security tools and how you use them is defined by your strategy. Your security strategy is described by its properties and its policies.

9.1.3.1 Do You Need Security?

Ask yourself this question, or better yet, these:

- Do you have some critical information for enterprise survival?
- Do you have the prerequisites (see 9.1, “ Prerequisites” on page 125)?
- Do you have the money and people needed to apply security?

9.1.3.2 If Yes, Scenario Proposal

The following is a proposed scenario, keep in mind that security is not the use of advantageous tools, but the business of assuring the survival of the enterprise.

1. Bring together all the persons concerned.
2. Brain-storm on the need for security.
3. Present all available tools and their costs: price (if not included in your base system) plus customization.
4. Weigh the costs against the risks.
5. Choose the best tools for your situation.
6. Define the security properties and policies.
7. Define how to use the tools needed to apply each property and each policy.
8. Explain these to your technical people, especially the administrators and operators.
9. Advertise these choices to all your users.

Security has to be understood by all the people to be as basic as remembering to lock the front door at home.

9.2 Different Environments

A set of tools in UNIX and additional ones from AIX permit administrators to configure very different types of environment. The security tools listed later in this document reference these types of environment. Therefore, building your security strategy will be facilitated by this approach.

9.2.1 Locked Environment

A locked environment is a local network of machines without any external connection, and where only administrators have *superuser* access on all machines. This is typical of a commercial enterprise, where the users access the machines only through the applications. The simplest case is the environment where users have no shell access and no compiler access.

9.2.2 Closed Environment

A closed environment is a local network of machines without external connection, but some users have *superuser* privileges on their machines. An example of this environment would be an enterprise where engineers work on their own workstation, but where the information is confidential, such as a military site or a manufacturer of military equipment.

9.2.3 Open Environment

An open environment is a local network which also requires access to external services as `mail`, `news`, remote file transfer (`ftp`), remote connection (`telnet`), remote archival (`archie`, `gopher`, `wais`). This is typical of universities or research centers, but many commercial enterprises have begun to implement this type of environment. The most common connection is the Internet.

For this type of IP connection, you have to use a firewall. The idea is to establish a perimeter defense. After performing a risk analysis, you must be sure that all the entry points into the network are protected equally well. You

also have to keep in mind that the firewall will need to protect the network from attacks (intentional or accidental) from the inside as well as the outside. Using routers with service/address couple filters facilitates the security building, but several issues may arise as a result (see Firewall issues, Chapter 10, “Wide Area Network Security” on page 137).

9.3 Many Distributed Security Tools

This section discusses many of the products that you can find on AIX machines. Some are part of the base operating system, some are LPPs, some are freeware, and some are new to the market.

This part advises you on elaborating your strategy. The constantly evolving UNIX world doesn’t allow you to pinpoint any single solution as being the best. Your strategy also depends on your type of environment.

For each tool, we will:

- Describe the tool.
- Explain the nature of any security threats that may arise from the use of the tools (but not too explicitly, since this is not meant to be a guide for hackers).
- Present some solutions to address these threats.

9.3.1 Address Resolution Protocol (ARP)

On a local network, the ARP protocol allows a workstation to spray the question, ″who has this internet address?″ Based on the first answer on the network, the asking workstation builds its ARP table, which is used for address resolution. This can be a serious threat because the first answer may be not necessarily have come from the right machine.

A solution is to pin the corresponding Ethernet/Internet addresses in a file, and after TCP/IP is started to pin the ARP table (see `arp` command, `-s` and `-f` arguments).

You can also disable ARP for a device with the `ifconfig` command.

This mechanism is also available for token ring and FDDI.

9.3.2 Network Information Service (Previously Called Yellow Pages)

With this product, a set of security servers distributes the identification and the authentication. The strategy is to show a single image for user administration. In this context, every user has the same uid across all the machines. This allows NFS to work properly, as we will discuss later (see 9.3.3, “ Network File System” on page 129).

With NIS use, you have to deny *superuser* access for all users except the administrator because the set of NIS servers is not well identified. NIS should only be used in the locked environment. If a hacker can access a *superuser* account on any one machine, he is able to become *superuser* on all machines. The only solution is to use a better distributed identification/authentication service such as NIS+ or DCE:

- NIS+ is a product of SUNSoft not yet marketed by IBM, NIS+ is a part of ONC chosen for COSE′s administration environment.

- DCE will be discussed later (see 9.3.5, “DCE” on page 129). DCE has also been chosen for COSE′s distributed environment.

9.3.3 Network File System

With this product, a file server distributes file systems on a local network. For NFS to work properly, all users must have a consistent uid accross all machines. To accomplish this, you must implement tools such as NIS, NIS+, DCE or others that will be proposed later (see 9.3.7, “ Remote Distribution” on page 130). The NFS server permits remote mounts through the `export` function, and authorizes remote machines to mount its exported directories.

If the export mechanism allows any remote machines to access a filesystem, a user with *superuser* privileges on a client machine may become, at any time, any user with all the accompanying privileges for that user′s files.

In an unlocked environment, you can use NFS securely if you follow some rules:

- Avoid superuser equivalence between machines.
- Export a user′s home directory only to the client machines where the owner user is likely to have a login session.
- Export binaries only to the machines under your control.
- Use the nosuid and nodev features.
- Use Secure NFS (machine identification/authentication). This LPP is not in the Base Operating System and perhaps is not available in all countries.

9.3.4 Kerberos

Kerberos is a very good solution for a closed environment and can be used for an open environment. Kerberos uses DES cypher and decypher functions for all identification/authentication transactions. Kerberos has been developed at MIT for the Athena project. It has been in use for a long time at MIT, and many other places as well. All the vendors have decided to use Kerberos, but this is done inside the DCE product from OSF.

While Kerberos is a good solution in the USA, the exports of DES are very limited outside the USA, so few non-US organizations use Kerberos. Furthermore, the DES encryption has national security implications in certain countries and usage needs special authorization. So far, only a few products have been kerberized, such as Secure NFS (see 9.3.3, “ Network File System”) and perhaps some kerberized `telnet` and `ftp`.

9.3.5 DCE

DCE is a complete single system image. Users no longer need to know on which machine they are located, or where the files they wish to use are stored. The cell represents a large machine composed of a set of heterogeneous machines. Security is an integrated DCE feature. Security is implemented with Kerberos identification/authentication and with some new features for access control. These latter features affect DCE′s weak degree of interoperability with Kerberos Version 5 from MIT. The OSF Single Interest Group for DCE security has decided recently (October 93) to add the audit function in DCE, a Toronto IBM labs product.

As DCE is a vendor product, the export of DES and usage problems are resolved. However today (November 93), DCE has to wait for enough important software applications to adopt it as a security network standard.

DCE has a very good future because it has been chosen by PowerOpen*, COSE and Unified UNIX.

9.3.6 Remote Commands

Remote commands are:

- `rlogin` for remote login
- `rsh` for remote execution
- `rcp` for remote file transfer

These commands are duplicated with the classic ARPA commands:

- `telnet` for remote login
- `rsh` for remote execution
- `ftp` for remote file transfer

There are also some more `R` commands: `rdump`/`rrestore` for remote backup and `rdist` (see 9.3.7, " Remote Distribution").

The `R` commands advantage is to suppress the passing of clear passwords on networks where some PC machine has installed a sniffer software which scans for passwords.

The `R` commands are generally more friendly commands, but introduce the use of the private .rhost file and general /etc/hosts.equiv file. These files allow equivalence between accounts without authentication once a user is first identified by a machine. This mechanism allows a user a one time authentication for a whole local network.

This means that, once a hacker gains access to an ordinary user account, he can access the same account all over the network. This may happen if some machines are not under control (as in the case of 9.2.2, " Closed Environment" on page 127).

The more practical `rcp` command loses its advantage over the `ftp` command when you properly implement a distribution system such as NFS. AIX adds new features to suppress the automatic login allowed by the too dangerous private .netrc file used by the ARPA commands, with the possibility of a clear password inside (see 9.3.9, " TCP/IP Features" on page 131).

9.3.7 Remote Distribution

In a closed environment, a way to have a single users administration system, is to use remote distribution (see `rdist` command). In this case, you must suppress the use of other `R` commands and the rlogind daemon, because a /.rhosts file is required. You build a rdist file for distributing your administration files from the security server, which is a very secure machine. Periodically or every time a modification is done, you update the administration files on the client machines.

This mechanism gives you a single image for user administration. The advantage is that a broken server can no longer disable local

identification/authentication for client machines. The /.rhost file is only placed on client machines and contains only one entry, the security server. Evidently, this security server must be well-identified by a static ethernet/internet address. Therefore, you control the distribution. In the case of a client machine is down for a while, at the next distribution, it will update the same data base than the other client.

The change of the password must be done by a special login on the security server. Besides, the Department of Defense has created a divergence between the main standard UNIX systems with respect to their implementations of C2 classification security guidelines. As a result, in a heterogeneous network, you have to customize or to rename some security files.

9.3.8 Centralized or Distributed Logging

In classical UNIX systems, the *syslog* daemon allows you to log some actions, such as:

- Instances of the switching user command, `su`.
- Envelopes of all the emails transferred by `sendmail`.
- Headers of all news transferred by the nntpd daemon.
- Instances of the `ftp` command and its sub-commands.
- Connections to the ftpd daemon.

You have to keep in mind, that syslogd daemons may converse among themselves. Therefore, you can centralize the log on one machine, or better yet, you can dispatch the logs to different machines.

Besides, syslog is an open product and it allows you to use eight channels for getting and distributing any other ASCII format logs. With a little C program, reading line by line from standard input and writing to syslog (see `openlog` and `syslog` system calls), you can for example, distribute the output of any local audit across the network.

Caution should be exercised, since the syslogd daemon uses the Datagram protocol (UDP/IP) which is not very reliable. If there is too much traffic on the network, you risk losing some log data.

9.3.9 TCP/IP Features

TCP/IP is a well tested set of tools for networking, which has led to many improvements in its security. The TCP/IP tools and a number of other AIX features allow you to manage your open environment in proper conditions.

Following is a list of security features. You can find more detailed explanations by a simple keyword search in InfoExplorer. You can also find some samples in the /usr/lpp/tcpip/samples directory.

9.3.9.1 Classic TCP/IP Features

Classic TCP/IP features are:

- The /etc/ftpusers file allows you to permit or deny `ftp` user access.
- The `ftpd` daemon changes the root of the file system's tree structure for the purpose of anonymous file transfer. This functionality is vital in the Internet

world, in order to be able to contain anonymous ftp users within a specific subset of your filesystem.

- If you use NIS, the /etc/netgroup file allows you to precisely define accesses to machines on the local network. You can use *netgroup* inside /etc/passwd, /etc/hosts.equiv and private .rhosts files.
- You can suppress the automatic routing feature of TCP/IP with the network option command (`no -a ipforwarding=0`).

9.3.9.2 AIX TCP/IP Features

AIX TCP/IP features are:

- User configuration allows you to permit or deny `telnet` and `rlogin` accesses. You can also limit these with the port list (see 9.4.1, " User Configuration").
- `securetcpip` modifies your system as described in 9.3.6, " Remote Commands" on page 130. This command also suppresses R commands and their associated daemons.
- Prevent use of private .netrc file by configuration of /etc/security/config file.
- Prevent non-authorized directories by tftpd daemon by /etc/tftpaccess.ctl file configuration. The `tftpd` daemon is required for booting X Window stations.
- Trusted path usage with `telnet` command (see 9.4.3, "Trusted Path" on page 133).

9.3.10 uucp Features

The *uucp* product is a nice way to distribute the mail or news facilities to an isolated machine. For instance, if a company has itinerant workers with portable computers or has employees working from home. *Serial Link Internet Protocol* has no possibility to authenticate the remote machine on a dial-up line. *uucp* needs the use of a login with password and allows to operate a call-back. For more information, see the /usr/lib/uucp/Permissions file. In addition, *uucp* logs all the transactions. The *uucp* product has been proven through many years of use, and is very reliable with respect to security.

9.4 Many Local Security Tools

This section discusses the integrated features of AIX that you have to use to improve your distributed security.

9.4.1 User Configuration

The user configuration function is extremely rich with AIX. NIS, however, truncates it, which is why we proposed a solution using `rdist`. This command permits you to distribute all the security files you want, the /etc/security/* files in particular.

The SMIT approach makes user administration much easier, and doesn't create any security threats. The new DSMIT product provides a very convenient way to administrate your network. However, DSMIT uses /.rhosts and /etc/hosts.equiv files with the rshd daemon. This solution is therefore only usable in a locked environmnent (see 9.2.1, " Locked Environment" on page 127).

The AIX security features you should use:

- No direct login for superuser, administrator must use the `su` command.

- Deny rlogin and telnet access if necessary.
- Define the list of accessible ports by user, indicating negative authorization with the ! character.
- Define the system group as the only group with the permission to use su on every account.
- Do not permit users to use cron at and batch daemons by *.deny or *.allow configurations of files in /var/adm/cron directory.
- Configuring the umask value to 027 is a good default. This denies all permissions for others and write permission for the group.
- Set the core variable to zero except if the user is a developer. This avoids the possibility of core dumps with confidential data inside.
- Define a default user configuration with very restrictive conditions and open facilities only on demand.
- Use the trusted path, especially where users have to change passwords in a secure way(see 9.4.3, “Trusted Path”).
- Do not permit users to use daemons (src, cron).

9.4.2 Password Configuration and Checking

A set of password restriction rules are available to enforce valid passwords. Some information on this subject is provided in the *Security Introduction* of InfoExplorer. In addition, the CERT anonymous ftp library contains the *crack* software program, which checks for trivial passwords. You should also find some good dictionaries about the most common trivial passwords. For other national languages, you should find localized dictionaries on national Internet backbones, such as inria.fr for France. You could also use some dictionaries for spell checking.

Since the number of possibilities for trivial passwords is enormous, checking for them involves a large amount of processing. As a result, such password checking can only be done on a local machine, and not by a usual network login. This information is provided for administrators wishing to anticipate some intrusions.

The password restrictions should apply only on local security. These restrictions should be set so that passwords are hard to guess, not hard to remember. Passwords that are hard to remember are often written down somewhere, which compromises system security.

9.4.3 Trusted Path

The trusted path is a solution to avoid a trojan horse. A trojan horse is a decoy that simulates a login banner or replaces a usual command. This is done to capture a user account′s name and password, or to use the user′s rights to perform an unauthorized action. If some trusted path feature is not implemented, there is no safe way to protect against this technique.

AIX implements a trusted path which is started with an invariable key sequence called Secure Attention Key (SAK). If you use SAK, you are assured of having either a safe login banner, or of executing (trusted shell) a very limited set of trusted commands included in the Trusted Computing Base (TCB). For example, inside TCB, you find the passwd command for changing passwords.

Before using the trusted path, you have to configure a user (tpath attribute in the /etc/security/user file) and configure a port (sak attribute in the /etc/security/login.cfg file). You have different options and for example, you can force the use of the trusted shell. This is used just for doing some critical operation such as password change (related to 9.3.7, " Remote Distribution" on page 130).

9.4.4 System Check

When you receive your AIX system, IBM gives assurance that this is a correct system and that no third parties have made changes to it. In theory, you receive your software from IBM in a sealed package.

9.4.4.1 Trusted Computing Base Check

After a certain amount of time has passed following the initial installation of your system, it is possible that some modifications may have been made that compromise your security. AIX provides you with a way to check the most secure elements of your system, which constitute the Trusted Computing Base. The /etc/security/sysck.cfg file describes all the characteristics of the all the files that make up the TCB. The `tcbck` command verifies the TCB through this file. It is very important to periodically verify your TCB (using, for example, a cron job).

This command can also be used to check the whole file system, by searching for:

- Files with *suid* or *sgid* bits
- Illegal links to TCB files
- Illegal TCB attribute set
- Special devices

9.4.4.2 Software Check

When you receive new software, it is important to verify its integrity as soon as possible. You can use the `sysck` command to check it (see also 9.4.6, "Software Audit" on page 135). This command uses the same */objrepos/inventory file as the `installp` command.

9.4.5 Audit

Often it is not possible to have strict assurance of either no intrusion or of no improper use of the information on your network. However, you have the possibility of detecting these events. This is done by auditing.

The aim of auditing is not to log the usage of CPU, or of printers. This is accounting, done for the purposes of charging users for system use (see `acct` command). The audit is meant to detect abnormal system usage by logging all security events and is very similar to the use of surveillance video cameras with very long recording tapes.

9.4.5.1 Security Event

A security event can be defined by three parameters (subject, action, object):

- Subject is a process represented by its different *uids*, real and effective both being important.

- Action is typically a critical system call such as FILE_Open, FILE_Write, USER_Change, SRC_Start, INSTALLP_Inst, DEV_Create, LVM_VaryonVG, and so on.
- Object is generally passive: a file such as /etc/passwd or /bin/passwd, or a (pseudo-)device such as a floppy disk, memory or a logical volume.

A good audit system must allow you to select by subject, by action, by object, or preferably by any combination of the three. AIX allows you to do this.

9.4.5.2 Audit Analysis

The most common error in auditing is to log too much data because this overloads the CPU and wastes file system resources. Besides, if the logs are too large, you don't find the time to analyze them.

In addition to the classic but critical analysis of the /var/adm/wtmp file in UNIX, you can log the different audit event classes in separate files. This log must be converted into an ASCII representation readable by the audit administrator. The best way to detect unauthorized activities is a statistical analysis, because such attempts at intrusion are infrequent. UNIX provides many tools for performing a good analysis, such as the awk interpreter.

9.4.6 Software Audit

When you receive new software that you have some doubts about (or even if you don't), for example a non-installp formatted software package (see 9.4.4.2, "Software Check" on page 134):

- Don't execute it with superuser privileges
- Audit its behaviour on critical access with the watch command

This command runs only if audit is not up on the system because it uses the same pseudo-driver, /dev/audit. For this reason and because you need to have superuser authority to run watch, you are advised to run this new software on a standalone machine.

The watch command outputs the audit trail (ASCII format) in a file that you specify. Following the execution, you can analyze the different security events of this software such as occurrences of the security file being opened.

9.4.7 Conclusion

You now have some idea of the extensive list of tools and techniques available to address security the AIX environment. It is easy to get lost in this discussion, and try to implement any or all of them without much thought. However, if you return to the original theme of this chapter, you will recall that we discussed the importance of first developing a strategy regarding system security.

However, once you have mapped out your strategy, the many available tools can help ensure that your system and your data remain both safe and secure.

Chapter 10. Wide Area Network Security

10.1 Internet and Private IP Networks

10.1.1 Information Exchange

Internet Protocol (IP) is a standard protocol for exchanging electronic information between people. This electronic exchange is not only limited to E-mail but includes all services as for example: file transfer, remote connection, WEB bulletin board service, and many others. The success of this protocol is its simplicity, its easy installation and use.

From the personal computer to the mainframe, the Internet Protocol is the most widely used network protocol. You can use it with a local area network, or a wide area network. Many organizations have developed a private IP network, sometimes with already complex existing networking connections between their LANs. The term organization means any group which gathers people together, such as an agency, university, firm, factory, company, and so on. These organizations need to exchange data with many other organizations in the world.

10.1.2 Internet

The world area network, which IP connects many organizations with, is called the Internet. The Internet is not a centralized organization; the National Science Foundation (NSF) who manages Internet is only in charge of managing IP addresses and naming subnetworks on the Internet. The routing is done by many Internet providers, private service companies who establish connections between them and build the base of the network. Each organization is bound to one provider and so may communicate with all the other organizations.

In the past, the Internet was suited to data processing and scientific pople, using unfriendly interfaces such as `telnet` or `ftp`. Today with the WEB concept (Mosaic), everybody who knows how to strike a keyboard and to click with a mouse, may use a very friendly Mosaic client (MacIntosh, Windows, OS/2 Warp, or Motif) to navigate in the WEB. WEB is a concept of hypertext and hypermedia bulletin board developed by NCSA.

10.1.3 Some Numbers

Currently, the number of private IP networks is more than 30,000, and NSF estimates more than 1,000,000 machines are connected to the Internet. Many machines are hidden by firewalls, therefore, the number is much greater. It is also difficult to estimate the real number of users on the Internet because it is not possible to know the internal structure of a private IP network. The traffic volume on the Internet has increased at such a rate that people talk about it in terms of an explosion growth. In fact, the growth is exponential and you need a logarithmic scale to represent the growth rate.

© Copyright IBM Corp. 1993,1994

10.2 Security Issues

10.2.1 Introduction

This general interconnectivity raises many security problems, because each organization has to protect its own private data. It also must protect access to the machines inside private networks against abusive external use. To achieve this, their private network is connected to the Internet by only one gate as shown in Figure 104.

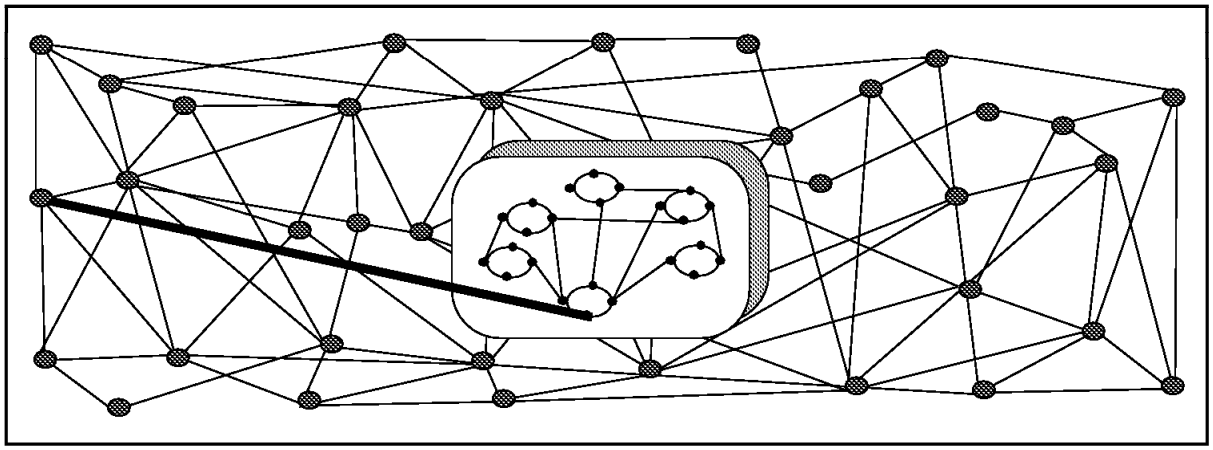

Figure 104. Internet and a Private IP Network

Since you just have this unique path, you could control all the traffic through this gate. We call this gate a firewall.

Depending on what your security requirements are and how much money you have, you can develop different firewall strategies. The most important advice is to define a simple strategy. On the other hand, security requirements are not to prevent internal users from using Internet facilities but to allow users of your organization to use all the Internet services without threatening the organization. It is even possible to integrate new data processing concepts as the nomadic computer and to allow internal users on external sites to use the organization facilities with the same conditions of security.

10.2.2 How to Create a Firewall

There are different products available to create a firewall. You can use some domain public software such as TIS or a private company's firewall, such as the SNG firewall from the IBM NetSP family. TIS has an elaborate firewall toolkit and documentation. You can request to be included on their E-mail list at fwalls-users-request@tis.com. Before you create your firewall, you might want to read the book, "Firewalls and Internet Security" from Addison-Wesley, or get the helpful white papers from TIS by way of *anonymous* ftp at the *ftp.tis.com site*. For SNG see 10.6, "Secure Network Gateway" on page 143.

10.3 The Firewall Concept

10.3.1 Permit or Deny

When you have to define your firewall strategy, you may think it is sufficient to prohibit everything which presents a risk for the organization and allow the rest. However, because of new attack methods, you need to anticipate how to prevent these attacks. Generally, it is too late to fix the damage and the prevention techniques can be expensive. The classic solution with a screened router is not sufficient today to insure security, because it is too low a level of communication protocol security.

A better strategy is to permit only the applications you have tested and have confidence in. If you follow this strategy, you have to exhaustly define the list of services you must run on your firewall. Each service is characterized by the direction of the connection (from in to out, or out to in), the list of users authorized, the list of machines where a connection can be issued, and perhaps the range of time of day you authorize this service.

10.3.2 Firewall Implementation

Generally to establish a connection through a firewall, you achieve it with an intermediate highly-secure machine which we called a bastion. This bastion has daemons to serve all the clients. Two techniques are possible: you can change the client program, giving it the same user aspect as the standard client, or you can change the server program, giving it new functions to achieve your security requirements. The more you modify classic daemons, the more you freeze the evolution of these.

10.3.3 Tunnel Concept

The best way is to use a tunneling method. Let′s say you have the sendmail daemon installed. This program is generally too big to be confident there are no bugs inside which threaten security. Therefore, you simulate a sendmail daemon which only understands the basic subcommands of the SMTP protocol and relays these subcommands to a true internal sendmail daemon on another machine behind the firewall. Therefore, direct attacks to sendmail daemon don′t work. If an attack can access the internal mailer, this may open a breach on the machine which hosts the internal mailer, but this one is not directly accessible by the Internet, and therefore the risk of intrusion is reduced.

10.4 Risks

10.4.1 Introduction

To understand how a firewall works, look at this example. Imagine a building where you want to protect the access and to control people who enter in. You define in the architecture of the building a unique lobby as the only entrance point of your building. In this lobby, you have some hostesses to welcome, some caretakers to watch over, some video cameras to record, and some badge readers to authenticate people who enter the building.

This works very well to access a private building. But if a non-authorized person succeeds in entering, no matter how they get in, there is no way to protect the

building against any actions from this person. However, if you supervise the movement of this person, you have a chance to detect any suspicious behavior from them because they don't know the bulding.

10.4.2 Firewall Definition

A firewall is a way to achieve shell security: welcome users, authenticate authorized users, refuse others, log all events and sometimes bring attention to suspicious events.

10.4.3 Compartmented Organization

If one firewall is not enough to comply with your security requirements, you may create compartments, which are firewalls around certain areas. This really reduces the risks because if there is a chance for the user to pass the first firewall, there is less risk the user will succeed in passing the second firewall, except if he has help from authorized people, but this is another issue.

10.5 Different Strategies

10.5.1 Basic Components of a Firewall

10.5.1.1 Screening Filter

The first and most used strategy is to separate the private IP network from the Internet by inserting a router between, as shown in Figure 105.

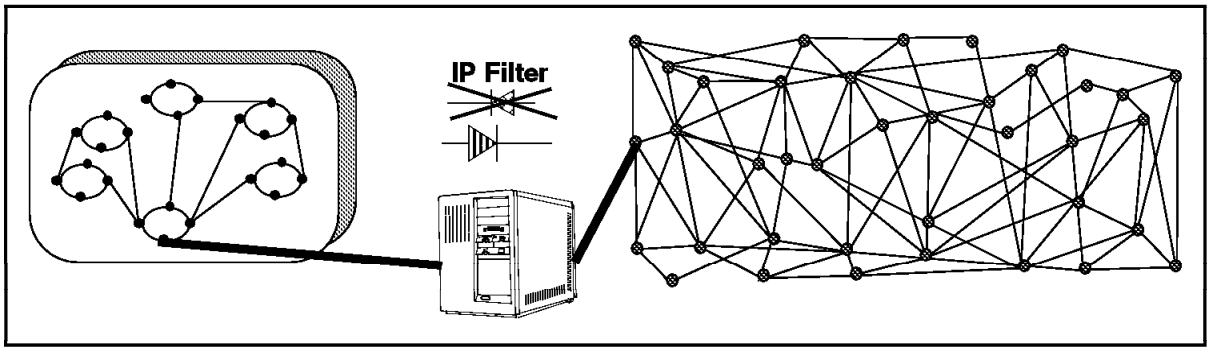

Figure 105. Screening Filter

This router filters all IP packets passing through and is called a a screening filter. This way you can prevent access to machines or to ports in the private network and also do the reverse; prevent an inside machine from accessing the Internet. But if you do this, there is no way to control what's happening at the application layer. If you allow a `telnet` daemon on several private machines behind the router, you have to manage security requirements on all these machines. A screening filter is a very useful tool to use in conjunction with other tools as a proxy server or application filter.

10.5.1.2 Bastion

A bastion is an intermediate machine where the IP forwarding is broken, which means no IP packet can go through this machine. If the routing is broken, since you are on this machine, you can use all the services from the Internet and all the services from the private network. Therefore, all the users who have an account on the bastion, with a double identification one for the bastion and one for the remote host, can use services on both the networks, as shown in Figure 106.

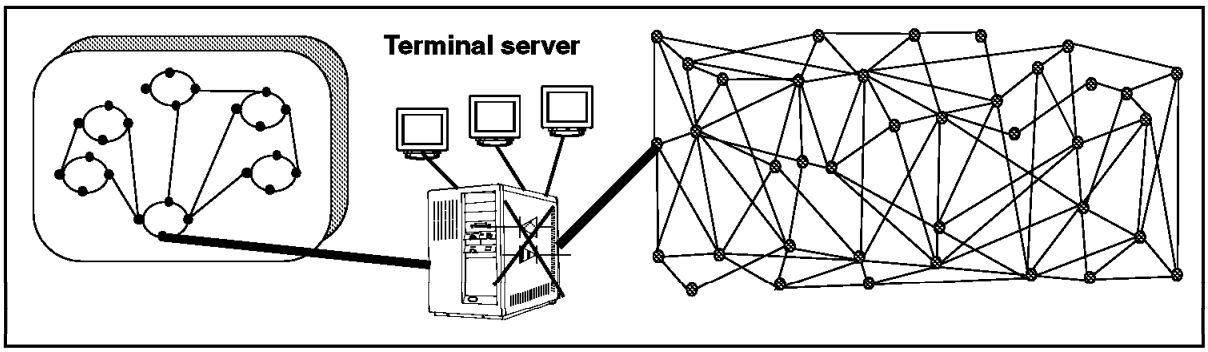

Figure 106. Bastion

But this has some disadvantages, because the bastion has to support many users some of who may have a simple password, which may allow an external hacker to impersonate a user and get in the private network. Besides this security point, a great number of users require a big machine. To avoid having users on this machine and to reduce load of this machine, there is another concept developed with the SOCKS server. See the SOCKS server in the SNG package.

10.5.2 Different Firewall Architectures

10.5.2.1 Dual-Homed Gateway

One good solution is to combine a screening filter and a bastion, as shown in Figure 107.

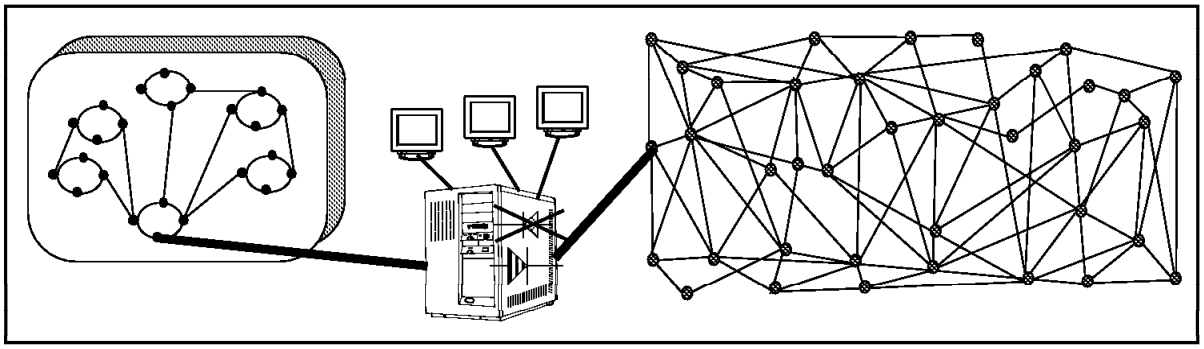

Figure 107. Dual-Homed Gateway

In this case, you can protect the dual-homed gateway from external attacks with filtering. For example, if you forbid external access to the telnet daemon, you reduce the threat of an external attack. If you have some nomadic machines which need to connect inside the private network, some techniques using smart card allow some secure external connections. Nevertheless, there are a great number of ports used by well-known services such as telnet to establish connections and if you block all these ports, you classic daemons don't work. Now if a hacker, no matter how he succeeds, runs a shell daemon on one of these ports, a shell daemon is just a telnet without login, he spends a long time before he finds the hole, because it is not easy to detect this kind of daemon. Besides, it is very difficult to administer a bastion that is too complex, and if you don't have something to protect your bastion, all of your security is lost.

10.5.2.2 Bastion Behind a Screening Filter

A better solution is to use the same solution but use two machines as shown in Figure 108.

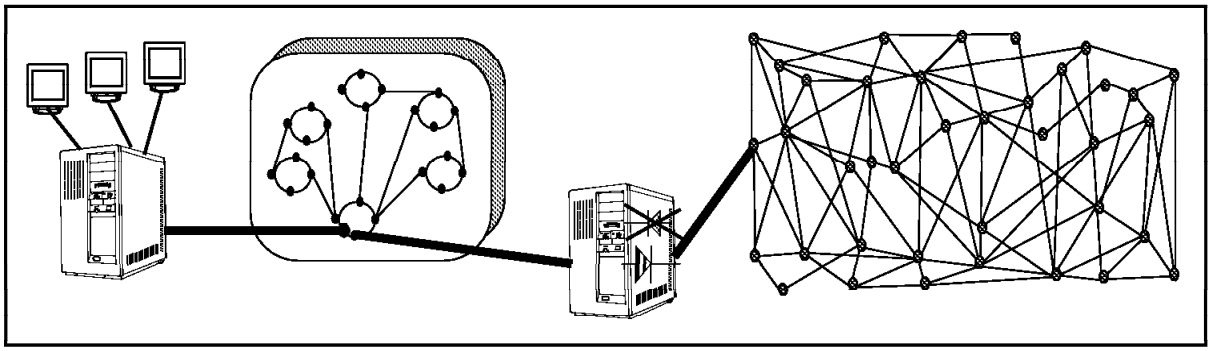

Figure 108. Bastion Behind a Screening Filter

In this configuration, the bastion is protected from external attack by the screening filter. This one is a very simple router without a daemon inside. This implies it is very hard to break into this machine. In this manner, you could hide the structure of your private organization, and protect a complex daemon such as sendmail.

10.5.2.3 Screened Subnet

If your organization has many people and it is possible that complicity may happen between an internal user and an external user, you must protect the organization against itself. The last but most expensive solution is to define a network between your private network and the Internet. This network is composed of two screening filters and one or several bastions as shown in Figure 109 on page 143.

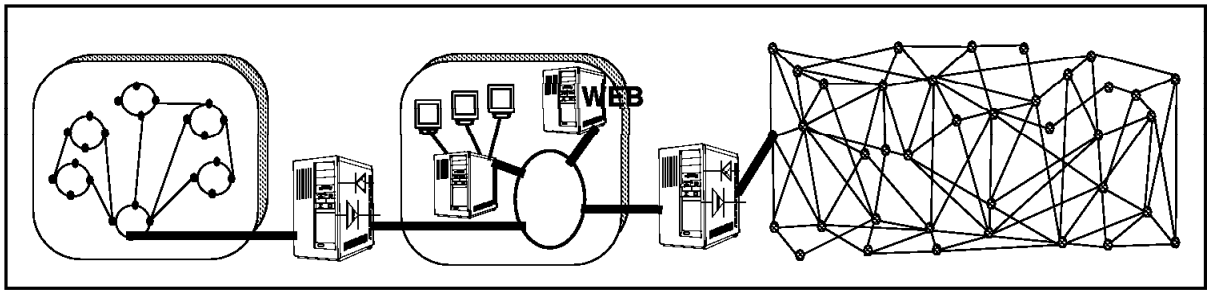

Figure 109. Screened Subnet

The architecture is very simple and each machine on this subnet performs only simple task(s), depending on the number of bastions you have.

10.5.3 Conclusion

The choice of your firewall architecture depends on your security requirements and also on the size of your organization. A small organization may choose a bastion behind a screening router and a bigger one a screened subnet. Your firewall architecture also depends on the software you choose, and many solutions exist today. One of these is called the Secure Network Gateway.

10.6 Secure Network Gateway

10.6.1 Introduction

Secured Network Gateway is a product from the NetSP family, but it can work without the other NetSP products. SNG is a tool box you can use to implement all the functions of different firewall architecture: screening filter and bastion. Once you choose your architecture and your security strategy, you use the needed tools you get from SNG. The administration of SNG is included in SMIT and is very user friendly since you may use the ASCII SMIT interface or the graphic MOTIF SMIT interface. The logging method used by SNG is the UNIX classic syslog daemon, so it is very easy to process logins in the same manner you already process for others daemons. SNG divides the world into two networks, the secure network and the non-secure network. A non-secure network is, for instance, the Internet. A secure network, for instance, is the private IP network. The following describes the different tools avalaible in the SNG kit.

10.6.2 SNG Tools

The SNG tool box is composed of:

- Proxy server
- SOCKS server
- IP filter
- Specific services such as domain name service and mail handling

All the tools using the complex filter configuration file have some useful tools to test the validity of the information inside. There is a set of testing tools to insure quality of the configuration files used by the proxy server and by the IP filter.

10.7 Proxy Server

10.7.1 How it Works

10.7.1.1 From Inside

This tool allows an internal user, using classic client commands such as `telnet`, to access the non-secure network. For example, the user uses `telnet` to access a bastion. This allows two connections, one to the secure network and one to the non-secure network. In this case, the bastion doesn't route the IP packets. When the user accesses the bastion, he needs to be authenticated on the bastion. If he succeeds, he has to use a new subcommand `telnet` to access the desired machine on the non-secure network.

10.7.1.2 From Outside

This method is usable also from the non-secure network to the secure network, but this raises other important security problems. If you use the Internet to connect to the bastion, you have to enter your login name and password to be identified. But, perhaps your login name and password go through gateways where some hacker is looking for login names and passwords at the IP level protocol, using iptrace on a UNIX machine or useful tools from the PC environment. If he catches yours, he may impersonate you and get in the organization with your identity. In this scenario, with the proxy server, you can use a more sophisticated tool of authentication such as a security identity card. This mechanism ensures a key, used as a password, is not reusable for another connection. Therefore, with this added mechanism, users from your organization may use your internal information system, even if they use a nomadic computer from outside.

10.7.2 Implementation

Users are recorded in the classic operating system security files with very restricted conditions. For example you have two new shells: a restricted shell and a shot shell. When you record them, you use new user administration commands or new SMIT menus. The AIX security files are useful to record all the new features bound with each user for this firewall aspect. The proxy daemons available are a `telnetd`, `ftpd`, and so on. The proxy `ftp` daemon is not usable from outside. The reason is, `telnet` is usable and in this case, you may `telnet` from outside to inside and then `ftp` from inside to outside. This reduces the way a hacker from outside may easily get information.

10.7.3 Advantages/Disadvantages

Once connected, the appearance and the behavior are unchanged. But this method needs to have a double connection, one on the client machine and one on the bastion machine. Nevertheless, this method is very useful because you don't change the client program on the client machine. Therefore, once you have installed your firewall, every user recorded in the firewall can have access to the non-secure network without any software installation in their machine.

10.8 SOCKS Server

10.8.1 How it Works

The SOCKS philosophy is the opposite idea to achieve a bastion. Now you need to have new clients, we call them SOCKSified clients, and you have a unique daemon, SOCKSd, on the bastion to establish connection between the non-secure and the secure networks. For example, you use a new `telnet` command called `rtelnet`, this is a SOCKSified `telnet` client. The argument of `rtelnet` is the same as if you are directly connected beyond the firewall, an environment variable keeping the name of the bastion who hosts the SOCKS daemon. If the SOCKS daemon allows you to, and if the remote host responds, your session begins on the remote host the same as if you use a classic `telnet` command with direct access to the remote host.

The daemons that come with SNG are:

- `rtelnet`, a remote connection client
- `rftp`, a remote transfer file client
- `rfinger`, a remote finger client
- `rwhois`, a remote rwhois client

10.8.2 Implementation

The SOCKS daemon uses a configuration file, /etc/socks.conf, to allow or deny connections through the bastion.

Each line of this file defines a rule that controls access through the bastion with:

- User ID or list of user IDs
- Address of client and mask (subnetting issue)
- Address of remote server and mask
- Operation field and port to define what service or what range of services
- Command to execute, very useful to log or to alert

For insuring the identity of a user, the SOCKS daemon calls an ident daemon on the client machine. The ident server answer the identity of the calling user, following the RFC 1413, this one obsoletes RFC 931. Once a connection is established, the SOCKS daemon transfers the data between the client and the server, this is routing at the application level.

10.8.3 Advantages/Disadvantages

The SOCKS daemon has become a new standard for firewall purposes. The advantage is the transparency of bastion. You only need to configure an environment variable in the general /etc/profile file or to define a list of SOCKS servers in the /etc/socks.conf file. Only the name of the client command is changed. The usage of the ident daemon prevents a user from using accounts other than his, as is possible in the proxy server. You may control with good granularity, the access to the non-secure network. This is a good way to define a user profile. But the SOCKS server is not meant to accept user′s connection from outsie, though it is possible to do a tunnel connection between the SOCKS server and an outside well-known machine. However, if one day a user knows

the name of this remote machine, they may impersonate this machine and have direct access to the secure network. The final advantage is the ability to SOCKSified a new client for specific purposes. For example, you could get from NCSA, a Mosaic SOCKSified client.

10.9 Filters

10.9.1 How it Works

The filter is a very useful tool to filter packets at the IP level. The machine is required to have two network interfaces, one declared secure and the other declared non-secure. The filter acts between these two adapters.

10.9.2 Implementation

The filter uses the /etc/security/fwfilters.cfg configuration file where each line defines a filter using the following criteria:

- Source IP address and mask (subnetting issue)
- Destination IP address and mask (subnetting issue)
- Type of IP protocol: TCP, UDP, and ICMP.
- Source port or port range
- Destination port or port range
- Direction of the IP packet: inbound or outbound
- Network interface (notion of secure and non-secure adapter)
- Routing, is accepted for firewall only or for forwarding

The first line that matches characteristics of an IP packet, stops scrubbing of the configuration file. The action bound with a line, permit or deny, is applied.

10.9.3 Advantages/Disadvantages

This is a complementary tool of the other tools, it is necessary to use it to protect the firewall itself and to avoid undesirable traffic through the firewall. It may be used as a screened router to protect a screened network.

10.10 Specific Services

10.10.1 Domain Name Service

10.10.1.1 How it Works

From the outside view, the name server on the firewall only knows itself and never gives information on naming inside the private IP network. From the inside view, this name server knows the Internet network and is very useful for accessing any machine on Internet by its name.

10.10.1.2 Implementation

To use it you have to configure it and to define:

- Non-secure domain name
- Secure domain name
- Non-secure name servers
- Secure name servers

Once you have defined these it works because it is not itself a name server but a way to permit or deny access between name servers.

10.10.1.3 Advantages/Disadvantages

This specific tool is useful to hide the internal organization structure.

10.10.2 Mail Handling

10.10.2.1 Implementation

You have to define in the firewall name service, the name of the machine which hosts the secure mail server. It works because it is not itself a mailer daemon but a way to tunnelize mail data between secure and non-secure networks.

10.10.2.2 Advantages/Disadvantages

This specific tool is useful in hiding the internal organization structure and protect the SMTP daemon from direct attack.

10.11 Conclusion

Secure Network Gateway is a very useful toolkit to design your firewall. Installing and configuring SNG is quite easy, since all administration utilities are integrated in SMIT. Keep this in mind for your firewall and make it simple.

Chapter 11. Storage Management

In this chapter you get an overview about storage management, a short description of available products for archive and backup functions and an overview of tape, optical media and disks.

11.1 Overview

In todays open client/server environment, data availability and integrity have become key concerns in both, the technical and the commercial market. These concerns are rooted in the movement of mission critical data to distributed workstations and the continued growth of data throughout the enterprise. Backing up this data without the use of comprehensive automated tools is time consuming and unreliable. Users need mass storage solutions which provide cost effective access to data across the networked environment.

Storage Management systems are designed to provide the user with data availability and disaster recovery of files and data. They typically provide three primary functions:

Backup and restore
This enables the system administrator to manage distributed resources in a way that data security is ensured. These solutions can be configured to provide centralized administration, automated backup and automated disaster recovery.

Hierarchical Storage Management (HSM)
These solutions optimize storage resource utilization by automatically locating data to the most efficient storage media as defined by the user availability requirements. HSM applications automatically migrate infrequently-accessed data from online storage media to less expensive media, such as tapes and/or optical media. An HSM solution can be configured to provide online storage of large amounts of data at costs typically associated with online media.

Media management
This provides support for different media, like tape and optical storage drives and libraries.

11.2 IBM Storage Management Products

Historically the UNIX operating systems data management capabilities were limited. Many system reliability functions were not initially designed into the product.

Today, UNIX applications are used in commercial markets, where transparent data management and availability tools are a must.

IBM offers a wide range of automated storage management solutions. Currently are available:

- AIX File Storage Facility/6000
- Unitree for AIX/6000*

© Copyright IBM Corp. 1993,1994

- ADSTAR* Distributed Storage Manager
- Legato NetWorker for RISC System/6000

You will get an overview of these solutions in the following pages, execpt for the Legato NetWorker for RISC System/6000 which is already described in 5.5.3, “Legato NetWorker” on page 55. For detailed documentation on these products see “Related Publications” on page xx.

11.2.1 AIX File Storage Facility/6000

AIX File Storage Facility/6000 (FSF/6000) provides automatic client disk space management and file migration to any NFS server. FSF/6000 creates a personal data cache on the client machines local disk storage (*client cache*) and automatically moves files between the client cache and a central data cache on an NFS servers storage (*central store*). These movements of files between the client cache and central store are transparent to the user.

FSF/6000 automatically:

- Copies files from the central store to the client cache, when they are needed.
- Copies file changes back to the central store to maintain a synchronized file system (*Automatic Write-Back*).
- Deletes files from the client cache to free up space, when they are not longer needed (*File Pruning*).

FSF/6000 allows you to customize and fine-tune parameters for the Automatic Write-Back and File Pruning operations, such as:

- Maximum time a file can remain dirty (after this time, a modified file is copied from the client cache to the central store to have consistent copies).
- Minimum age of a file before it can be pruned (describes the amount of time a file can remain in the client cache after it has been backed up to the central store).

There are more parameters tuneable to get better performance and disk space usage. There are also some commands available, which allow users to list information about FSF/6000 files, to start write back or prune operations and to pin files to prevent pruning. Detailed information is available in the *AIX File Storage Facility/6000 - Installation, Planning and User's Guide*.

FSF/6000 is a client storage management software, which works with a standard NFS server and needs therefore no special server software. It provides no backup/restore and archive functions, but in conjunction with a UniTree server, (see 11.2.2, “Unitree for AIX/6000” on page 151) FSF/6000 provides a continous and transparent management of files from the disks of AIX/6000 clients to archival media managed by UniTree.

FSF/6000 provides the following features:

- Reduces client disk space requirements
- Reduces or eliminates the need for local backup/restore procedures and end-user involvment in these procedures
- Enhances performance, since files actually reside locally when they are needed

- Allows users to continue to work with files residing locally, if the connection to the server fails
- Provides extended disk capacity when used with a server-based file management system, as UniTree for AIX/6000

11.2.2 Unitree for AIX/6000

The IBM UniTree for AIX/6000 Licensend Program Product provides hierarchical storage managment on the RS/6000 to archive data. UniTree manages simultaneously two different layers of storage media and presents itself as a standard file system to the user. This mean, the user see and accesses all files, even though migrated to tape or optical library, as if they are residing on the server disk.

UniTree provides three main functions for automatic storage managment:

Migration This is copying a file from a higher level of the UniTree storage hierarchy (disks) onto the next lower level (tape or optical) of storage within a site-defined period after it is first moved onto the higher storage. This means that less frequently used files are migrated to less expensive media, while critical and frequently used files retain on the highest level of storage for immediate access.

Purging This selective deletion of files from a higher storage level, when that higher level storage usage exceeds a site-defined *High Watermark* threshold, and space is required for new or cached files. Files marked for purging are those which first, have migrated to the next lower level of storage and second, are determined to be eligible for purging based on the UniTree purging algorithm for that level.

Caching On demand restoration of a file from lower level storage to the servers hard disk (highest level storage).

UniTree uses a virtual file system, which can be accessed like a remote file system by using either the File Transfer Protocol (FTP) or the Network File System (NFS) protocol. Using NFS or FTP is the only way to access data in the UniTree file system and therefore you need to install the FTP and NFS server daemons provided with UniTree. This servers can be accessed by the standard FTP and NFS clients.

It is important to note, that there is currently no way, to access both the AIX and UniTree file systems concurrently via NFS. This implies, that you cannot export an AIX (JFS) file system on a file server running UniTree.

It is also important to understand, that files which are required to be migrated to tape or optical storage, *must* physically reside in the UniTree file system. In other words, if you want to migrate files to tape or optical media, you have to create or copy them into the UniTree file system using either NFS or FTP.

For reliability, UniTree allows up to 15 copies of a migrated file and can cause each copy to be placed on a separate storage unit. UniTree has a database, called *Name Server Database* implemented, which is used to maintain the UniTree virtual file system. These database is the key to all datas maintained by UniTree. Therefore UniTree holds a shadow copy of this Name Server directory on a separate magnetic disk. Every transaction affecting the Name Server

directory structure is written to both copies simultaneously. In case of a disk crash affecting one of the copies, service will be interrupted briefly while another copy is made, and the Name Server is reinitialized. Other critical structure files also maintain shadow files.

In addition to shadow copies, backup copies of directory structures are taken periodically. Duplicate copies of file metadata are maintained on magnetic disks. When media or device error occours, UniTree acts to protect and restore data without disrupting operations, ensureing a graceful recovery.

For the case, that UniTree is not able to resolve problems automatically, the system manager is provided with an *Disaster Recovery* procedure, what implies, that it is neccessary to backup the UniTree file system metadata.

UniTree can be used with a cluster of RS/6000s running HACMP/6000. UniTree currently supports both the High Availability component and the Concurrent Resource Manager component of HACMP/6000.

UniTree supports magnetic disks, automated tape libraries, optical disk systems and manually mounted tape drives. At this time you cannot use tape and optical media simultaneously!

11.2.3 ADSTAR Distributed Storage Manager

IBM′s ADSTAR Distributed Storage Manager/6000 is a highly reliable, high performance, network based backup and archive product. A RISC System/6000 is used to provide backup and archive services for client systems, such as PCs, UNIX workstations and LAN-based UNIX and PC servers.

ADSTAR Distributed Storage Manager handles a pool of storage devices, which can be disks, tapes and optical media. These media are hierarchically organized without a limit on the number of the hierarchies. ADSM automatically places backup and archive data in this storage pool using the administrator-defined policies.

The information about the files and data in this storage pool is maintained in a database and a recovery log file. These database and the recovery log file is automatically mirrored to be sure, that in case of a disaster all data stored in the storage pool can be restored to a consistent state.

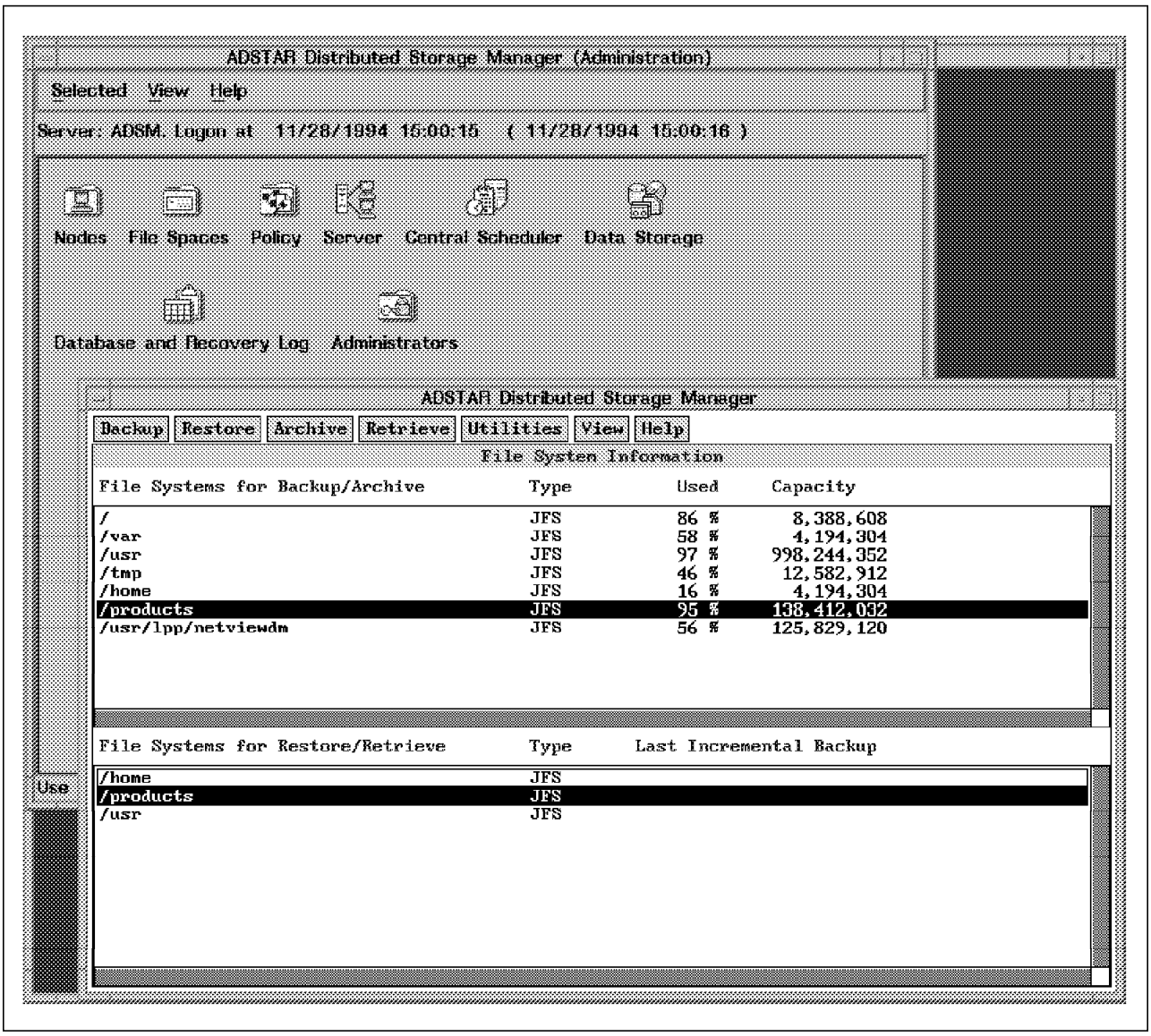

Figure 110. ADSM/6000 Server Administration and Client Graphical User Interfaces

Client systems can backup/restore and archive/retrieve their files and data into/from this storage pool by using a Backup-Archive client program. This Backup-Archive client communicates with the server program according to user or administrator defined schedules or by user request, and handles all the necessary tasks to backup/archive the data. In Figure 110, the second screen shows the Client Graphical User Interface with its backup/restore and archive/retrieve functions.

For the following systems, the Backup-Archive client program is available:

- Apple** Macintosh** Operating System
- AIX
- Bull BOS/X**
- DEC Ultrix
- DOS
- Hewlett Packard HP-UX

- Microsoft Windows
- Novell** Netware**
- OS/2
- SCO** UNIX 386/SCO Open Desktop
- Sun Microsystems SPARC**/Solaris**

The administration of the ADSTAR Distributed Storage Manager System can be done from any workstation in the network and can be organized in different classes, such as allowed to change policies, allowed to control storage resources, and so on (see Figure 110 on page 153, first screen). This allows to spread the administrator functions to more than one person and can also be done from every machine in the network.

There is also an Application Programming Interface (API) available, which allows an application to communicate with an ADSM server for the purpose of sending, querying or getting data from the ADSM storage pool.

The ADSM API is currently supported on following platforms:

- AIX
- HP-UX
- Micrososft Windows
- SunOS**

Besides AIX, the ADSTAR Storage Management is available on MVS, OS/2 or VM for storage and media management. ADSM is the successor product to the IBM Workstation Data Save Facility/VM (WDSF/VM) program product and can be used with existing WDSF/VM clients.

Since all ADSM servers share a common code base across platforms, migration between different server platforms is a simple procedure of exporting the data and metadata from one server and importing it to another platform.

11.2.4 Legato NetWorker for RISC System/6000

Legato NetWorker for RISC System/6000 is a backup/restore product, which has no archive functionality.

This product is already described in 5.5.3, “Legato NetWorker” on page 55.

11.2.5 Comparison of the Products

File Storage Facility/6000 is a client migration system, which enhances the performance against a normal NFS mount and gives you the ability to continue your work, even when the network is temporarily unavailble. FSF/6000 will then synchronize the files, after the network is available again.

UniTree for AIX/6000 provides a virtual file system, which can be accessed via FTP or NFS and is therefore transparent to the user. But, all files which should be archived have to reside physically in the UniTree file system.

FSF/6000 and UniTree for AIX/6000 are complementary products. FSF/6000 provide space management and migration of client files to the UniTree server file system and UniTree will then manage those files from the server disks to

optical disk or tape hierarchy. With using this products together, the client disk space is expanded to the size of the automated library.

ADSTAR Distributed Storage Manager is a backup/restore and a archive/retrieve system, which offers the ability to use different storage hierarchies for backup or archive. The server part is available within MVS, VM, AIX and OS/2 and the client is available also for non IBM systems. ADSM provides a very user-friendly Graphical User Interface on AIX and OS/2. FSF/6000 and UniTree for AIX/6000 functions are accessible only through the AIX command line interface.

Legato Networker for RISC System/6000 is a backup/restore system which enables you to automate these functions in a networked environment.

The following chart provides a detailed comparsion.

Table 1. Comparison of Storage Management Products

Function	ADSM	Legato Networker	UniTree	FSF/6000
Client Backup/Recovery				
Client directed	Yes	Yes(#)	No	No
Server scheduled	Yes	Yes	No	No
Cron scheduled	Yes	No	No	No
Client Archive/Retrieve				
Client directed	Yes	No	Yes	No
Server scheduled	Yes	No	No	No
Cron scheduled	Yes	No	Yes	No
Automatic Migrating/Cacheing of files (Space Management)				
On Server	Yes	No	Yes	No
On Client	No	No	No	Yes
Number of Storage Hierarchies	Unlimited	2	2	1
Data Compression				
Client to Server	Yes	Yes	No	No
On Server	Yes	Yes	No	No
Supports Disk, Tape and Optical concurrently	Yes	Yes	No	No
Storage transparent to User	No	No	Yes	Yes
OPEN NFS Protocol between Client and Server	Sockets	No	Yes	Yes

(#) indicates recovery only

11.3 Removable Media

There are currently two technologies for removeable media available, tape and optical. Both are cheaper than disks but also slower. It is depending on the application, which removable media is the best to choose. For instance, tape technology is the best for a backup/restore application while optical storage

provides random accessibility, which is neccessary for applications such as online documentation or multimedia.

The following table will allow you to choose a technology for your application. Be advised that this mapping is only generally true and there could be reasons, such like given technologies, to choose others.

If you run following applications, we recommend the use of tape technology:

- Backup/restore
- Archive
- Data interchange
- Software distribution
- Hierarchical storage management

You will choose optical technology, when you run:

- Online documentation
- Multimedia
- Storage for infrequently used data
- Microfilm replacement
- Software distribution
- Hierarchical storage management

11.3.1 Tapes

Tape storage media is divided into two classes: drives and libraries. Tape drives usually can accommodate one tape, but there are drives with an automatic cartridge loader available, which can accommodate more. Tape libraries are normally used, when a tape drive can′t meet the capacity requirements. Tape libraries also offer the ability to access tapes in an unattended mode.

11.3.1.1 Drives

The following table will show some characteristic of availabel tape drives.

Table 2. Tape Drive Characteristics

Model	Drive	Native Capacity (GB)	Data Rate (KB/sec)	Save Rate (GB/hour)
5GB 8mm	7208-011	5	500 (1MB/sec compressed)	1.8 (3.6 compressed)
2.3GB 8mm	7208-001	2.3	245	0.88
2GB 4mm	7206-001	2	183(366 compressed)	0.66 (1.32 compressed)
4GB 4mm	7206-005	4	400(800 compressed)	1.44 (2.88 compressed)
1.2GB 1/4″	7207-012	1.2	300	1.08
1.2″ Reel	9348-012	160MB@6250bpi 40MB@1600bpi	768@6250bpi 208@1600bpi	2.76

Note in the table, that compressed data rates and capacities assume two time average compression. Also note, that all drives except the 2GB 4mm drive, are also offered as internal features.

11.3.1.2 Libraries

Tape libraries are used, when a tape drive can't meet the capacity requirements or when an unattended operation of the tapes is needed.

The following table shows characteristics of available tape libraries.

Table 3. Tape Libraries Characteristics

Library	Model	Total Capacity (GB)	Number of drives	Data Rate (MB/sec)	Average exchange time (sec)
10 Cartridge 8mm	0840-001 (EXB-10e)	50	1	0.5	49
54 Cartridge 8mm	0572-001/002 (LS/380L)	270	2	1	49
210+ Cartridge 1/2″	3494-L10	168+	1-8	3	7-17
6,440 Cartridge 1/2″	3494-L20	5,152	4-16	2.5	7-17

Note, that compressed values for 8mm systems assume 2:1 ratio and 34XX systems assume 3:1 IDRC ratio.

11.3.2 Optical

Like tape storage products, optical ones are also divided into two classes: drives and libraries. The differences between these classes are the same as the one for tape storage products.

11.3.2.1 Drives

Table 4 shows technical parameters of different optical drives.

Table 4. Optical Drive Characteristics

Drive	Model	Capacity	Read Data Rate	Write Data Rate
CD-ROM	7210-001	600MB	150KB/sec	
CD-ROM	7210-005	600MB	330KB/sec	
Rewritable Optical(1)	7209-001	595MB	620KB/sec	207KB/sec
Rewritable Optical(2)	7209-001	650MB	680KB/sec	227KB/sec
Rewritable Optical(1)	7209-002	1.19GB	1.4MB/sec	467KB/sec
Rewritable Optical(2)	7209-002	1.3GB	1.6MB/sec	433KB/sec

(1) for 512 bytes/sector media
(2) for 1024 bytes/sector media

11.3.2.2 Libraries

You will use an optical library instead of a single drive, when having the same reasons as for a tape library (capacity, unattended operation). See Table 5 on page 158 for a comparison of technical paramters of the different libraries.

Table 5. Optical Libraries Characteristics					
Library	**Model**	**Total Capacity**	**Number of drives**	**Read Data Rate**	**Write Data Rate**
32 Cartridge R/W Optical(1)	3995-063	38GB	2	0.96MB/sec	0.32MB/sec
32 Cartridge R/W Optical(2)	3995-063	40GB	2	1.05MB/sec	0.35MB/sec
144 Cartridge R/W Optical(1)	3995-163	171GB	4	0.96MB/sec	0.32MB/sec
144 Cartridge R/W Optical(2)	3995-163	188GB	4	1.05MB/sec	0.35MB/sec

(1) for 512 bytes/sector media
(2) for 1024 bytes/sector media

11.3.2.3 Optical Positioning

CD-ROM technology is read/only and can't be used for traditional write applications. However, the extremely low cost of the discs, compared to tape or rewriteable optical technologies, makes CD-ROMs ideal for software distribution.

The 3995 Optical Library Dataserver Models 063 and 163 add a new level of price, capacity and performance to the traditional storage hierarchy of magnetic disk and tape. Infrequently accessed information residing on magnetic disk or tape, paper or microfilm is ideally suited for optical storage, offering an opportunity for increased productivity and reduced storage costs.

The 7209 External Rewriteable Optical Drive is a standalone unit which provides online access to a single disk cartridge, and additional storage capacity is achieved by manually inserting another disk drive. The 7209 is intended for a personal workstation storage environment, whereas the 3995 Models 063 and 163 are intended for enterprise-wide needs where gigabytes of online storage is required.

11.4 Disk Storage

Depending on the amount of needed disk storage you can choose between single disks and disk arrays. The following section will give you an overview of available disks and disk arrays.

The primary characteristics to select disks or disk arrays for a specific application are:

- Capacity
- Performance (both sequential and random)
- Reliability
- Costs

It is also depending on the application what configuration should be used, for instance, for a random access application (typical of transaction processing) two 1GB disks will generally perform better than one 2GB disk.

11.4.1 Disks

There are SCSI attached and serial attached disk drives available. The SCSI attached have a capacity between 200MB and 2.4GB, while the serial attached are between 1.07GB and 2GB. The serial can only be used in the 9333 disk subsystem (see 11.4.2.3, “IBM 9333 High Performance Disk Drive Subsystem”).

The disks differ besides their capacity in random and sequential performance. You should choose your disks according to your application needs.

11.4.2 Disk Storage Subsystems

The following section is intended to show the characteristics of available disk subsystems. Following subsystems will be discussed:

- IBM 7203/7204 External Disk Drives
- IBM 9334 SCSI Expansion Unit
- IBM 9333 High Performance seriell attached Disk Drive Subsystem
- IBM 7135 RAIDiant Array
- IBM 9570 HIPPI attached Disk Array Subsystem

11.4.2.1 IBM 7203 and 7204 External Disk Drives

The IBM 7203 External Portable Disk Drive provides storage in high-security environments where the user routinely removes the data from the system, so it can be locked away. The easy removeability and handling of the disk modules can also be used to replicate the system configuration of one machine on several other machines.

The IBM 7204 External Disk Drive provides expansion storage for a single disk drive. This is a practical way to expand storage where one or two extra disks are required beyond what will fit inside a system unit. If more than two expansion disks are required, the 9334 should be considered.

11.4.2.2 IBM 9334 SCSI Expansion Unit

The IBM 9334 SCSI Expansion Unit is used where rack or deskside storage is required. The 9334 provides attachment of SCSI disk drives and selected removable media devices. The 9334 offers the most cost-effective solution for moderate storage requirement.

The 9334 SCSI Expansion Model 500 or 501 are as deskside and the Model 010 are 011 as rack mounted available.

11.4.2.3 IBM 9333 High Performance Disk Drive Subsystem

The IBM 9333 High Performance Disk Drive Subsystem is designed to provide the fastest response time when an application makes a large number of requests for short blocks of data. Transaction processing applications are typical response time intensive applications. Since the 9333 has multiple paths from its controller to the system unit it can be attached to eight systems for high availability or data sharing.

The 9333 storage units are attached to the system unit with a serial interface adapter. This adapter attaches up to four 9333 subsystems, so a single Micro Channel slot can attach up to 16 disk drives for up to 32GB of storage.

11.4.2.4 IBM 7135 RAIDiant Array

The IBM 7135 RAIDiant Array provides the greatest flexibility of any of the RS/6000 storage subsystems. It can be used with one to four processors for a variety of data sharing and availability configurations. It permits more storage per Micro Channel slot and more total system storage than any other solution.

The user can set up the RAIDiant Array to operate in essentially any mixture of RAID levels. In addition to redundancy in the data, the RAIDiant Array has both redundant power supplies and cooling fans and can have a redundant array controller.

Chapter 12. Centralized Problem Management

When your enterprise has many users and machines, you will need your own support center for system management and also for user support. This chapter shows you a way to monitor and track problems of client systems and users.

Let's consider an example of a customer support center processing a customer problem from its detection to its resolution. This customer is supposed to run critical business applications on a network of RISC System/6000 servers. The users are using various kinds of terminals, X-stations or workstations to use those applications. The network is TCP/IP based.

This example will use the following IBM products:

- AIX NetView/6000
- AIX Trouble Ticket/6000

This section contains no detailed information about these products. Some configuration information and scripts are provided in Appendix A, "Problem Determination Sample Source Files" on page 169.

12.1 Using AIX NetView/6000 as Problem Monitor

The error logging mechanism of AIX is very powerful. Monitoring the error log on a critical AIX machine is a very important task. Our experiences in large customer sites has shown that on production machines, *any* error log entry coming up should be treated as a suspicious problem.

A nice feature of AIX NetView/6000 is the SNMP proxy-agent named trapgend, which can be installed on all RISC System/6000 remotely from the central AIX NetView/6000 management workstation. This proxy-agent will automatically send an SNMP trap to NetView/6000 when an error with the flag Alert=True occurs in the AIX error log. Any LPP and the base operating system come with its own catalog of error templates. On an AIX machine you will have 300 to 700 of those templates, which can be viewed with the `errpt -t` command. Very few templates (around 70) have their Alert flag set to True. On a production machine you would typically like to have all the error templates with Alert=True. In Appendix A, "Problem Determination Sample Source Files" on page 169, there is a shell script named errupdate.sh which sets all Alerts to True.

© Copyright IBM Corp. 1993,1994

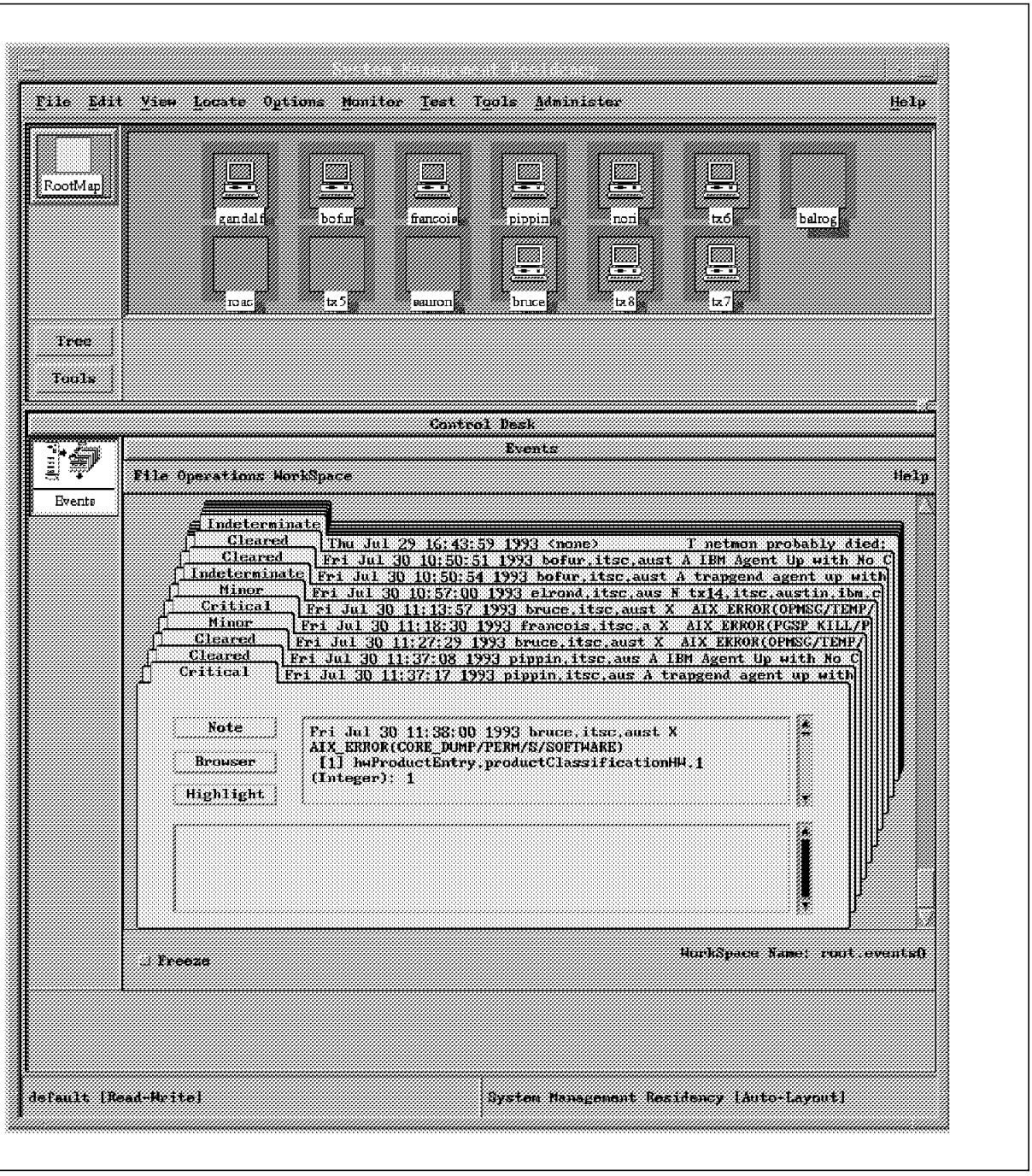

Figure 111. NetView/6000 Windows with an AIX Error Trap Card

On the AIX NetView/6000 side, the traps produced by trapgend are, like any other traps, displayed on the *Events* cards window. You can use all AIX NetView/6000 functions, like filtering, automation and so on. By default, NetView/6000 displays the raw content of the trap in the card which is mainly some hexadecimal characteristics of the error. To get more readable alerts, you have to create the trap definition for the error templates. The shell script `errlist.sh`, provided in Appendix A, “Problem Determination Sample Source Files” on page 169, will create the readable trap definitions in the trapd.conf configuration file for any existing error template. In Figure 111, you can see a trap card which has been sent by an client machine when a running program died with a *core dump* condition. This is typically the kind of error you should

care about on a production machine: when one of the running programs ends abnormally.

12.2 Using AIX Trouble Ticket/6000 to Manage User Problems

Problem management usually consists of:

- Identifying the problem
- Determining what the problem is
- Assigning some resource or action to it
- Monitoring the problem
- Reporting on it
- Resolving the problem

It is usually not too difficult to execute this on an individual basis, but to integrate and manage these process is a challenge to many enterprises.

AIX Trouble Ticket/6000 can help you to manage this. The main functions of AIX Trouble Ticket/6000 are:

- Create incidents (one for each user problem) from manual operator input or from AIX NetView/6000 input
- User can submit incidents using the UNIX mail system
- Support center operators could create and manage Trouble Tickets (a collection of incidents)
- Enterprise problem management policies could be implemented in AIX Trouble Ticket/6000 (security, escalation rules, notification and so on)
- Manage an inventory of your resources

Figure 112. Main Window of AIX Trouble Ticket/6000 and the Incident Report List Window

In Figure 112, you see the main window of AIX Trouble Ticket/6000 showing a summary of the open trouble tickets and unattached incidents. In the bottom window you have the list of the pending incidents that are waiting to be assigned to a Trouble Ticket. In this list, you recognize the AIX_ERROR/CORE_DUMP trap which is reported by the proxy-agent on a client and passed from NetView/6000 to Trouble Ticket/6000.

Figure 113. Main Window of AIX Trouble Ticket/6000 and the Trouble Ticket List

In Figure 113, the AIX_ERROR is assigned to a trouble ticket. You can see there are currently two other trouble tickets open. Every trouble ticket is assigned to an operator responsible for it.

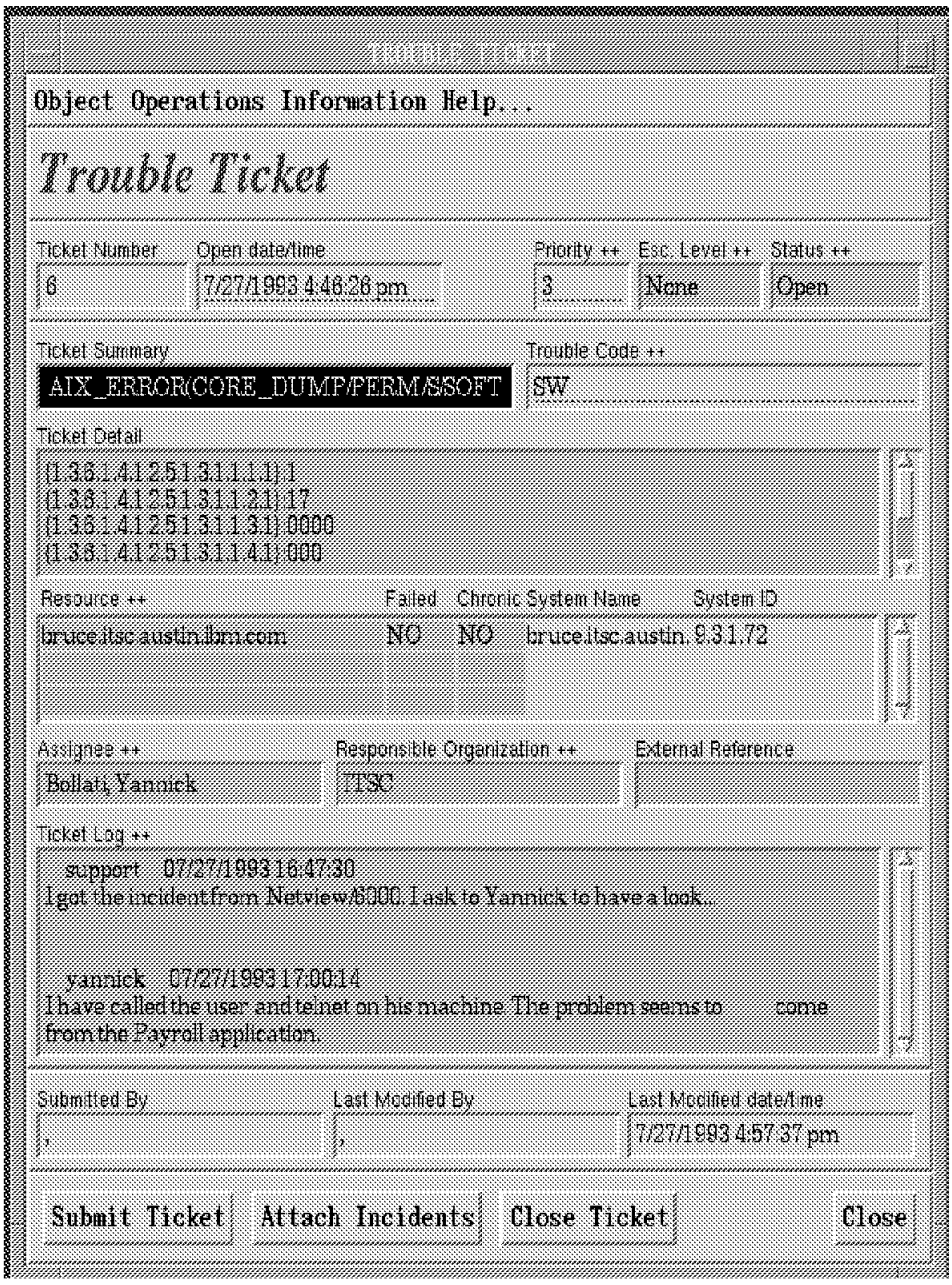

Figure 114. Details of the AIX_ERROR Trouble Ticket

In Figure 114, you see the detail window of the trouble ticket AIX_ERROR/CORE_DUMP where all the related information is stored. A trouble ticket is a live entity which will be updated during times. In the ticket log you can see, that this one had been created by user support on 7/27/1993 at 16:47 and assigned to Yannick Bollati. The same day at 17:00, the user yannick had written some update information explaining that he had called the user and that the problem seems to come from the payroll application. As shown, you can track your problems till they are resolved.

AIX Trouble Ticket/6000 is a very complete management system for support centers. There are many other powerful functions which are not described in the

previous example. You can get this information in the available Trouble Ticket/6000 documentation (see “Related Publications” on page xx).

Appendix A. Problem Determination Sample Source Files

These sample Korn Shell and C source files had been used to add the AIX error log templates as trap definitions in AIX NetView/6000 and Incident Filter Rules in AIX Trouble Ticket/6000.

A.1 Making Error Log Templates Alertable

On a typical AIX machine you will find around 700 error log templates. You can see them using the errpt -at command. You can see that around 70 of them has a stanza Alert=True. This mean that if, on this machine, you have installed the trapgend proxy-agend delivered with AIX NetView/6000 for AIX 3.2, any error with the stanza Alert=True coming up will be converted to an SNMP trap. This trap will be received by AIX NetView/6000. The Korn shell script errupdate.sh will modify all the error log templates with a stanza Alert=True.

```
#!/bin/ksh
# errupdate.sh
# AIX System Management Residency
#
# International Technical Support Center, Austin, Texas
#
# (c) IBM Copr. July 1993

errpt -t | grep -v "Error_Description" | awk '{printf("=%s:\nAlert=True
\n\n",$1)}' > errupdate.in

errupdate < errupdate.in
```

Figure 115. errupdate.sh Korn Shell Source File

Usage of errupdate.sh:

```
# errupdate.sh
0315-045 0 entries added.
         0 entries deleted.
         651 entries updated.
#
```

A.2 Adding AIX Error Log Templates as Traps in AIX NetView/6000

The AIX error log templates are installed on a machine during the installation of the corresponding LPP. For this reason, the number of those templates may be different from one machine to another. Some templates may exist on one machine and not on the other one. The errlist.sh script takes the alertable error templates and creates another script, which then creates the trap definitions using the addtrap command. If you run the created script command on your AIX NetView/6000 machine, the corresponding trap will be added or changed online in your trapd.conf file. You should do a backup of your original trapd.conf file before changing anything to it. After adding the trap definitions, any alertable errorlog entry will be sent in a readable form to AIX NetView/6000 by the trapgend proxy-agent.

© Copyright IBM Corp. 1993,1994

```
#!/bin/ksh
# errlist.sh
# AIX System Management Residency
#
# International Technical Support Center, Austin, Texas
# Yannick Bollati - IBM France
#
# (c) IBM Copr. July 1993

errpt -t | grep -v "Error_Description" | awk '{printf("%s AIX_ERROR(
%s/%s/%s/%s)\n",$1,$2,$3,$4,$5)}' | hex2dec > errlist.out

# Create the addtrap commands for Critical errors (option -S 4)
cat errlist.out | grep -v "/TEMP/" | awk '{print "addtrap
-n IBM_AIX_ERRORS -i 1.3.6.1.4.1.2.6.4.1.2.1 -l SYSMAG_TRAP -g 6
-s\"",$1,"\" -S 4 -o X -t 3 -c \"Error Events\" -f \"-\" -F \"",
$2,"\\\\n\\\$*\";echo \"",$2,"\";sleep 10"}'

# Create the addtrap commands for Minor errors (option -S 3)
cat errlist.out | grep "/TEMP/" | awk '{print "addtrap -n IBM_AIX_ERRORS
 -i 1.3.6.1.4.1.2.6.4.1.2.1 -l SYSMAG_TRAP -g 6 -s
\"",$1,"\" -S 3 -o X  -t 3 -c \"Error Events\" -f \"-\" -F \"",$2,
"\\\\n\\\$*\";echo \"",$2,"\";sleep 10"}'
```

Figure 116. errlist.sh Korn Shell Source File

```
/*
# hex2dec.c
# AIX System Management Residency
#
# International Technical Support Center, Austin, Texas
# Yannick Bollati - IBM France
#
# (c) IBM Copr. July 1993
*/

#include <stdio.h>

main()
{
    long l;
    char t[256];

    while(scanf("%x %s",&l,t)!=EOF)
              printf("%d %s\n",l,t);
}
```

Figure 117. hex2dec.s Source File

Usage of errlist.sh :

```
# cc hex2dec.c -o hex2dec
# cp /usr/OV/conf/trapd.conf /usr/OV/conf/trapd.conf.good
# errlist.sh > errlist.run_me
# ksh errlist.run_me
```

Executing the errlist.run_me file could take a long time (several hours).

A.3 Adding the AIX Error Traps to the Incident Filter Rules of TT/6000

If you like to use AIX Trouble Ticket/6000 to manage your AIX error log incidents, you will need to declare all those new traps into the Incident Filter Rules. This could be done using the AIX Trouble Ticket/6000 graphical interface or by appending those new rules to the file /usr/lpp/tt6000/site/OVfilterspecEVNX.dat. This file is used by one of the TT/6000 daemons named nvev_nxd which is charged to take the designated traps from NetView/6000 to the Incident List of TT/6000. The Korn shell file mktt6000.sh will print on the standard output one line per trap. This list must be append to the file /usr/lpp/tt6000/site/OVfilterspecEVNX.dat. You should to a backup of the original OVfilterspecEVNX.dat file and stop the TT/6000 daemons with the nx_halt command before changing anything.

The Korn shell mktt6000.sh uses a small C program to do some smart hexadecimal to integer conversion.

```
#!/bin/ksh
# mktt6000.sh
# AIX System Management Residency
#
# International Technical Support Center, Austin, Texas
# Yannick Bollati - IBM France
#
# (c) IBM Copr. July 1993

errpt -t | grep -v "Error_Description" | awk '{printf("%s AIX_ERROR(
%s/%s/%s/%s)\n",$1,$2,$3,$4,$5)}' | hex2dec > errlist.out

# Create the tt6000 filter file for critical errors
cat errlist.out | grep -v "/TEMP/" | awk '{ printf("\"1.3.6.1.4.1.
2.6.4.1.2.1\" 6 %s 00000000 (0, 0.000000)\n",$1,$2) }'


# Create the tt6000 filter file for none critical errors
cat errlist.out | grep  "/TEMP/" | awk '{ printf("\"1.3.6.1.4.1.2.
6.4.1.2.1\" 6 %s 00000000 (0, 0.000000)\n",$1,$2) }'
```

Figure 118. mktt6000.sh Korn Shell Source File

Usage of mktt6000.sh :

```
# nx_halt
# cp /usr/lpp/tt6000/site/OVfilterspecEVNX.dat \
/usr/lpp/tt6000/site/OVfilterspecEVNX.dat.good
# mktt6000.sh > mktt6000.append_me
# cat mktt6000.append_me >> /usr/lpp/tt6000/site/OVfilterspecEVNX.dat
# nx_init
```

Appendix B. CookBook on NetView for AIX and other products

This document is written from experiences in building customers system management platforms using TCP/IP networks (Ethernet, Token Ring, X.25). The platforms are based on the following products:

- NetView for AIX V3R1 (*without* database implementation)
- Systems Monitor for AIX V2R1
- PTX and PAIDE V1R2 for AIX 3.2.5
- Trouble Ticket/6000 V3R1 (*without* database implementation)

Note: This is a CookBook. It has not the ambition, nor the aim of explaining how to use these products. So, it does not develop their functionalities, but rather deals directly with difficult or complicated customization points, which have to be solved. It also tries to cover important functionality extensions (such as the use of NetView End User Interface APIs, personal applications integration in NetView's GUI, expansion of the AIX errorlog for trapgend, application supervision by Systems Monitor and trapgend, and others), illustrating them with samples as concrete and simple as possible. So, the best way to use this cookbook would be, while running the different products on a platform, to have access to User's Guides and all other necessary standard documentation.

B.1 NetView for AIX V3R1

This section develops a cookbook on the NetView for AIX V3R1 product.

B.1.1 Installation

Installation is done using `smit installp` on a AIX V3.25 system, with X11R5 and Motif 1.2.3 (not Motif 1.2.0). It needs a minimum size of 110MB of disk space in /usr (or in the filesystem /usr/OV to be created and mounted). To store online help (Dynatext, online books) will need 25MB of disk space.
It is also recommended to have a minimum of 192MB of paging space, as well as a minimum of 64MB of memory (for a maximum number of 5000 objects in the network; if your network is bigger, increase the memory and disk space size, according to the Installation and Configuration document). If you implement multi-operator management (discussed in section B.1.2.18, "Multi-Operator Management" on page 201), you will need 32MB of additional memory per operator.

You will have to verify that the following X11 fonts are already in the system:

- X11fnt.ibm850.pc.fnt
- X11fnt.coreX.fnt
- X11fnt.kanji.aixfnt (for Japanese language only)

Some PTFs on bos.obj, bosnet.snmpd.obj, X11rte.obj, X11rte.ext.obj and X11rte.motif1.2.obj are prerequisite to NetView V3 installation. Here is a list of these PTFs (notice that some may have been superseded at the moment of your installation):

- U428290 (prerequisite PTF is U428228)

© Copyright IBM Corp. 1993,1994

- U428198 (prerequisite PTF is U428228)
- U428199 (prerequisite PTF is U428228)
- U431144 (prerequisite PTFs are U424153, U428197)
- U431144 (prerequisite PTFs are U424153, U432051, U432036, U431402, U428372)
- U432350 (prerequisite PTFs are U424153, 428371)
- U428196

On 11/15/1994, there is a PTF on NetView for AIX: U434186. It notably corrects a problem about offloading node discovery to the Systems Monitor Mid-Level Manager. It needs approximately 60MB of disk space in /usr.

Installation takes approximately 90 minutes (but this naturally depends on the computer's power).

B.1.2 Configuration

After having installed NetView, you need to configure it. Here are the main customization steps you should process to get the best of its exploitation.

B.1.2.1 Common Commands

After the installation, include /usr/OV/bin in the PATH environment variable. You can now ask for the status of the daemons, as well as stop and restart them, by running respectively these commands from the AIX command line:

- `ovstatus` (daemon)
- `ovstop` (daemon)
- `ovstart` (daemon)

The `nv6000` command verifies and runs the daemons and the Graphical User Interface `ovw` application. The `Control` submenu of `smit nv6000` performs the same operations. One usually finds the command line interface more practical.

B.1.2.2 The Seed File

The seed file allows you to do two things: accelerate the very first automatic discovery of the network by the netmon daemon (after installation process and at each map and database regeneration) and limit to the subnetworks you want for all discovery processes.

To create a seed file, you will have to edit a file (pathname is indifferent, /usr/OV/seeds/seedfile, for example) with the list of hostnames or IP adresses of nodes to be discovered (gateways make the best seeds). Notice that you can use wildcard characters and IP address ranges to limit the discovery to different subnetworks.

Here is an example of a seed file:

```
9.3.1.74                      # expands initial discovery process
router1.division.company.com  # expands initial discovery process
129.35.16.100-200             # limits discovery process
129.*.18.*                    # limits discovery process
```

You have now to load the seed file using SMIT:

1. Enter smit nv6000
2. Select Set Options for netmon daemon
3. Fill in the full pathname of the seed file in the Load seed file from field

How is the seed file interpreted? Every node appearing in the seed file belongs to a subnetwork. That subnetwork and all its nodes will be discovered. If there is a gateway inside, then the other subnetworks connected to the gateway will also be discovered in an *Unmanaged* status. So, you will have to manually ask for their management to NetView. All their nodes and gateways will then be discovered, as well as new subnetworks linked to the gateways. The newly discovered subnetworks will be in the Unmanaged status.

You can also ask netmon to use a Systems Monitor Mid-Level Manager seed file, using smit nv6000→Configure→Set options for netmon daemon. The MLM seed file will contain the list of MLM nodes. In this case, the MLM will entirely manage nodes of its domain, polling them for status and configuration changes. It then reports them to NetView. We will see this function in the section B.4.3.2, "The MLM Node Discovery Table" on page 215.

B.1.2.3 The /etc/hosts file (or Domain Name Server)

Every object in the network database (node, subnetwork, ...) is identified by a label and a selection name. The label is showed in NetView topology map while the selection name is the object's unique ID in the database. To have the hostname label in the topology map, you must have configured the relation hostname IP address in the /etc/hosts file or within a DNS configuration. This must be done *before* the database is created (initial discovery process). Every subsequent hostname update (in etc/hosts or DNS) will be applied in the database (selection name), but *not* in the map (label). You then will have to manually modify the label (using NetView menu: Edit→Modify/Describe→Symbol).

B.1.2.4 The /etc/snmpd.conf file

When configuring the SNMP agent, you will have to indicate in the /etc/snmpd.conf file the name or IP address of NetView, the community name NetView will use to query the agent, the access type it will have on its MIB (readWrite, readOnly, writeOnly, none), and whether you want to limit the access to a portion of MIB. To authorize managers to interrogate the agent, you add as many lines as the number of managers, or use a network mask. Do not forget to configure the file for each SNMP agent in your network.

It is also in this file that you configure to which manager you want to route SNMP traps, with which community name, and which filter mask (fe or 1111 1110 is the filter mask that blocks no trap). Figure 119 on page 176 shows a sample of an /etc/snmpd.conf file.

```
logging         file=/logdir/snmpd.log   enabled
logging         size=0    level=0

community       public
community       auscom1 netview_node 255.255.255.255 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

# If Systems Monitor (MLM) is installed, trap destinations should be
# configured via the MLM's trap destination table - NOT IN THIS FILE.
trap            public  netview_node
#trap           public  127.0.0.1    1.2.3   fe    # loopback
```

Figure 119. The /etc/snmpd.conf File

B.1.2.5 The /usr/OV/conf/ovsnmp.conf file

Parallel to the definition of a SNMP manager in the snmpd.conf agent file, you have to list all SNMP agents the NetView manager has to manage in the ovsnmp.conf file. By default, agents are interrogated with the community name, public, but if you have given another community name in snmpd.conf, you have to report it in ovsnmp.conf.

The file is updated by running the Options→SNMP Configuration menu in NetView for AIX. Do not edit it directly, as NetView V3 now uses an ndbm database to maintain the SNMP configuration.

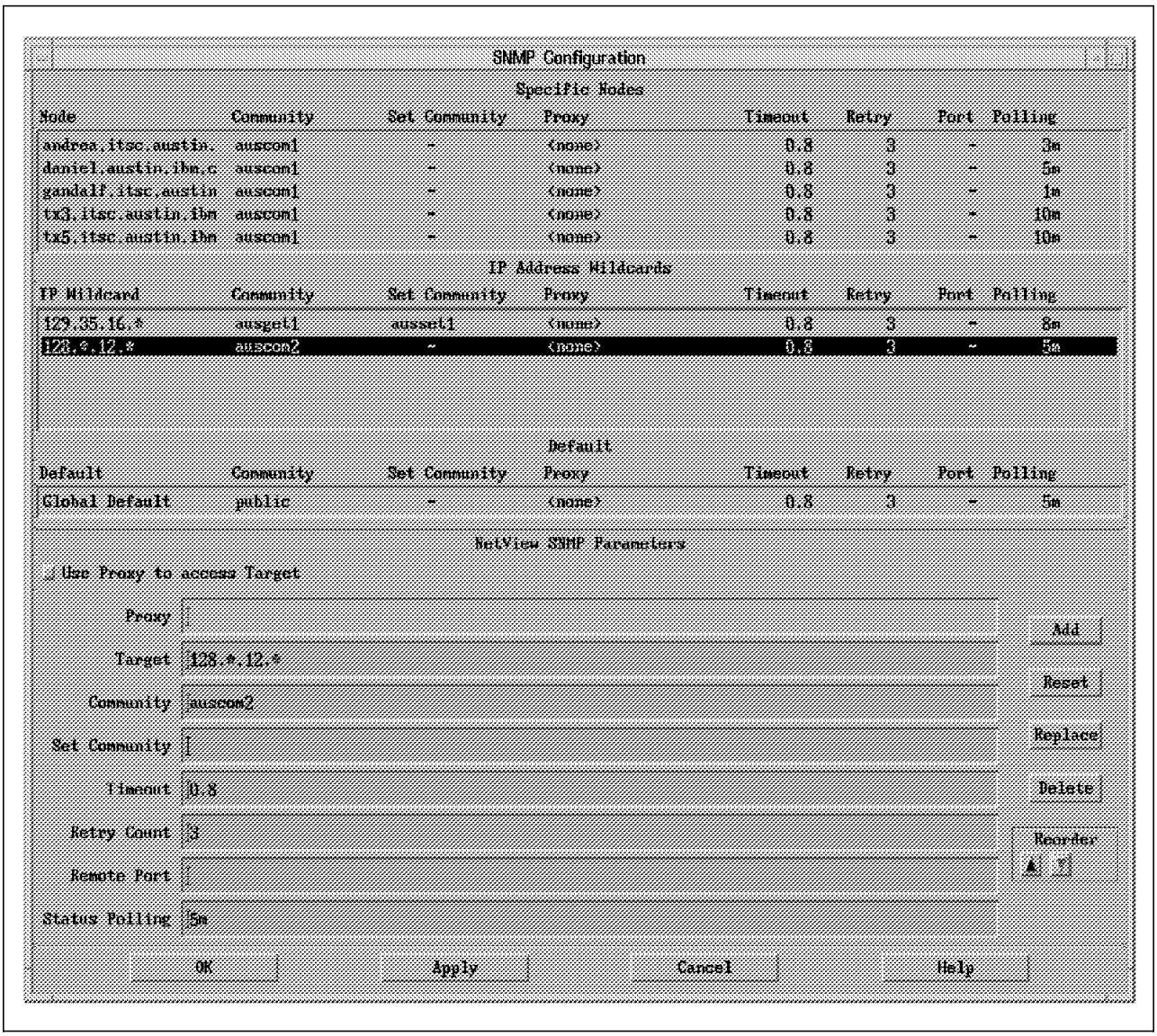

Figure 120. SNMP Configuration Panel

Figure 120 shows a sample of what can be configured.

B.1.2.6 The /usr/OV/conf/oid_to_type and /usr/OV/conf/C/oid_to_sym files

In the topology map, you should be able to distinguish at one glance the SNMP agents from the IP agents (nodes that do not run SNMP network management protocol). In fact, with the interrogation of the SNMP SysObjectId agent variable, NetView associates a symbol and a vendor (IBM, HP, ...) to a node, using, the oid_to_sym and oid_to_type configuration files, respectively.

If the SysObjectId variable is unknown in those files, then the agent will be graphically represented as an IP agent (with a generic icon, which is an empty icon), which would be particularly annoying in the graphical representation of a huge network. Consequently, you will have to modify the files in the following way (use an editor -vi- or `smit nv6000→Configure→Configure object identification registration files→Update oid_to_type/oid_to_sym registration file`):

- oid_to_type
  An example of this file is shown inFigure 121 .

```
1.3.6.1.4.1.2.3.1.2.1.1.2:IBM:IBM RS/6000:50    # SNMP Agent for AIX 3.2
1.3.6.1.4.1.23.1.1.1:Novell:Novell Lantern
1.3.6.1.4.1.36.2.15.1.5.5:DEC:DEC VMS Station:  # DEC VMS
1.3.6.1.4.1.42.2.1.1:Sun:Sun Microsystems SunOS
```

Figure 121. The oid_to_type File

- oid_to_sym
  An example of this file is shown inFigure 122 .

```
# IBM Network Enterprises
1.3.6.1.4.1.2.3.1.2.1.1.2:Computer:Workstation # IBM AIX 3.2 workstation
1.3.6.1.4.1.49.2.3.7:Connector:Multi-port      # IBM 8250 TokenRing Mgmt Mod
1.3.6.1.4.1.2.2.1.2.4:Computer:Main Frame       # IBM MVS TCP/IP Agent
```

Figure 122. The oid_to_sym File

NetView for AIX, at initial discovery process (at installation step) or at a topology map regeneration, uses these two files to associate a vendor and an SNMP agent type (oid_to_type file), as well as a class and subclass of symbols (oid_to_sym file) to the SNMP agent it newly discovered, after having requested its SysObjectId. For example, in this case, when NetView discovers a new RISC/6000 computer (where SNMP is installed standard with TCP/IP), it requests its SysObjectId. The computer answers SysObjectId=1.3.6.1.4.1.2.2.1.2.2. NetView then creates the new discovered object in its database and topology map, with the attributes Vendor=IBM, SNMP Agent=IBM RS/6000 and with the symbol of Class=Computer and of SubClass=Workstation.

NetView represents a RISC/6000 computer and an IBM Xstation in the same way (Workstation icon). If this visually bothers you, configure different representations of the two hardwares in the files. For example:

- oid_to_type

```
1.3.6.1.4.1.3.1.1:IBM:IBM X Station:       # IBM AIX X-station
```

- oid_to_sym

```
# 1.3.6.1.4.1.3.1.1:Computer:Workstation     # IBM AIX X-station
1.3.6.1.4.1.3.1.1:Server:Terminal          # IBM AIX X-station
```

From now on, RISC/6000 nodes will be represented with the Workstation symbol, and the IBM Xstations with the Terminal symbol.

On the other hand, a new version of an SNMP agent (Novell, CISCO, usually attributes a new SysObjectId. If this is the case for your platform, do not forget to update the files as in the following example about Novell:

- oid_to_type

```
1.3.6.1.4.1.23.1.1.2:Novell:Novell V2
```

- oid_to_sym

```
1.3.6.1.4.1.23.1.1.2:Server:Terminal  # Novell V2 (new version)
```

To have the modifications taken into account by NetView, you have to regenerate the topology map. This makes it *important* for you to think of these customization points during installation of the product. If other elements have, in the future, to be customized, do it manually in the map (Edit→Change Symbol Type object context menu). For the database update with the new elements, you won't have to do it. NetView processes it on a daily base, see the Poll for Configuration Changes parameter in the Options→Topology/Status Polling Interval menu.

An interesting point brought by NetView for AIX V3 is the ability to discover nodes, directly in the *Unmanaged* status, using the oid_to_type file. The following entry will, discover the OS/2 platform of your network unmanaged (instead of making you unmanage nodes manually, which can become rapidly fastidious especially for a big LAN):

```
1.3.6.1.4.1.2.2.1.2.2:IBM:IBM TCPIP OS2:hp1.U   # IBM TCP/IP Agent on OS2
```

Anyway, you can manually manage (Options→Manage objects menu) the objects at any time.

B.1.2.7 Filtering the events

NetView lets you filter the display of events/traps by the nvevents application. For that, you have to define filters (simple or compound - compound means executing AND/OR/NOT logical operations on simple filters), accessing the Tools→Filter Editor of NetView menu. The possible filtering criteria are the trap originator (nodes), the trap type (trapgend, 6611, ...) and selected traps. It is important to understand that filtering a trap will not only stop its display (the trap will be logged in /usr/OV/log/trapd.log anyway), but also not execute the automation procedure customized for the trap (Options→Event Configuration→Trap Customization menu).

You can also prevent an avalanche of traps from happening. For example, you can display only the first three traps of the same type coming in the minute. On the other hand, you can decide to display from the fourth trap coming in a two minutes time period. You activate/disactivate the filters using the Options→Filter Control menu of the nvevents application.

Lets consider this interesting example. Imagine you want to display traps from a certain set of nodes including CISCO routers. For those CISCO routers you want to operate in the following way: for the LinkDown and LinkUp traps, display only the first of the day in a period of 15 minutes, the night in 30 minutes. In fact, when a CISCO router is being installed/tested, it is usual that it sends traps every eight seconds, which will have the effect of filling the event cards tray of the nvevents application at a high speed. And when it's full, new incoming traps will erase old event cards. You will then, very quickly, loose the display of traps that you would appreciate still to have.

So, to solve that problem, three simple filters have to be created:

1. *Considered_Nodes*: Defines, in the From Objects Equal To List field, all the nodes from which you want the traps display.
2. *CISCO_Link_Day* lists, in the From Objects Equal To List field, all CISCO routers concerned. In the Events Equal to Selected fields, give the identification of traps LinkUp and LinkDown (respectively of generic numbers 2 and 3, and specific numbers 0 and 0). Fill up the Date and Time Ranges fields (from 07h00 to 20h00, for all days of the week, for example). To configure that you want to display only the first incoming trap in 15 minutes, put in the Frequency Field the value 1, in the Time Interval field the value 900 (15 minutes = 900 seconds), and click on Less Than or Equal To.

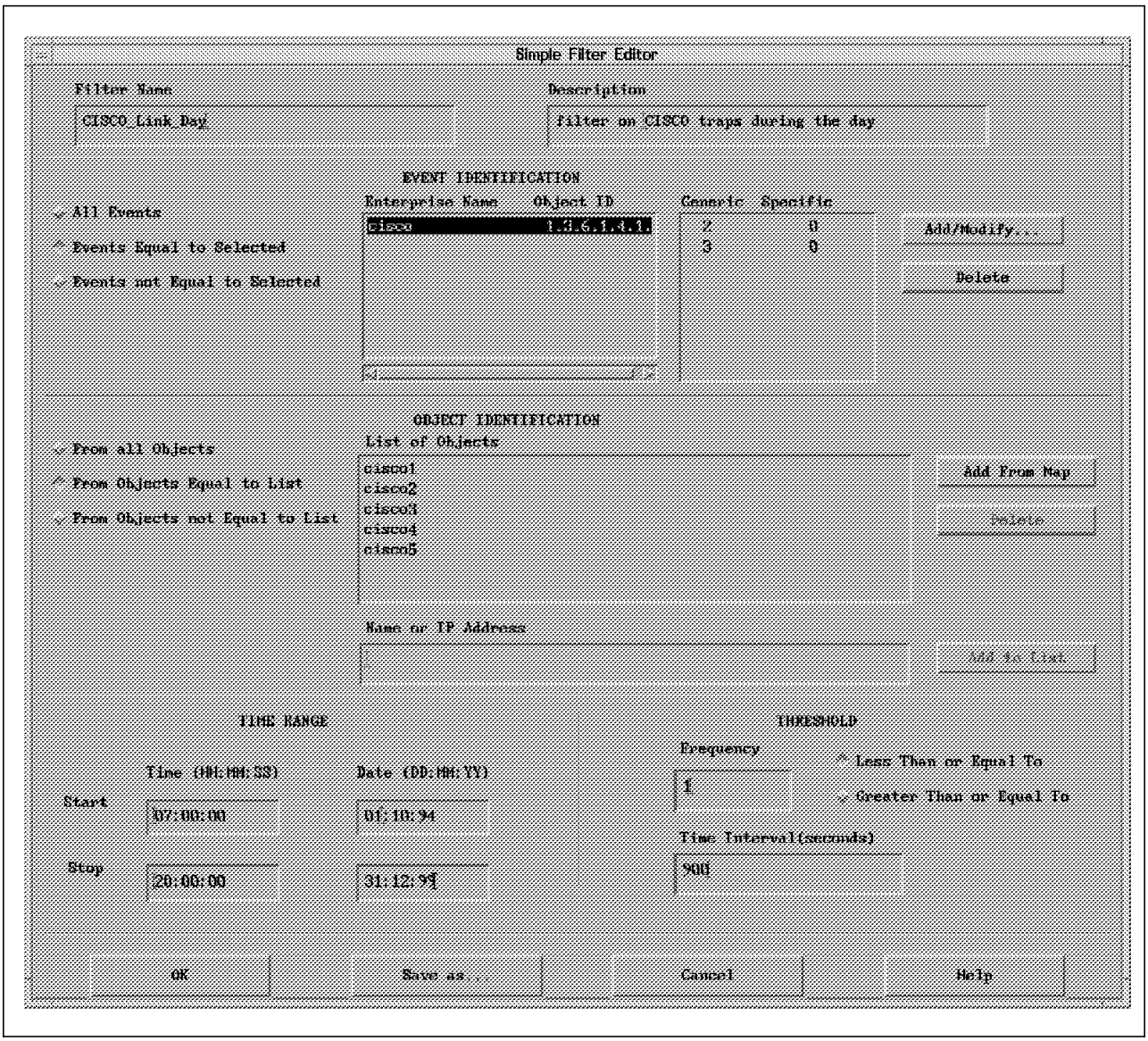

Figure 123. Filter Editor Panel

3. *CISCO_Link_Night* lists, in the From Objects Equal To List field, all CISCO routers concerned. In the Events Equal to Selected fields, give the identification of traps LinkUp and LinkDown (respectively of generic numbers 2 and 3, and specific numbers 0 and 0). Fill up the Date and Time Ranges fields (from 20h00 to 07h00, for all nights of the week, for example). To configure that you want to display only the first incoming trap in 15 minutes, put in the Frequency Field the value 1, in the Time Interval field the value 1800 (30 minutes = 1800 seconds), and click on Less Than or Equal To.

Now, you create a compound filter, named General_Filter, using the operation of logical OR:

```
General_Filter = CISCO_Link_Day OR CISCO_Link_Night OR Considered_Nodes
```

Note: What is the difference between a trap and an event?
In a simple way, an event is a displayed trap (after the process of filtering and X11 formatting).

B.1.2.8 Logical Views

For a big network, the Xstation or hft screen will be too small to show with precision the network topology. To solve this problem, you will have to create logical views, partitioning the network. To do this, add an icon of type container (Network or Location), explode that icon in a submap, and place the nodes, subnetworks, and other objects with cut/paste operations. Use cut rather than copy because the ipmap application, which manages the graphical topology map, does not support it.

On the other hand, to still have your objects managed by netmon, after a cut/paste operation, you have to follow some (logical) rules:

- No symbols can be moved from the Root submap
- IP network symbols and gateway symbols can be moved from the Internet submap
- No symbols can be moved from the Network submap (to have consistency with IP addresses)
- Computer symbols and connector symbols can be moved from the Segment submap
- No symbols can be moved from the Node submap

B.1.2.9 Actions Automation on Trap Reception

In the `Options→Event Configuration→Trap Customization` menu, you can select the product (enterpriseId) that you want to automate actions on reception of particular events (NetView, trapgend, Systems Monitor, 6611, ...). You also can create/modify an event giving it a particular title, the title that appears on the top of the event card and which is configured in the field Event Log Message. Some variables may be integrated in the title, for example: $A for the agent′s name (hostname, IP address), $G for the generic trap number, $S for the specific trap number, $C for the used community name.

You can give it a new event category, status and severity. For example, the `Status Event` category has the `User2` status (violet as the standard color) and the `Critical` severity which means at the trap reception an event card will appear, positioned at Critical and the agent′s color on the topology map becomes violet.

It is important to notice that predefined NetView events are not authorized to change the agent status (colors). Any other enterpriseId can do it (trapgend, Systems Monitor, cisco, ...) on all *Enterprise Specific* traps (generic number 6). To bypass this restriction, it just takes, at the trap reception, to automatically resend the specific trap `58916871` with a news status using the `snmptrap` command (you force the status change). This point will be developed in the section B.1.2.12, “Change of Status on NetView Traps Reception” on page 188.

In the `Command for Automatic Action` field, any AIX command can be passed. AIX commands such as beeps: `/usr/OV/bin/ovxecho` or `ovxbeep` (this includes standard to different types of beep you can create, for example, from red and noisy, to yellow and less noisy) or any executable program such as a shell script or a C program either simple or complex.

These are some examples of automated actions on the reception of the following NetView events:

- **Node Up**: Run a beep (ovxecho)

- **Node Down**: Run a beep (ovxecho)
- **Interface Up**: Run a shell script which sends an ICMP echo (ping) on the agent, look for connection informations (netstat), and search in the AIX error log already registrated anomalies on the interface (token-ring or Ethernet). Results are sent via mail (SMTP) to an operator to be specified (you can choose root@netview_node by default). This gives the following shell script:

```
administrator=$2
(echo "`date`\n\t\tALERT !!! INTERFACE UP ON THE NODE $node\n")
>/tmp/interf_up$$.res
(echo "\n\n`date`\n\t\tPING ON THE NODE $node\n";ping -c 3 $node)
>>/tmp/interf_up$$.res
(echo "\n\n`date`\n\tERRORLOG OF NODE $node: INTERFACE ANOMALIES ON
`date -u +%D`\n") >>/tmp/interf_up$$.res
rexec $node errpt | grep tok | grep `date -u +%m%d` | grep `date -u +%y`
>>/tmp/interf_up$$.res
ovxecho Interface Up on node $node
sleep 3
mail $administrator </tmp/interf_up$$.res
rm -f /tmp/interf_up$$.res
```

B.1.2.10 Reception of Non-Preconfigured Traps

It may happen, in the case of CISCO routers for example, that a trap arrives on NetView without having been previously configured in the /usr/OV/conf/trapd.conf file. So what you will do first is to ask to the SNMP supplier (CISCO) for the signification of the incoming trap identified by an enterpriseId, a generic and a specific trap number. This signification will be written down in the Event Log Message field of the Options→Event Configuration→Trap Customization menu. As the traps bring their own variables, it should be interesting to display them in the event card generated by NetView. For that, first display all the variables existing in the trap you receive, writing the following sentence in Event Log Message:

```
CISCO trap: $# variables are: \n$*
```

You can now rewrite your Event Log Message, for a better comprehension, in the following way:

```
CISCO trap: $# variables are: \nVariable_1: $1 \nVariable_2: $2
\nVariable_i: $i
```

B.1.2.11 Data Collection and Thresholds

To achieve *performance management*, NetView for AIX provides a data collection tool you can access from the Tools→Data Collection and Thresholds menu.

The data collection can run on any MIB variable (standard or private), if that variable is naturally a numeral (type Integer, Counter, Gauge). It is important to notice that concerning the variables of Counter type, it's the variation of their value (the *delta*) that is collected and not their real value (which is, by the way, very interesting). After having specified the instance(s) of the variable to be collected in a menu of MIB Browser type, you choose the collection mode: either a memorization of collected data (Store mode) or a detection of exceeded thresholds (Check Thresholds mode), or both.

You specify the polling interval (every five minutes for example), the concerned nodes, and for exceeded thresholds detection, the threshold value itself and the trap to be sent to NetView. Obviously, at the moment that you specify a new trap, you shouldn't forget to configure it (event log message, category, severity, automated procedure, ...) in the Options→Event Configuration menu.

It is important to notice that when you configure a trap creation at a threshold detection, you also can position a rearm condition (value of the collected MIB variable, under which the measures have to pass for the threshold detection process be reactivated). The rearm trap does not exist in the Data Collection menu. In fact, all threshold traps are odd numbers and the rearm trap is the even increment of the threshold trap. Do not forget to create it also in the Options→Event Configuration menu.

Let's consider the following example. We would like to manage more precisely the IBM 6611 router,
which is a RISC/6000 hardware, with its MIB variable, ComputerSystemLoad (that gives the percentage of CPU resource used at a t time). Notice that this variable belongs to the private MIB of trapgend, SNMP proxy-agent implemented on all RISC/6000 computers, and managing especially the AIX errorlog.

To do this, in the Tools→Data Collection and Thresholds menu, specify the variable ComputerSystemLoad after having gone through trapgend's private MIB in the MIB Browser submenu that is called from your request. You will find it in the netView6000SubAgent subtree. As this variable has only one instance, you will give the 0 instance number. You specify the node on which you want to collect. If you want a threshold detection, choose the Check Thresholds mode and indicate the threshold critical value, as well as the polling interval and the trap to send to NetView, consequently to the detection of a threshold exceeded. You don't need to identify the rearm trap, as it will be the even increment of the threshold trap. For example, if you specify 1001 as threshold trap, the rearm trap will automatically have the trap number 1002.

By default, NetView traps 58720263 and 58720264 are the respective traps of threshold detection and rearm.

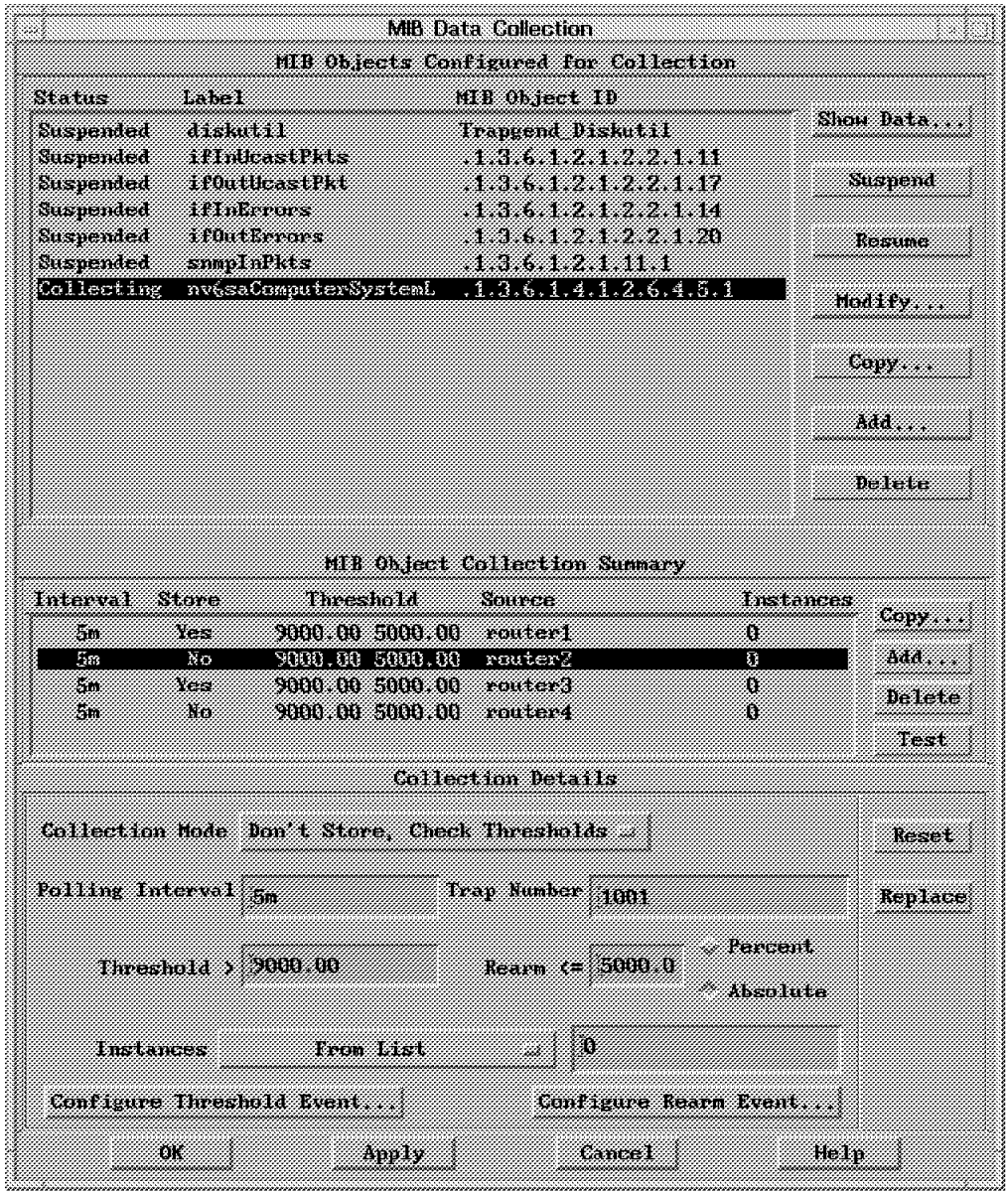

Figure 124. MIB Data Collection Panel

If you want to define other personal traps, you have to choose a trap number in the interval [1001,1999] (see Figure 124).

For example, lets give the number 1001 for the threshold trap in the Data Collection menu. You will then have to define it along with the rearm trap 1002 (see Figure 125 and Figure 126 on page 187) in the MIB Data Collection panel by clicking on the Configure Threshold Event and Configure Rearm Event buttons, with the following characteristics:

```
EnterpriseId = netview6000
Generic Number = 6
Specific Number = 1001
Event Log Message = $3
Category = Threshold Events
Severity = Critical (for example)
Command for Automatic Action = /usr/OV/procs/thresh_exceed $A $3
```

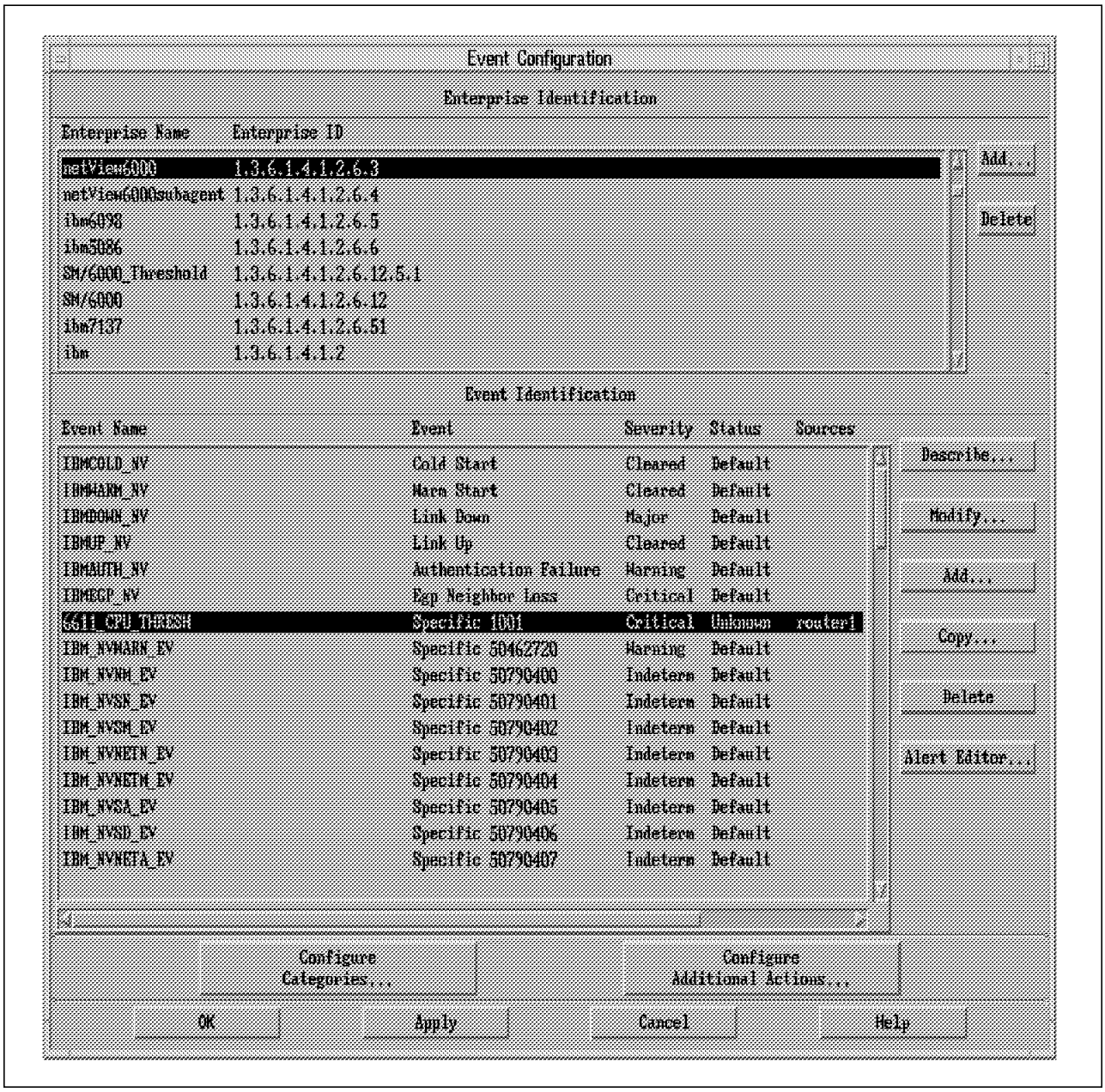

Figure 125. Event Configuration Panel

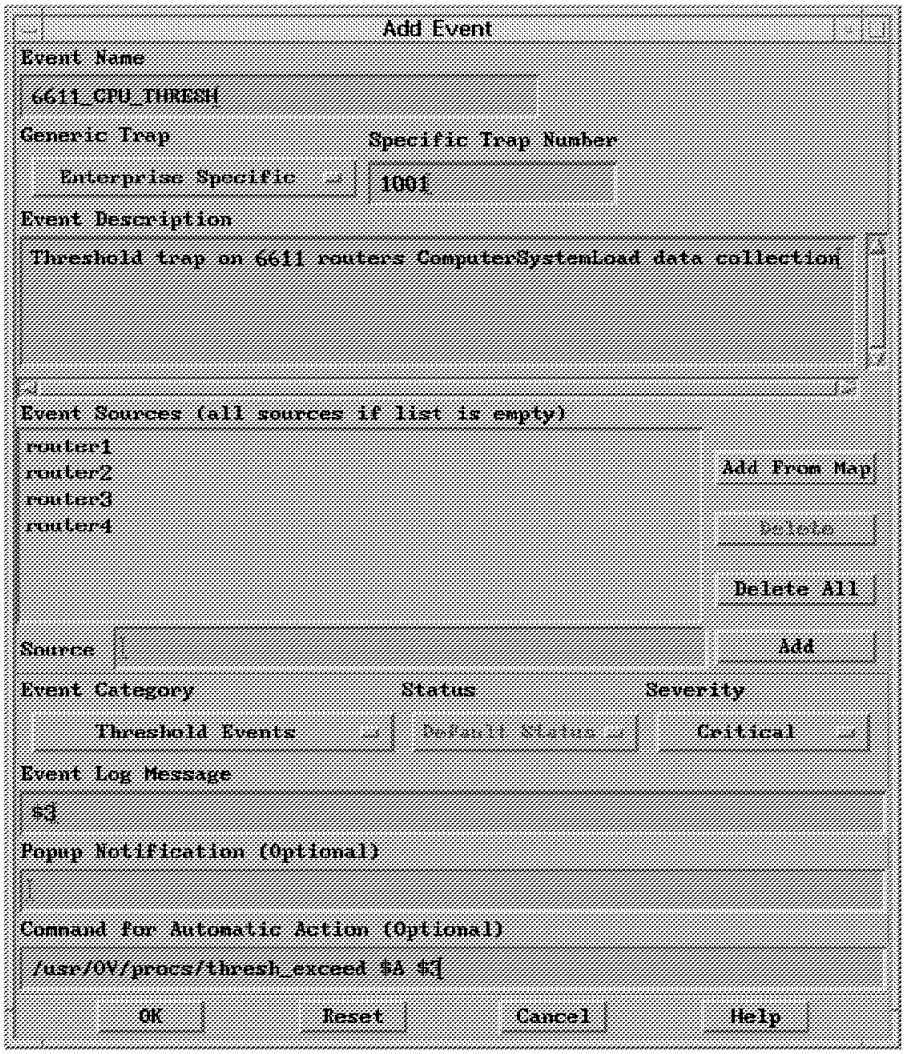

Figure 126. Event details

The $A variable identifies the SNMP agent sending the trap. The $3 variable contains the data brought by the trap. These variables will be exploited by the shell script thresh_exceed. By the way, here is its code:

```
value=$2
value=`echo $value | cut -f2 -d":"`

/usr/OV/bin/ovxecho "$node: Threshold ComputerSystemLoad exceeded = $value"
```

Define the rearm trap of number 1002 in the same way with Severity Warning (for example) and of Command for Automatic Action, /usr/OV/procs/thresh_rearm $A $3, which code is:

```
value=$2
value=`echo $value | cut -f2 -d":"`
value=`echo $value | cut -f1 -d"."`

/usr/OV/bin/ovxecho "$node: ComputerSystemLoad returned normal = $value"
```

B.1.2.12 Change of Status on NetView Traps Reception

No trap with the enterpriseId NetView6000 is authorized to change the status of an object (Status: Normal, Marginal, Critical, User1, ...). For example, a threshold trap in a data collection cannot modify the status (and consequently the color) of an object. In some cases, you may want to directly visualize the incidence of a threshold detection and of a rearm on the concerned agent. But NetView doesn't allow that in the trap customization (see Status button of the Options→Event Configuration→Trap Customization menu where that field is inaccessible). By the way, this gives significant security in the status management of the objects by NetView (more precisely by netmon, the daemon of supervision of the network topology).

If you want the status (and so the color) to be modified the same, you will have to configure your trap so that it sends the following trap, the only trap of NetView6000 enterpriseId authorized to change the object status:

```
enterpriseId = netview6000
generic trap = 6
specific trap = 58916871
```

For example, write the following program in the Command for Automatic Action field of the Options→Event Configuration→Trap Customization menu: /usr/OV/procs/trapNewStatus $A nv6000 User1

```
#Example of utilization:
###    trapNewStatus node1 "nv6000" User2
#New Status: User1: color Pink; User2: color Violet; Up; Down; etc...
#
node=$1
netview=$2
NewStatus=$3
#
/usr/OV/bin/snmptrap $netview .1.3.6.1.4.1.2.6.3.1 \
 $node 6 58916871 1 \
 .1.3.6.1.4.1.2.6.3.1.2.0 Integer 14 \
 .1.3.6.1.4.1.2.6.3.1.3.0 OctetString $node \
 .1.3.6.1.4.1.2.6.3.1.4.0 OctetString "Object status is" \
 .1.3.6.1.4.1.2.6.3.1.5.0 OctetString $NewStatus
```

You should make sure that the concerned object symbol type is Symbol or Object, and not Compound Status in the Edit→Modify/Describe→Symbol menu of the object. A Compound Status type obeying the status propagation rules, you will not see the impact of your command on the symbol.

Note: Any other specific trap of enterpriseId different from netview6000 can do that change of status (see the Status button no more greyed in the Options→Event Configuration→Trap Customization menu).

B.1.2.13 Programming the Graphical Interface: Use of EUI APIs

A very interesting aspect of the actions automation on trap reception that you can implement is the manipulation of topology map components (symbols, submaps, maps, ...) using the EUI APIs (End User Interface APIs). Manipulating those objects is managed by very complex programs (IPC programmation in C, APIs, ...), and for this reason, the NetView for AIX ITSO in Raleigh (Dave Shogren's team) wrote completely satisfactory samples covering almost all development needs with the NetView EUI APIs. A redbook has been written to explain and document the samples. Read it if you want to go thoroughly in the EUI APIs. Its reference is: *GG24-4059 Examples of Using AIX NetView/6000 APIs*.

The README file included with the ITSO-Raleigh wteuiap6 package is very clear for its installation and its use. After having restaured it (using the tar command) in a personal directory, it just then takes running the make install command which will compile all the components and install them in different directories of NetView. The make uninstall command runs the inverse action. You should then stop NetView (GUI and daemons) and rerun it to get the platform ready to respond to EUI requests. Those requests are managed by the wtdriver6 program, which is to be executed in the following way:

```
Usage: wtdriver6 [flags] command [...]
Flags:
       [-h wtpbxd-hostname]  - Specify the machine that is running wtpbxd
                               (default is local)
       [-b]                  - Send the request to all displays (broadcast)
       [-d target-display]   - Send the request to a specific display
       [-f submap-name]      - Define the focus (just for this command)
       [-m map-name]         - Send the request to a specific map

Commands:
       stat
       msg      message
       focus    submap_name
       popup    submap_name
       submap   submap_name              [layout] [background-image]
       submapof submap_name object_name [layout] [background-image]
       copy     symbol_name submap_name submap_name
       move     symbol_name submap_name submap_name
       sort     submap_name [keys]
       getlabel object_name
       add      symbol_name symbol_type
                    [x y]
                    [label symbol_label]
                    [x y]
                    [label symbol_label]
                    [submap submap_name]
                    [exec appl_name action_name]
       del      symbol_name
       connect  symbol_name symbol_name
       set      symbol_name status
       assoc    object_name field_name [field_value]
       delobj   object_name
       cloneseg network_name segment_number
```

```
########## continued ##########
       Note:
              keys are:
                        l  - symbol label
                        t  - symbol type
                        s  - symbol status
                        o  - object status
                        c  - compound status

              some symbol types are:  mf     for  "Computer:Main Frame"
                                      ws     for  "Computer:Workstation"
                                      cr     for  "Connector:Multi-port"
                                      ap     for  "Software:License"
                                      ss     for  "Software:Process"

              [ If you want to check for the abbreviation table, ]
              [ edit wteuiap6.c and search for symabbrev         ]

                                       or anything valid in /usr/OV/symbols/C
                                       such as from:        /usr/OV/symbols/C/Cards

                                      "Cards:Audio"
                                      "Cards:Video"
                                      "Cards:Thin LAN"

                                       another example, from: /usr/OV/symbols/C/Server
                                      "Server:File System"

                    set status may be:  unknown
                                        normal
                                        marginal
                                        critical
                                        acknowledge
                                        up
                                        down
```

Lets consider a concrete case of using the program. We want to process an IDNS backup for a router (CISCO for example), so that when an interface (a link) goes down, there is an automatic topple over another interface accessing an IDNS network.

Usually, the two interfaces are configured with the *same IP address*. So, the problem on NetView for AIX will be that you see an interface (as there is only one IP address), that does not reflect at all what's going on. In fact, when the link goes out of order, the card symbol becomes red (Status=Critical); then at backup, it goes green (as IP address responds to ping again). This won't be realistic since that symbol represents the primary communication card which is still unable to communicate.

Always at backup, a Interface up trap on the node arrives on NetView. When all returns normal, you again receive that trap, which then looses all signification (returning to normal situation or passing to backup mode).

To solve that problem, you create an object of IP Cards type in the router node submap, which you will call BackUp for example. You will then configure the two traps: Interface Up and Interface Down, so that they execute the following shell scripts on reception by NetView for AIX:

- Automatic action on the Interface Down trap

```
#####################################################################
#Program run by the trap "Interface XXX down" (when NetView
#can no more ping the IP address represented by XXX)
#
#Interest of the program: graphically show on NetView topology map the
#IDNS BackUp on a CISCO router
#
#Naming convention: all CISCO routers have their hostname beginning
#with "cisco"
#####################################################################
#
node=$1
value=$2
community=$3
interface=`echo $value|cut -f2 -d" "`
node_cisco=`echo $node | cut -f1 -d"o"`
value2=BackUp
backup=$node$value2
nameInterf=$node:interface

(echo "`date '+%D %T'`\t$node \t\t$value") >> /logdir/IDNS.log
sleep 120

if [ "$node_cisco" = "cisc" ]
then

 if [ "$interface" = "Serial0" ]
 then

  res1=`snmpinfo -m get -h $node -c $community IFOPERSTATUS.1`  #Status of Serial0
  RES1=`echo $res1 | cut -f2 -d"="`
  RES1=`expr $RES1`

  res2=`snmpinfo -m get -h $node -c $community IFOPERSTATUS.2`  #Status of Serial1
  RES2=`echo $res2 | cut -f2 -d"="`
  RES2=`expr $RES2`

  if [ "$RES2" = 1 ] && [ "$RES1" = 2 ] #Serial0 down and Serial1 up
  then

  /usr/OV/raleigh/wtdriver6 set $backup up
   (echo "`date '+%D %T'`\t$node \t\tBackup on IDNS (Serial1)
established") >> /logdir/IDNS.log
   ovxecho "$node: IDNS link well established"
  fi

  if [ "$RES2" = 2 ] && [ "$RES1" = 2 ] #Serial0 down and Serial1 down
  then

  /usr/OV/raleigh/wtdriver6 set $backup down
   (echo "`date '+%D %T'`\t$node \t\tBackup on IDNS (Serial1) non
successfull") >> /logdir/IDNS.log
   ovxecho "$node: IDNS Link non established"
  fi

  if [ "$RES2" = 2 ] && [ "$RES1" = 1 ] #Serial0 up and Serial1 down
  then

  /usr/OV/raleigh/wtdriver6 set $backup User1
  /usr/OV/raleigh/wtdriver6 set $nameInterf up
   (echo "`date '+%D %T'`\t$node \t\tBackup IDNS over, Return to
normal") >> /logdir/IDNS.log
   ovxecho "$node: Backup IDNS over, Return to normal"
  fi

 fi
fi
```

- Automatic action on the Interface Up trap

```
#####################################################################
#Program run by the trap "Interface XXX up" (as NetView can ping again
#the IP address represented by XXX)
#
#Interest of the program: graphically display on NetView the
#IDNS BackUp on a CISCO router
#
#Naming convention: all CISCO routers have their hostname beginning
#with "cisco"
#####################################################################
#
node=$1
value=$2
community=$3
interface=`echo $value | cut -f2 -d" "`
node_cisco=`echo $node | cut -f1 -d"o"`
value2=BackUp
backup=$node$value2
nameInterf=$node:interface

(echo "`date '+%D %T'` \t$node \t\t$value") >> /logdir/IDNS.log
sleep 120

if [ "$node_cisco" = "cisc" ]
then

 if [ "$interface" = "Serial0" ]
 then

  res1=`snmpinfo -m get -h $node -c $community IFOPERSTATUS.1` #Status of Serial0
  RES1=`echo $res1 | cut -f2 -d"="`
  RES1=`expr $RES1`

  res2=`snmpinfo -m get -h $node -c $community IFOPERSTATUS.2` #Status of Serial1
  RES2=`echo $res2 | cut -f2 -d"="`
  RES2=`expr $RES2`

  if [ "$RES2" = 2 ] && [ "$RES1" = 1 ] #Serial0 up and Serial1 down
  then

  /usr/OV/raleigh/wtdriver6 set $backup User1
   (echo "`date '+%D %T'` \t$node \t\tBackup on IDNS over, Return to
normal") >> /logdir/IDNS.log
   ovxecho "$node: IDNS link liberated, Return to normal"
  fi

  if [ "$RES2" = 1 ]  && [ "$RES1" = 2 ] #Serial0 down and Serial1 up
  then

  /usr/OV/raleigh/wtdriver6 set $backup up
  /usr/OV/raleigh/wtdriver6 set $nameInterf down
   ovxecho "$node: Backup on IDNS continues, do not consider the trap
Interface Serial0 up"
  fi

  if [ "$RES2" = 1 ] && [ "$RES1" = 1 ] #Serial0 up and Serial1 up
  then

  /usr/OV/raleigh/wtdriver6 set $backup User1
   (echo "`date '+%D %T'` \t$node \t\tBackup over, but IDNS link
not liberated") >> /logdir/IDNS.log
   ovxecho "$node:  Backup over, but IDNS link not liberated"
  fi

 fi
fi
```

The BackUp object we created, representing the IDNS BackUp interface card of the router, will be coloured pink (Status = User1) in the normal mode. It becomes green if the BackUp is well established in the degrade mode, and red if not. The symbol representing the primary interface in NetView will just indicate the reality.

B.1.2.14 Application Interrogating MIBs

The Tools→MIB Application Builder menu lets you *dynamically* integrate personal applications in NetView's menus, without having to create or modify a file in /usr/OV/registration/C, nor exiting the ovw graphical application and rerunning it. There is the same a limit to that function, which is the application has to interrogate MIB variables (standard or private). The variables are queried in a table or graphical form for all their instances. If you want to specify only some particular instances, you have to manually modify the file generated by the MIB application Builder tool in the /usr/OV/registration/C/ovmib directory. The graphical form accepts the query of variables belonging to different MIB subtrees, even to different MIBs. This is not possible for the table form.

Figure 127. MIB Application Builder Panel

You also can add a Help menu to your new application (see Figure 127).

After having entered all the definitions for the application, exit the Tools→MIB Application Builder menu. Now, if you activate the Monitor menu, you will see the name of the application you just created.

B.1.2.15 New Private MIBs Integration

NetView for AIX brings in the /usr/OV/snmp-mibs directory most of the private MIBs on the market, which describe hardware and software SNMP agents (IBM 6611, CISCO, synoptics, wellfleet, Novell, etc...). You may receive new MIBs or updates of existing MIBs.

The Options→Load/Unload MIBs menu lets you load or unload the MIB file (location directory is to be precised). Concretely, this operation will *visually*

integrate the declarations of the new MIB variables in the differents menus of NetView (MIB Browser, Data Collection and Thresholds, MIB Application Builder, ...).

Before processing the load/unload MIB operation, you *can absolutely query* the SNMP agent on the new MIB, using the `snmpget` or `snmpinfo -m get` command from the AIX command line. The point is that you have to give the complete SNMP pathname of the variable in that case (`.1.3.6.1.4.1.etc...`).
The act of integrating new MIBs to NetView makes those operations much easier to use. But keep in mind that it is the only thing it does.

B.1.2.16 Current Administration of the Product

During the use of NetView, you have to periodically ensure that daemon log and trace files and data collection files do not take too much disk space. You also should think of maintaining topology databases.

Log and trace files are located in the /usr/OV/log directory. You should particularly think of the following files:

- ***trapd.log***: In this file are logged systematically all the traps coming on NetView. There won't be any filtering at this level, that's why the logfile size can increase very quickly. The trapd daemon automatically clears trapd.log after having saved it in a trapd.log.old file when it reaches a specified size, 4096KB by default (see `smit nv6000→Configure→Set options for trapd daemon`).
  You can configure your own maintenance action in this SMIT menu. A possible choice would be the /usr/OV/bin/trapd.log_Maint script, which archives trapd.log datas in a relational database.
- ****.trace***: These are all of the *.trace files (netmon.trace, trapd.trace, snmpCol.trace, ...). Check these especially when you have run the corresponding daemons in tracing mode.
  NetView for AIX provides shell scripts in the /usr/OV/cron directory for routine maintenance: netmon.trace_Maint, snmpCol.trace_Maint, trapd.trace_Maint. The programs keep the last two versions of the trace files and clear them. Use the crontab to registrate them.

Maintaining topology databases is possible for:

- Resolving database inconsistencies (deleting unneeded objects from the IP topology database): Use `smit nv6000→Maintain→Resolve inconsistencies between ovtopmd and ovwdb databases`
- Compressing the IP topology database: Use `smit nv6000→Maintain→Compress the IP topology database`
- Removing old snapshots: Use `smit nv6000→Maintain→Manage map snapshot→Remove map snapshots`

You should also verify data collection files (especially when data is regularly collected) in the /usr/OV/databases/snmpCollect directory. Think of using reasonable polling intervals, especially when you want to check thresholds and *store* the results.

The smit nv6000→Maintain→Manage crontab entries menu enables you to automate, on a regular period to specify, the administration operations. This is an example of a script routine maintaining data collection files:

```
#########################################################################
#Example of execution: CopieCollect diskutil.1
#
#This program is run by a crontab every beginning of the month:
#it dumps in ascii format the data collection file given as a
#parameter, renames the resulting file as in the format
# "collected_file.month_year", and erases the data collection file.
#
#########################################################################

file1=$1
fichCol=/usr/OV/databases/snmpCollect/$file1
month=`date +%m`
year=`date +%y`
month=`expr $month - 1`

if [ "$month" = 0 ]
then
 month=12
 year=`expr $year - 1`
fi

file2=$file1.$month"_"$year
fichLog=/logdir/$file2
snmpColDump $fichCol > $fichLog
cat /dev/null > $fichCol
```

The crontab entry for that program will be:

```
1 0 1 * * /usr/OV/procs/CopieCollect diskutil.1
```

The program will then be executed every first day of the month at 00h01. Obviously, you will create a crontab entry for each interesting data collection file.

B.1.2.17 Integration of Personal Applications

NetView for AIX gives the possibility for each user to integrate his own applications in a submenu of the GUI. To do that, the user has to declare in a file (new or already existing) in the /usr/OV/registration/C directory (and *only* that directory) the way he wants his new application to behave: calling menu (menu bar, tool palette, context menu, ...), various attributes and program executed.

To explain this possibility, let's take the following registration file: /usr/OV/registration/C/XYZ.reg as an example:

```
/*
 Registration for Customer XYZ Corp. applications.
*/
Application "Applications XYZ"
{
    Description { "Applications for Customer_XYZ" }

    Copyright { "(c)Copyright 10/1994, Andrea LI-SAI, IBM France" }
    Version "1.0";

    MenuBar "XYZ" _X
    {
        <100> "Applications..." _A f.menu "Applications";
    }
```

in the menu bar XYZ appears. By clicking on it with the mouse, or by typing Alt-X, you will see Applications..., whose behaviour is developed in the Menu "Applications" section.

```
    Tool "syscheck"
        {
        Icon Bitmap "/usr/OV/bitmaps/C/XYZ.xbm";
        LabelColor "red";
        SelectionMechanism drag-drop, double-click;
        Action "actions1";
        }
```

In the tool palette, you can see a new icon, with Tool_1 XYZCorp written inside (bitmap XYZ.xbm), red entitled syscheck. By double-clicking or making a drag-drop with the mouse, the action described in the Action "actions1" section is executed.

```
    Menu "Applications"
    {
        <100> "Printer Status..."               _P f.action "actions2";
        <100> "Connected ROOT Users..."         _C f.action "actions3";
        <100> "Data Collection Format..."       _D f.action "actions4";
    }
```

Clicking on the Applications... submenu as described earlier, the choices: Printer Status..., Connected ROOT Users..., Data Collection Format... appear in NetView's GUI. The choices execute the actions developed in the corresponding Action "actions*i*" sections.

```
    Action "actions1"
    {
        Command '/usr/bin/X11/aixterm -sb -e /usr/OV/service/syscheck';
    }

    Action "actions2"
    {
        MinSelected 1;
        MaxSelected 1;
        SelectionRule isNode;
        NameField "IP Hostname";
        Command '/usr/bin/X11/aixterm -title "${OVwSelection1}" -sb -e
/usr/OV/procs/prog_lpstat "${OVwSelection1}"';
    }
```

To activate that submenu, you will first have to click on minimum one node and maximum one node. On the other hand, the node must have the isNode capability (see the Edit→Modify/Describe→Object menu).

You can pass a parameter in the Command field. Here, the parameter is the selected node itself. It's called ${OVwSelection1} (${OVwSelection2}, if 2 nodes have been selected, ...), and its type is IP Hostname (it can also be IP address). The prog_lpstat shell script will first verify wether the concerned node is UNIX or not, and then run the lpstat command.

```
    Action "actions3"
    {
        MinSelected 1;
        MaxSelected 1;
        SelectionRule isNode;
        NameField "IP Hostname";
        Command '/usr/bin/X11/aixterm -title "${OVwSelection1}" -sb -e
/usr/OV/procs/prog_users "${OVwSelection1}"';
    }
```

The prog_users shell script will first verify wether the concerned node is UNIX or not, and then run the who | grep root command.

```
    Action "actions4"
    {
        Command '/usr/bin/X11/aixterm -sb -e
/usr/OV/procs/FormatCollect';
    }
}
```

Detailed explanations are given in *NetView for AIX Programmer's Guide* documentation, especially on these kinds of registration files syntax.

One of the programs run by the above Action paragraph is /usr/OV/service/syscheck. It checks the NetView for AIX entire platform: total amount of paging space; free Disk Space in /, /usr, and /tmp; X Windows version; AIX Version and Release level; Hostname, IP address; products Version and PTF's installed (NetView for AIX, Systems Monitor, Trouble Ticket/6000, LMU/6000, SNA Manager/6000, ...).

Here is what the other program does:

```
#Program /usr/OV/procs/FormatCollect
#
echo "Enter the name of the data collection file: "
read file
/usr/OV/procs/CopieCollect $file

#CopieCollect is a program which code has been explained in the
section B.1.2.16, "Current Administration of the Product" on page 195.
```

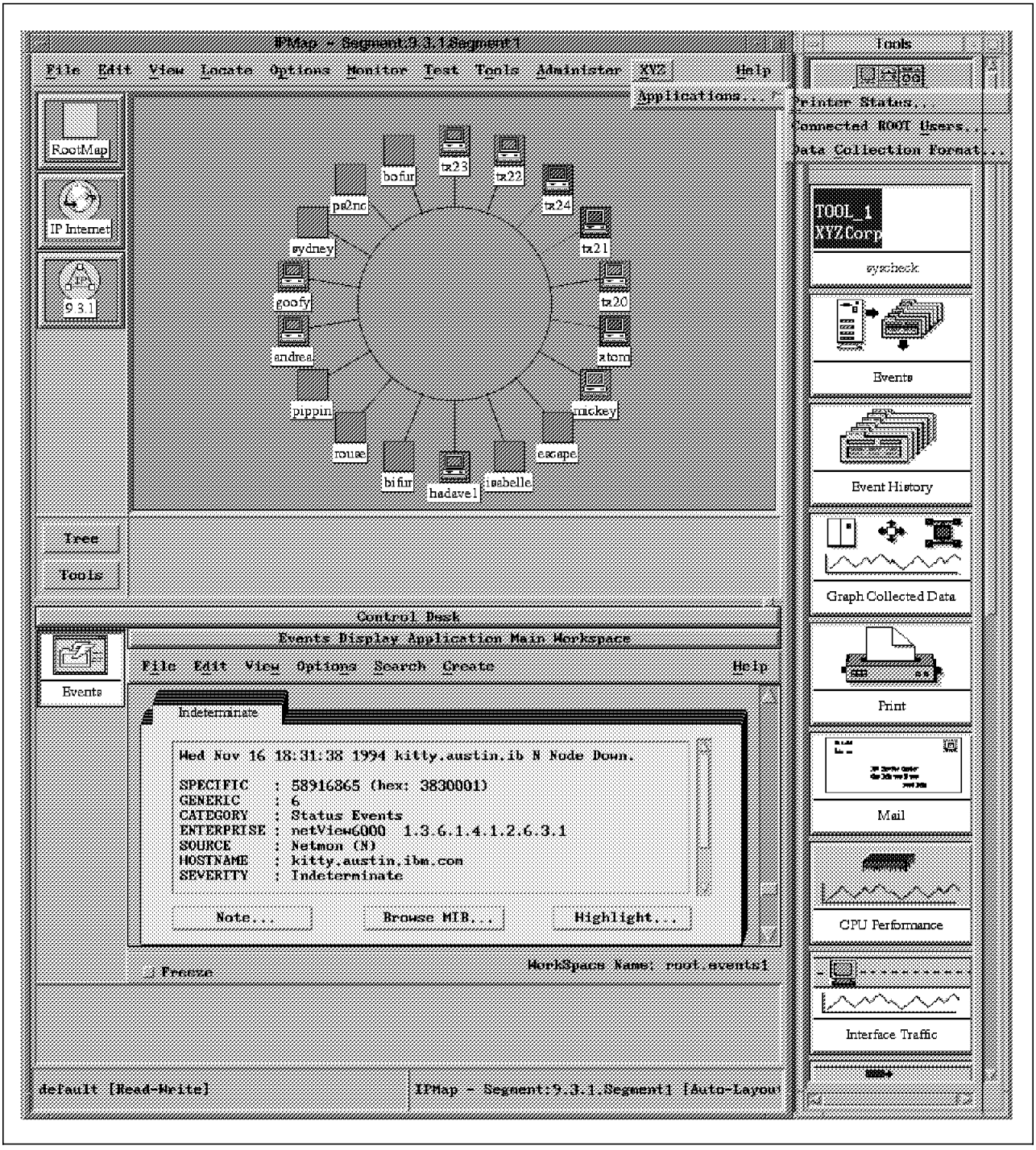

Figure 128. Customization of the NetView Graphical User Interface

The graphical result of the previous registration file XYZ.reg is shown in Figure 128.

Therefore, you can see that one can very easily integrate his own applications, running simple or more complex actions, such as personal tools, or even execution of other products (Trouble Ticket/6000, MPNP: application customizing IBM 6611 routers, ...).

B.1.2.18 Multi-Operator Management

A multi-operator management means that at least 2 different user profiles exploit NetView for AIX: an administrator profile, which will be root, since only root can run/stop NetView daemons, and a user profile authorized to run only the ovw graphical application (and not the daemons).

This user can have a personal map with read/write accesses, and he will open only in visualization (read/only and refresh) other active maps that do not belong to him. In the same way, the user will have his own event workspaces (traps) when he runs the Events or the Event History application.

We can go much further in the multi-operator management. It is possible to customize NetView's GUI *for every user profile*. The big interest of this possibility is, for example, to differentiate administrator and user menus. Another possibility is to define category of users that can access only to these or these menus.

To configure that functionality, you have to modify the .profile file of the user by adding the following line:

```
export OVwRegDir=/u/andrea/usr/OV/registration/C   (or any other directory)
```

All the registration files of menus, submenus, executed actions, ... of NetView ovw graphical application, are stored in the /usr/OV/registration/C directory.

You can now customize the user: andrea files in his own /u/andrea/usr/OV/registration/C directory, without taking the risk of touching general registration files in /usr/OV/registration/C, which only have one copy.

Features such as configuring a NetView for AIX Backup Manager and monitoring em and paging space are not covered in this cookbook. See the *NetView for AIX Version 3 Administrator's Guide* for more information.

B.2 Trapgend

This section will notably show you how to use trapgend to supervise applications.

B.2.1 Installation/Customization

Installation of the trapgend daemon on the local node (NetView) is automatically done during NetView installation phase. To install it on remote nodes, you use smit nv6000→Install/Configure subagent (trapgend) on remote RISC System/6000. You can also process a multiple installation via a shell script, by listing the concerned remote nodes. This is a kind of a shell script you can run:

```
/usr/OV/bin/nv6000_smit subagentR install node1 Root_Pwd public IP_@_NV/6000
/usr/OV/bin/nv6000_smit subagentR install node2 Root_Pwd public IP_@_NV/6000
/usr/OV/bin/nv6000_smit subagentR start node3 Root_Pwd public IP_@_NV/6000
/usr/OV/bin/nv6000_smit subagentR stop node4 Root_Pwd public IP_@_NV/6000
/usr/OV/bin/nv6000_smit subagentR remove node5 Root_Pwd public IP_@_NV/6000
```

You can see here, that apart from installing, you can also run other multiple operations. Allowed operations are: `install`, `start`, `stop`, `test`, `remove`, `addtrap`, `deletetrap`.

Which functions do the trapgend daemon bring to NetView? Trapgend is a free service furnished with NetView's licence. Its principal function is to wake up when an error flagged: `Alert=True` (or with an Alertable status) is written into the AIX errorlog. Trapgend will transform the error in a SNMP trap and send it to NetView, which has first been configured to be its trap destination in the /etc/snmpd.conf SNMP agent file.
Beforehand, the trap will be filled with the data reported in the errorlog, which you can visualize with the `errpt -a` command (the data is notably the error description with the different causes, recommended actions and detailed data).

The traps corresponding to standard errors of the AIX errorlog (whose list you can obtain running the `errpt -t` command) are configured in standard in NetView. You can find this configuration in the `Options→Event Configuration→Trap Customization` menu with the enterpriseID of netView6000SubAgent.

On the other hand, trapgend has a small private MIB, concerning especially the CPU and the filesystem load. If you have installed the Systems Monitor product or PTX and PAIDE (performance toolbox), on your platform, we would advise you to use their CPU and filesystem private MIB variables. Indeed, the trapgend CPU load variable gives some curious values that are a bit difficult to interpret.

B.2.2 Expanding the AIX Errorlog and Supervising the Applications

The `errpt -tF Alert=1` command gives the list of all errors having the Alertable status (meaning it is transformable into SNMP traps by trapgend). There is approaching 800 types of errors which are standard reportable in the AIX errorlog (see `errpt -t` command), alertable or not. They cover most of the AIX system problems, from a diskette defection to errors of configuration.

To position the error (identified by its ERROR_ID in hexadecimal) to an alertable status, you will have to run the following command:

```
echo =99999999:\nAlert=True | errupdate   #99999999: ERROR_ID in hexadecimal
```

By the way, to remove it from the AIX ODM database, the next command is to be executed:

```
echo -99999999: | errupdate   #99999999: ERROR_ID in hexadecimal
```

For the needs of applications supervision, you can define new *personal* errors in the following way:

1. Create a create_error1 file, similar to this:

```
SET E
+ "Enter here the Error Description"
SET F
+ "Enter here the Error Causes"
SET I
+ "Enter here the Installation causes"
SET P
+ "Enter here the Probable causes"
SET R
+ "Enter here the Recommended actions"
+ "Other Recommended actions"
SET U
+ "Enter here the User causes"
SET D
+ "Detailed datas:"
+ "Other Detailed datas:"
```

Figure 129. The create_error1 Sample File

2. Take it into account with the errmsg create_error1 command. The file create_error1.out is created.

3. Create a create_error2 file, like this:

```
+ERROR_LABEL:                  #ERR_APPL1, for example

   Comment  = "Comments on error"
   Class  = S                  #S:Software, H:Hardware, O:Operator

   Log  = True
   Report  = True
   Alert  = True               #this error will then have the Status Alertable

   Err_Type = TEMP             #TEMP:Temporary, PERM:Permanent, PERF:Performance,
                               #PEND:Pending, UNKN:Unknown

   Err_Desc = E000             #"E000" is to be recuperated in create_error1.out
   Prob_Causes = E000
   User_Causes = E000
   User_Actions = E000
   Inst_Causes = E000
   Inst_Actions = E000
   Fail_Causes = E000
   Fail_Actions = E000,E001

   Detail_Data = 80,0101,ALPHA
   Detail_Data = 80,0103,HEX   #ALPHA:Alphanumeric, DEC:Decimal,
                               #HEX:Hexadecimal; 80: length of the detailed datas
```

Figure 130. The create_error2 Sample File

4. Take it into account with the errupdate create_error2 command. The file create_error2.undo is created. You will find in this file the ERROR_ID (in hexadecimal) of the newly created personal error.

To supervise an application (a C program, for example), depending on your application state, it just takes writing the adequate ERROR_ID into the AIX errorlog indicating this state to be able to follow step by step the execution and the performance of the application.

```
/*********************************************************************
*  This program is used to generate an error in the AIX errorlog
*  It's compiled in the following way:
*              cc -o err_gene err_gene.c -lrts
*
*  For execution, he works with 3 or 4 parameters:
*  1) ==> The error in decimal or hexadecimal (preceded by 0x)
*  2) ==> a concerned resource name (tr0, hdisk0, ...) on max 16 characters
*  3) and 4) ==> detailed datas
*
*  Example: err_gene 39891074 hdisk1 DetailedDatas1 DetailedDatas2
*********************************************************************/
#include <stdio.h>
#include <sys/errids.h>

main(argc,argv)
   int argc;
   char *argv[];
{
   ERR_REC(200) *p_err;
   int Size;
   unsigned long errid;
   char nameres[ERR_NAMESIZE];
   char detail[101];
   int result;

   if (argc < 4)
        {
        printf("Usage: err_gene Error_ID(Decimal or 0xHexa)
ResourceName DetailedDatas1 DetailedDatas2(optional)\n");
        exit(0);
        }

   if (strncmp(argv[1],"0x",2) == 0)    sscanf(argv[1],"%x",&errid);
   else                                 sscanf(argv[1],"%d",&errid);

   Size = sizeof(ERR_REC(200));
   p_err = malloc(Size);

   strncpy(nameres,argv[2],ERR_NAMESIZE);
   p_err->error_id=errid;
   strncpy(p_err->resource_name,nameres,ERR_NAMESIZE);
   strncpy(detail, argv[3], 100);
   sprintf(p_err->detail_data,"%-100s%-7s", detail, argv[4]);
   result=errlog(p_err,Size);
   if (result)
        {
        printf("Problems to write in the AIX errorlog\n");
        }
   else
        {
        printf("Error generated in the AIX errorlog: \n");
        printf("\tin decimal: %d\n",errid);
        printf("\tin hexadecimal: 0x%x\n",errid);
        printf("\tName of the resource: %s\n",nameres);
        }
   exit(0);
}
```

Figure 131. C Program Generating an Error in the AIX Errorlog

Figure 131 gives the code extract to be included in the C application to be managed, which will report its errors in the AIX errorlog.

The trapgend daemon sends a trap to the NetView for AIX manager each time your C application calls the previous code.

Obviously, you will have to define that trap in the Options→Event Configuration→Trap Customization menu of NetView so that it can be correctly transformed into a NetView event card. The trap is to be configured in the following way:

```
EnterpriseId = netView6000SubAgent
Numero Generique = 6
Numero Specifique = 99999999
Event Log Message =
Error: $%20 \nProbable Cause: $%21 \nUser Cause: $%22 \nInstall Cause: $%24
       \nFailure Cause: $%26 \nDetailed Datas: $%28 (for example)
Category = Error Events
Severity = Critical (for example)
Popup Notification = trapgend error 99999999 generated  (for example)
```

It is important to notice that the trap has to be entered in *decimal* in NetView's menu. Since we know only its hexadecimal form, you must convert it to decimal. This is a possible way to make the conversion, using the bc command (korn shell):

```
You type:      bc
You type:      ibase = 16
You type:      99999999   (hexadecimal)
The system gives: 2576980377 (decimal)
Ctrl-D to exit
```

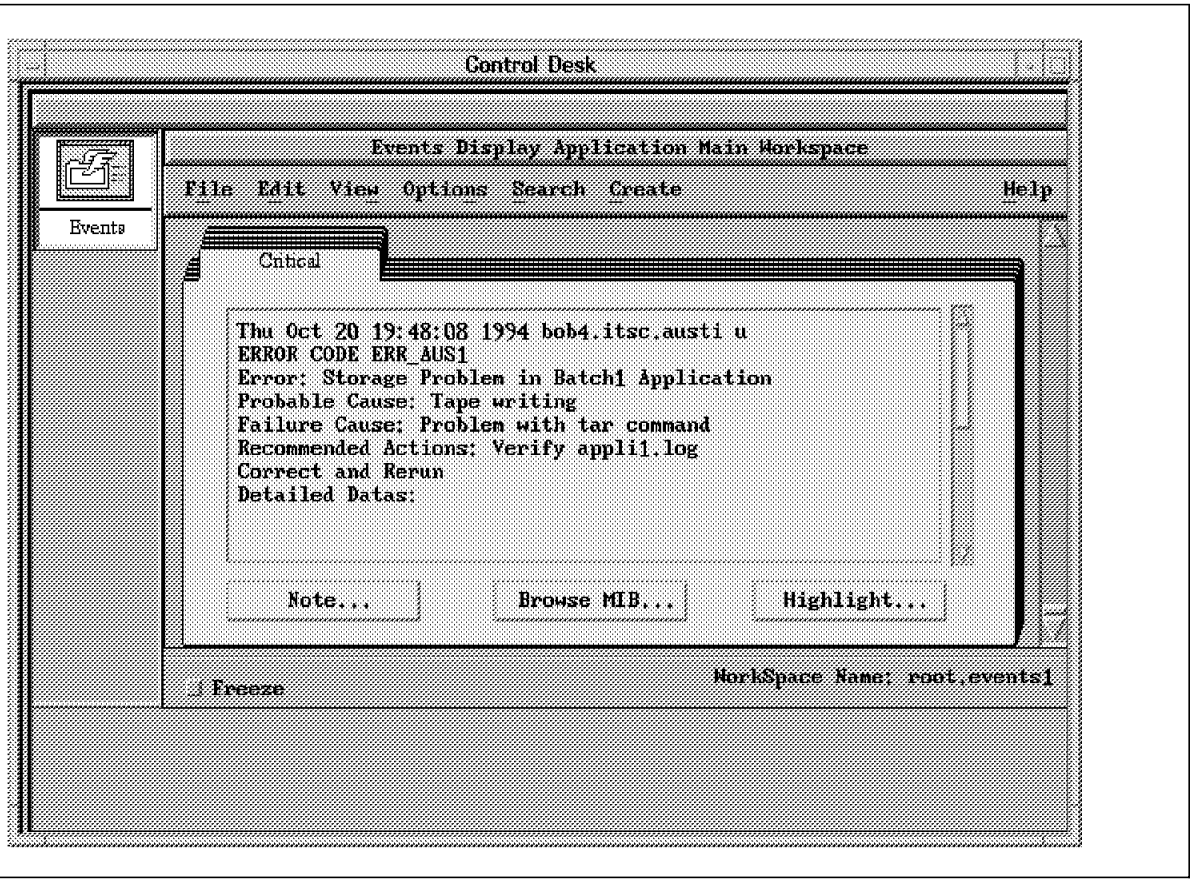

Figure 132. Example of a Trapgend Trap

Figure 132 shows an example of a trapgend trap, displayed by NetView.

B.3 SNMP Interface with PTX V1R2 for AIX 3.2.5

The performance toolbox xmservd agent, launched by the xmservd -p3 command, can *dynamically* generate a private SNMP MIB, which can be then used by NetView. The private MIB contains variables of the UNIX system that are very interesting, for their meanings (CPU, Memory, FileSystems, Disks, Processes, IPC, System Calls, LAN, ...) and number.

Processing the SNMP interface is explained well in PTX User's Guide documentation. Lets resume here with what has to be done:

- Write dosmux in the /etc/perf/xmservd.res file. This file will be copied from the /usr/lpp/perfagent directory
- Stop xmservd and rerun it so that it takes into account that modification
- Do a SIGINT: kill -2 PID_xmservd. This generates a MIB in /etc/perf/xmservd.mib. We advise you to copy the MIB into /usr/OV/snmp_mibs NetView directory.
- Run make -f Make.mib (Make.mib is located in /usr/samples/perfagent/server). This updates the /etc/mib.defs file. It is now possible to query and modify the PTX variables, with the snmpinfo command (get, get_next, set).

- Load the xmservd.mib MIB in NetView to make it queriable in the menus MIB Browser, Data Collection and Thresholds, and MIB Application Builder. To do that, call the Options→Load/Unload MIBs menu.

If you are unable to make the SNMP interface run correctly, there may be a SMUX (SNMP Multiplex) conflict with other SNMP proxy-agents. To solve this problem, have a unique Enterprise ID for the xmservd subagent in the files /etc/snmpd.conf and /etc/snmpd.peers:

- /etc/snmpd.conf file:

```
smux            1.3.6.1.4.1.2.3.1.2.1.3         xmservd_pw      # xmservd
```

- /etc/snmpd.peers file:

```
"xmservd"       1.3.6.1.4.1.2.3.1.2.1.3        "xmservd_pw"
```

B.4 Systems Monitor for AIX V2R1

This section develops a cookbook on the Systems Monitor for AIX V2R1 product.

B.4.1 Installation/Customization

Systems Monitor for AIX is installed using smit installp. The product takes about 35MB of disk space on /usr if you install the Configuration Application (the EUI), the Mid-Level Manager (MLM), and the System Information Agent (SIA). 10MB more are required for online documentations (DynaText views and Systems Monitor User's Guide book).

The installation process on the NetView local node automatically starts the midmand and sysinfod daemons. They will also be started at the machine's boot (/etc/rc.tcpip file has been updated during installation). You can start the daemons with flags (if you want to specify another configuration directory, collect file, log file, ...).

Systems Monitor main directory is /usr/adm/smv2. The collection and the log directories are located there, as well as the MLM and SIA configuration directory. *All files* in these directories will be charged in Systems Monitor's local MIB each time you launch the midmand or sysinfod daemon.

You will also find the ovsnmp.conf file in /usr/adm/smv2. This file belongs to a manager (NetView or Mid-Level Manager). The /etc/snmpd.conf file belongs to an SNMP agent. Configurations (essentially community names used by the manager to query the agent) entered in those 2 files must match, for the request (SNMP get, get_next, set) to be accepted by the agent. If not, authentication errors will be sent from the agent to its manager and possibly to NetView using the MLM's trap destination table.

For example, if you configure a MLM in your network, you will need the following customization:

- /etc/snmpd.conf file on the SIA:

```
community public
community auscom2 mlm_node 255.255.255.255 readWrite
community auscom1 netview_node 255.255.255.255 readWrite
```

- /usr/adm/smv2/ovsnmp.conf file on the MLM:

```
sia_node:auscom2:*:8:3:300:::
```

- /etc/snmpd.conf file on the MLM:

```
community public
community auscom1 netview_node 255.255.255.255 readWrite
```

- /usr/OV/conf/ovsnmp.conf file on NetView:

```
sia_node:auscom1:*:8:3:300:::
mlm_node:auscom1:*:8:3:300:::
```

  Also a trap destination must be set up for mlm_node to forward traps to netview_node. Do not configure it in its /etc/snmpd.conf file, but via the the MLM's trap destination table.

  To start/stop MLM and SIA daemons, you can use the AIX command line or smit smv2 (menus are: Remote or Local operations→Install/Control the SIA or Install/Control the MLM). You can also use NetView EUI's menu: Administer→Systems Monitor V2→ Install/Control from map, especially for remote multiple operations (installation, start/stop, ...). Shell commands are:

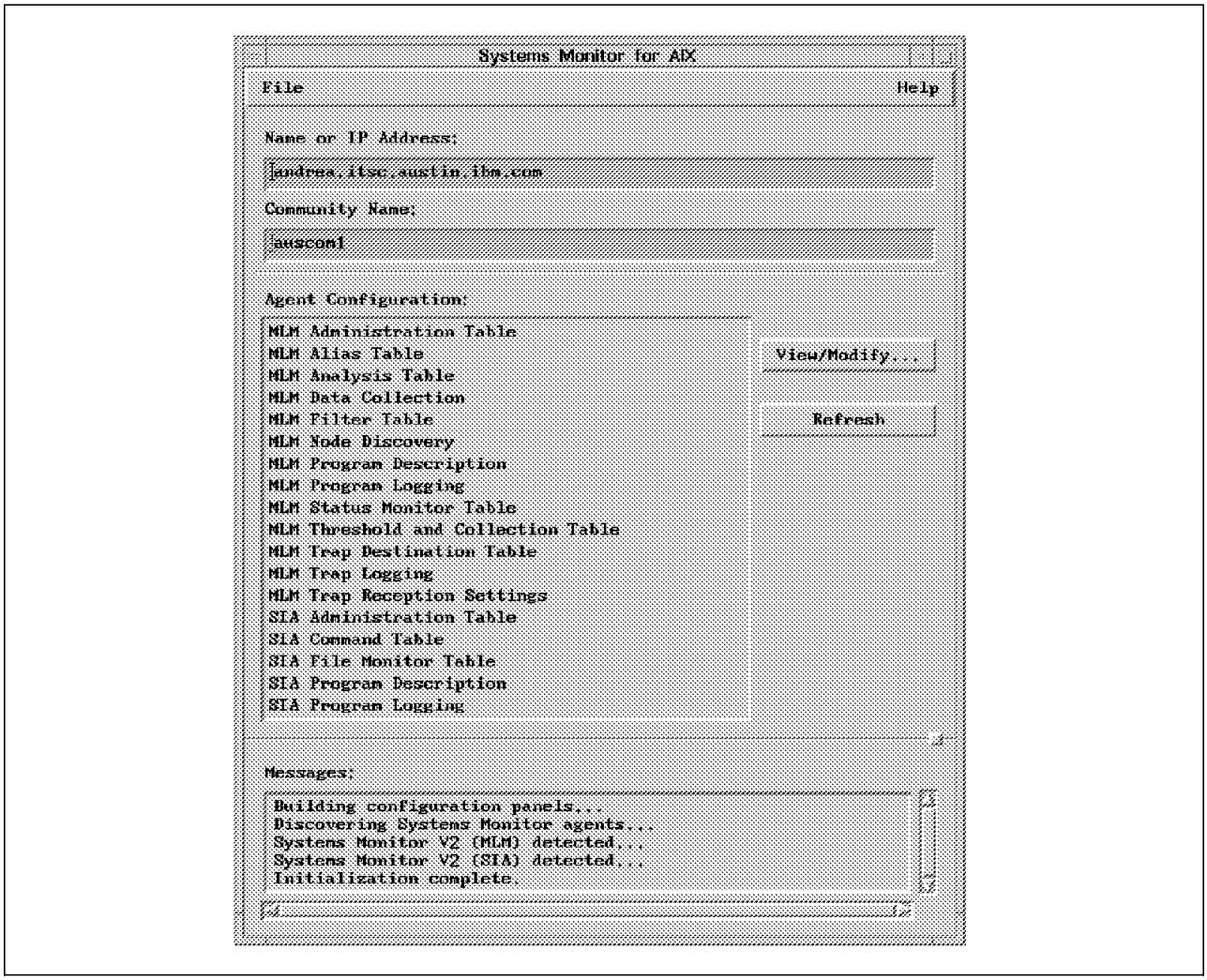

Figure 133. Systems Monitor for AIX Graphical User Interface

  - smsia: Starts the sysinfod daemon (with saved flags, if you have saved any)
  - smsia stop: Stops the sysinfod daemon
  - smmlm: Starts the midmand daemon (with saved flags, if you have saved any)
  - smmlm stop: Stops the midmand daemon

– smconfig -h node: Starts the Configuration Application on a node hosting MLM or SIA program (see Figure 133).

Use of the Systems Monitor on NetView's GUI is accessed by the menus:

– Monitor→Mid-Level Manager, Monitor→System Information: Queries MLM and SIA private MIBs (graphical or table results).

– Tools→Systems Monitor V2 Configuration Application: Runs the EUI application that customizes Systems Monitor MIB management tables (Command Table, Threshold Table, ...), and saves them in configuration files, so that those new files are used any time the daemons (sysinfod, midmand) are started. You have first to click on a node in the topology map to query the EUI. The smconfig -h node shell command also starts the Configuration Application on a node hosting MLM or SIA program.

– Administer→Systems Monitor V2→SMIT: Runs SMIT Systems Monitor.

B.4.2 Configuration Files

Systems Monitor configuration files are located in /usr/adm/smv2/sia/config for the System Information Agent and in /usr/adm/smv2/mlm/config for the Mid-Level Manager. The first time you run the SIA or the MLM, the default configuration files sysinfod.config and midmand.config are loaded respectively. If you move modifications in the Management MIB table (creation/suppression of commands in the Command Table, of threshold entries in the Threshold Table, etc...), they will be automatically saved in the /usr/adm/smv2/log directory, in the files smMlmCurrent.config and smSiaCurrent.config respectively (these files are called *resume files*). Next time you start the daemons, the resume files are loaded.

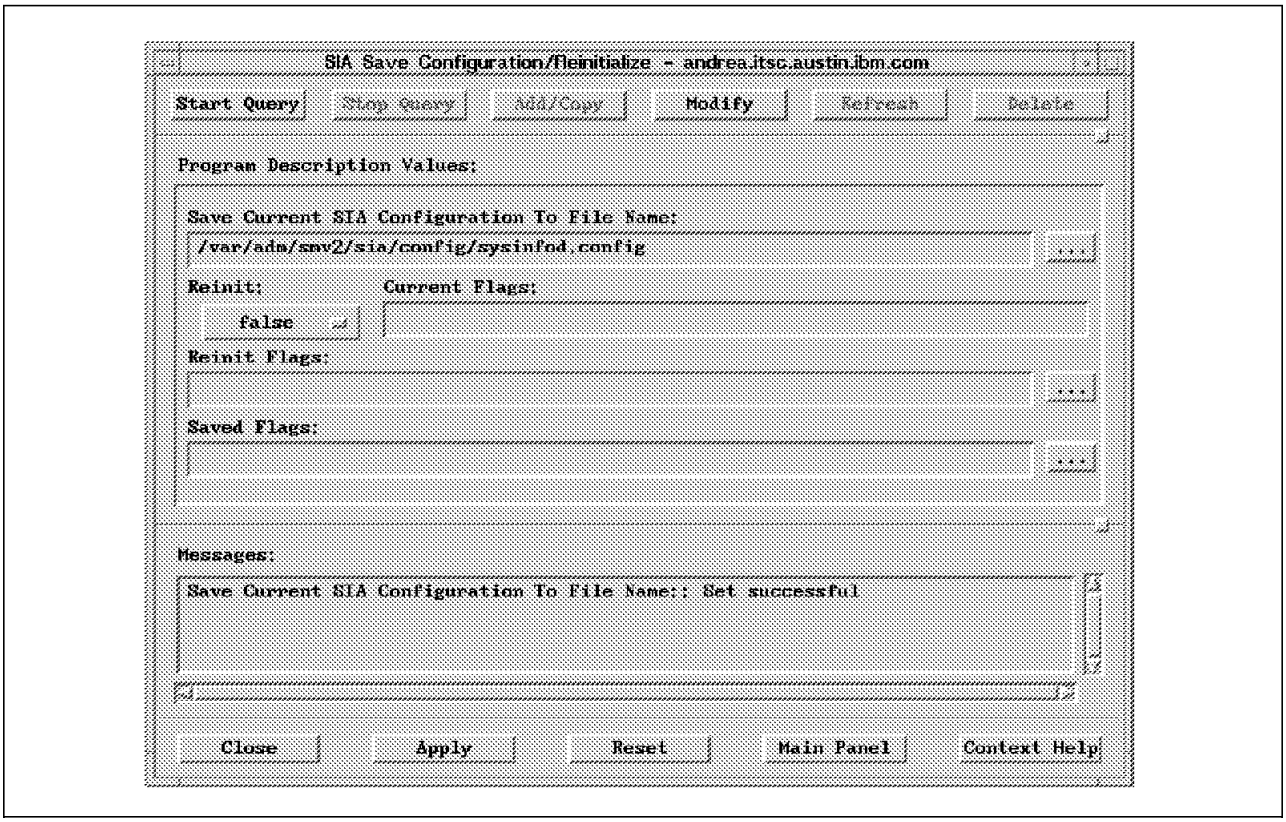

Figure 134. Saving Systems Monitor Configuration Files

You may want to save your work in the default configuration files or other files so that you can reinitialize the daemons with different Systems Monitor configuration files or copy them to other network nodes (Figure 134 on page 211).

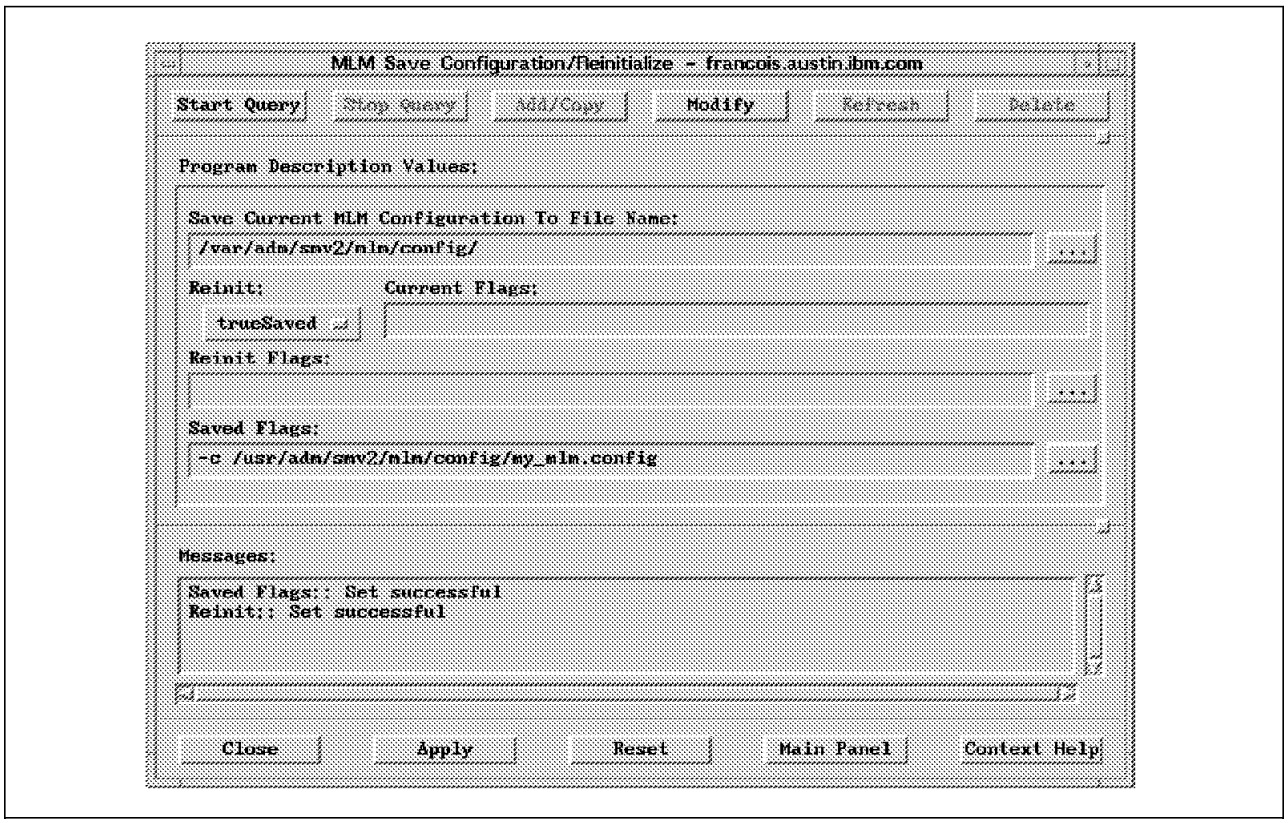

Figure 135. Reinitializing Systems Monitor with Another Configuration File

To start the daemons with your own configuration files (user-specified or default), you have to reinitialize them, like in Figure 135. You then stop and restart the daemons, so that the new configuration files are taken into account. By the way, the current resume files are then removed and new resume files are created from the configuration files you specified.

B.4.3 Management MIB Tables

The management functions of Systems Monitor for AIX including, polling, thresholding and data collection, analysis, trap filtering, node discovery and status monitoring, are provided by the Mid-Level Manager. This section will explain how to configure these tables and describe how they interact. It will also develop the SIA Command Table which extends Sys tems Monitor private MIB by creating variables customized to your applications, and the SIA File Monitoring Table which monitors the output of applications, or other important information, and take action based on that output.

B.4.3.1 The MLM Alias Table

Distributed management in Systems Monitor is implemented by the concept of an alias representing a group of nodes. All operations in the MIB management tables concern a set of nodes (alias) or a node (hostname or IP address).

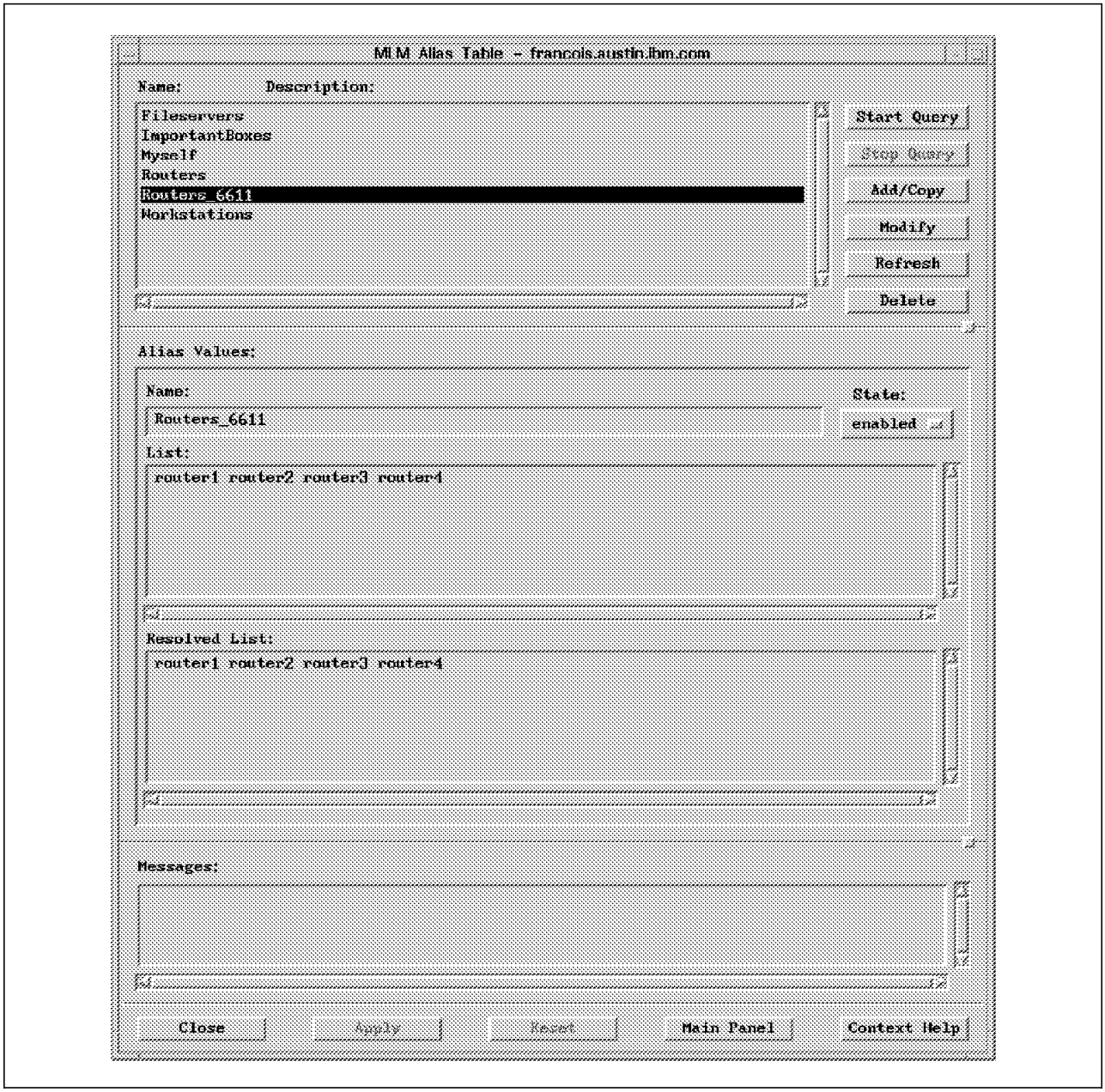

Figure 136. Systems Monitor Alias Table

For example, the following entry in the MLM Alias Table creates a set of all the IBM 6611 routers of a network (see Figure 136).

```
smMlmAliasName[Routers_6611] =  "Routers_6611"
smMlmAliasList =  "router1 router2 router3 router4"
smMlmAliasState =  enabled
```

Operations of data collection in the Threshold Table targeting MIB variables on Routers_6611 will simultaneously apply to the 4 routers.

B.4.3.2 The MLM Node Discovery Table

An important change brought by Systems Monitor V2 is the full delegation of network management tasks (new node discovery, status and configuration change polling) to the MLM by NetView. To process this functionality, you have to specify to the daemon of discovery, netmon, which MLMs are concerned. So, exit all running NetView GUIs, and go to `smit nv6000→Configure→Set options for netmon daemon`. Specify there that you want to use Systems Monitor MLM feature, and indicate the full path of Systems Monitor MLM seedfile (/usr/OV/seeds/mlmseed, for example).

```
# /usr/OV/seeds/mlmseed file
andrea.itsc.austin.ibm.com     # IP address is also possible
```

From now on, as NetView for AIX finds an MLM in the network, it updates that MLM Alias Table with the nodes within the MLM's subnetwork (*same subnetmask*), adds itself to the MLM Trap Destination Table, and tells the MLM to take over status checking and discovery in the same subnet. As the MLM finds nodes, it will forward discovery and interface status traps to NetView, which will then make all appropriate changes (new nodes, status -color- changes, ...) in the topology map.

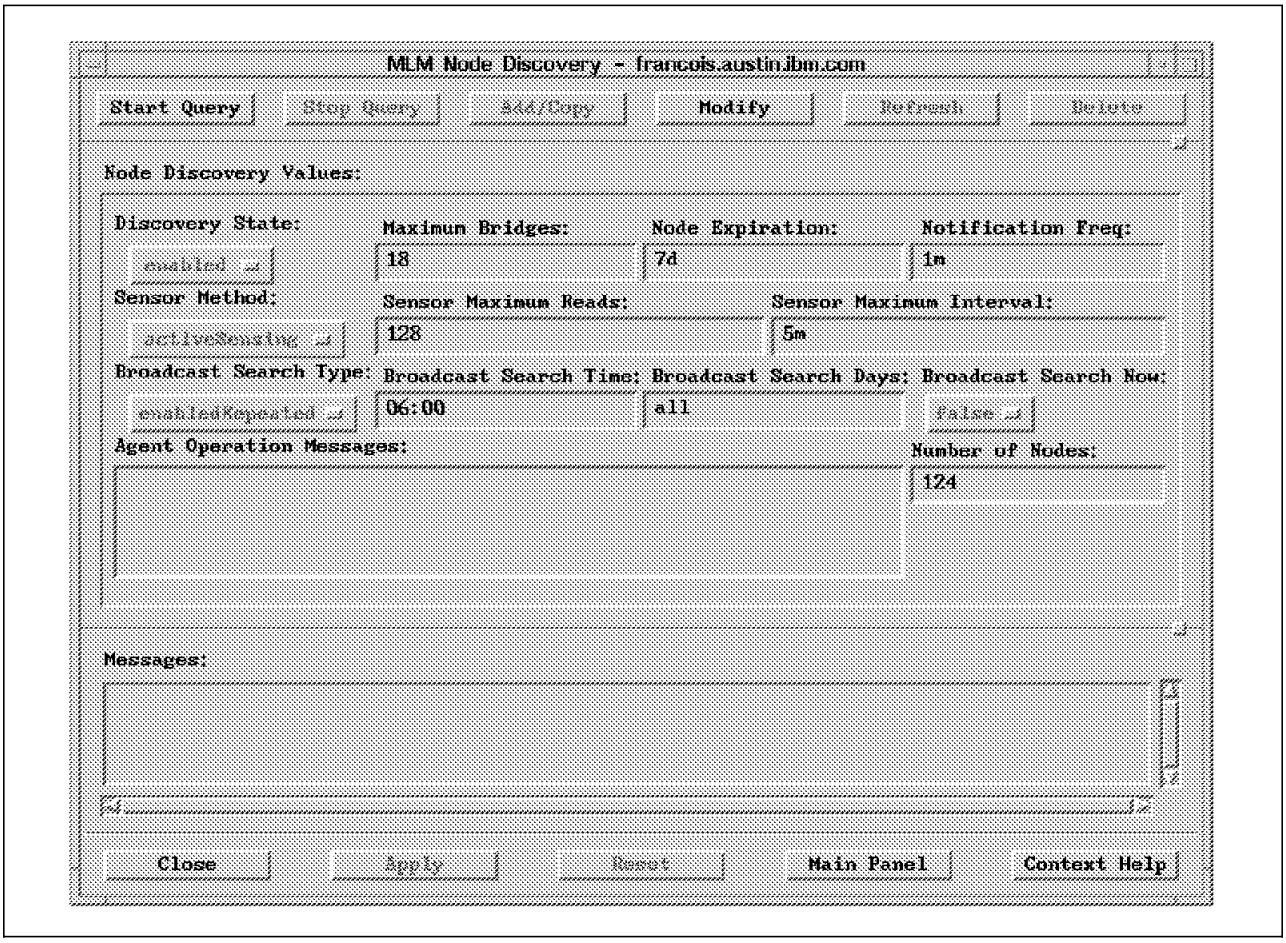

Figure 137. Running MLM Node Discovery

You will now have to customize the MLM Node Discovery Table as shown in Figure 137.

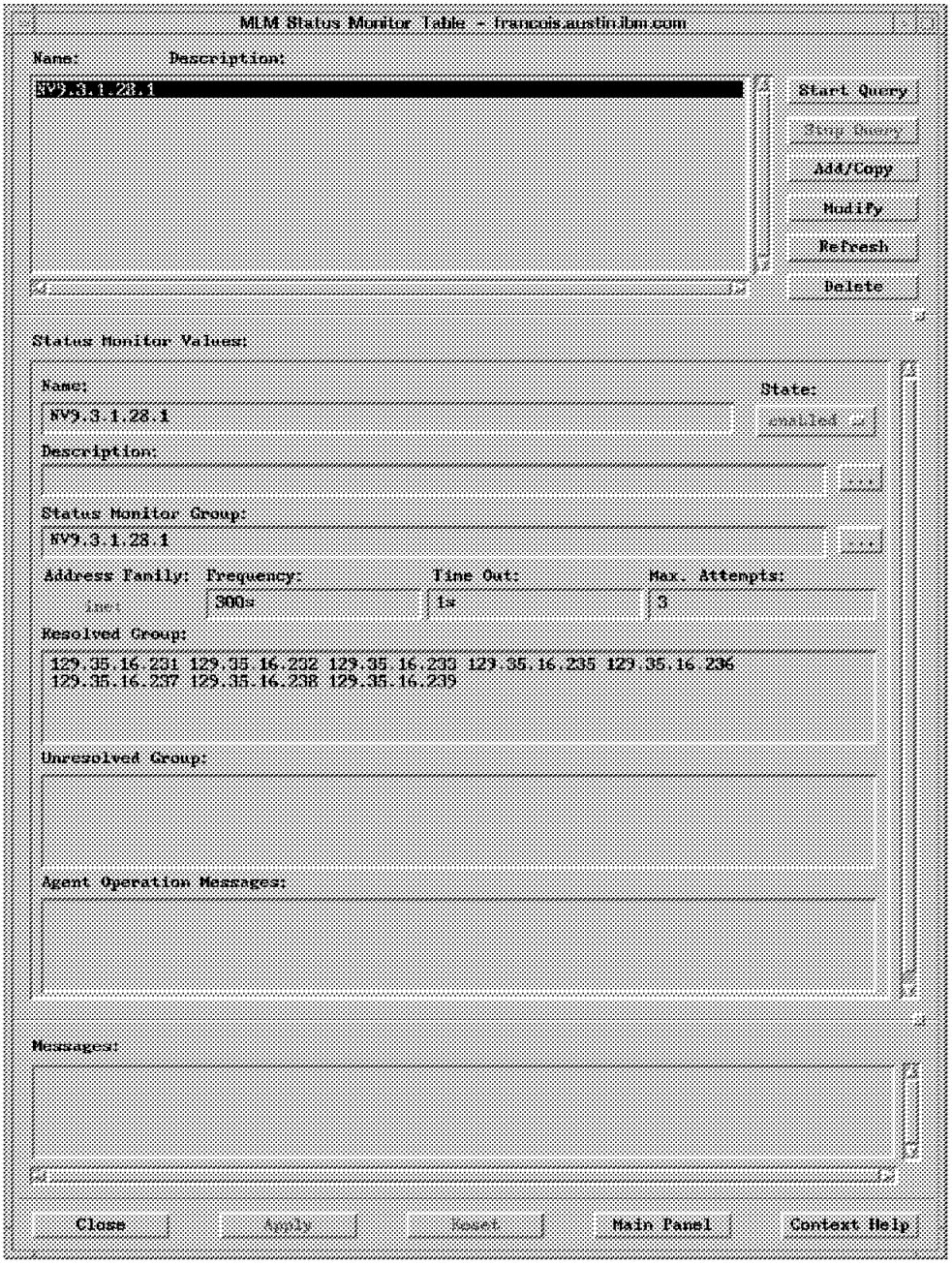

Figure 138. Systems Monitor Status Monitor Table

As nodes are discovered, they are grouped under aliases and added to the Alias and Status Monitor Tables (see Figure 138). The Status Monitor Table determines default polling frequencies (5 minutes), time-out value (1 second), and the maximum retries (3). The alias has the form NVxxx.xxx.xxx.xxx.n, where xxx.xxx.xxx.xxx is NetView for AIX IP address and n is the group's number. Each alias has a maximum of 24 nodes.

B.4.3.3 The SIA Command Table

The Command Table is used to extend the Systems Monitor private MIB to variables customized to your application. With the ability to supervise the shared memory it accesses, Systems Monitor can manage your application from its heart.

Creation of New Private MIB Variables

The main function of the Command Table is to create new private MIB variables that do not exist in any standard provided MIB (Netview, Systems Monitor, trapgend, Performance Toolbox, ...). Notice that those variables may already exist in standard MIBs, but not in the form you would like to exploit it (counter instead of percentage, for example).

This *fundamental* functionality of the product, very simple of use, makes it possible to generate, without long and fastidious developments, proxy-agents on the newly created private MIB variables. Indeed, it is *not* a new proxy SNMP agent daemon that is created, in the sense of an independent daemon that can be installed on other machines. By using the Command Table, we extend the sysinfod SIA proxy-agent, as well as its private MIB, to the new variables, which, in terms of functionality, can be considered to be the same thing.

Lets take the following example. We want to have the number of collisions coming on the tr0 token-ring interface of a node. We then create an entry in the Command Table with the following parameters:

```
smSiaCommandName =  "COLLIS"
smSiaCommandGetStringAndParameters =  "netstat -i | grep tr0 | grep -v Link | awk
'{print $9}'"
smSiaCommandOutputResultIndex =  integer
smSiaCommandState =  enabled
```

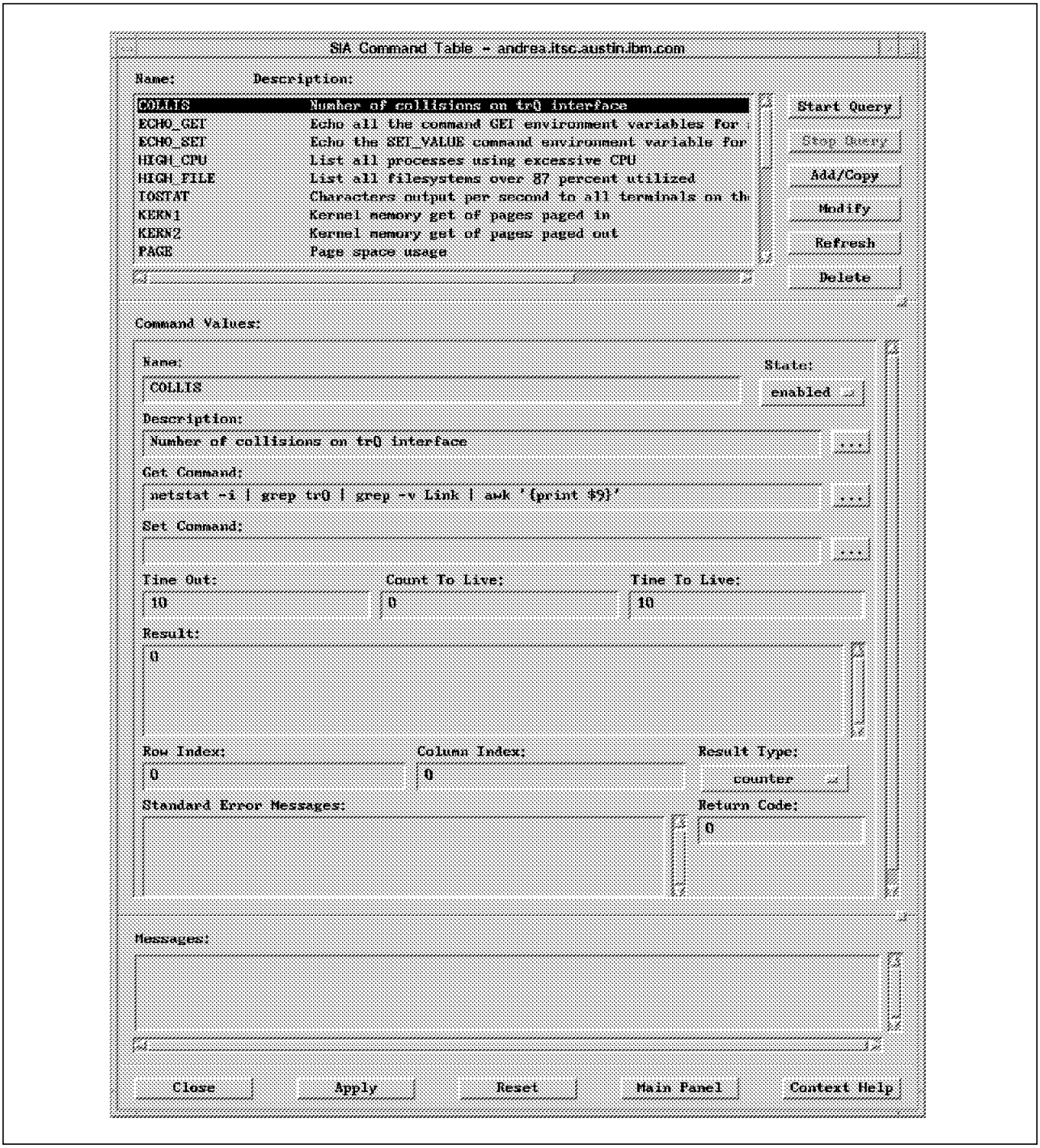

Figure 139. Systems Monitor Command Table

The Command Table panel you should get is shown inFigure 139

From now on, there is a new variable in Systems Monitor MIB, named .1.3.6.1.4.1.2.6.12.4.1.1.14.COLLIS, which gives the collisions number on the tr0 interface of a node, each time you run a Start Query on the COLLIS entry of the Command Table and also in NetView MIB Browser.

What is the SNMP label of the new variable? It just takes remembering the integer type we gave it and that it was created in the Command Table. The

variable will then be located in the subtree .iso.org.dod.internet.private.enterprises.ibm.ibmProd.systemsMonitor6000.smSiaCommand.smSiaCommandTable.smSiaCommandEntry, and its value will be a certain instance of the smSiaCommandIntegerResult variable. The instance is its name, COLLIS, translated in the ASCII form. In the SNMP figured writing, it gives: .1.3.6.1.4.1.2.6.12.4.1.1.14.COLLIS.

Thus, any new MIB variable (of type integer) created by the Command Table will be labeled .1.3.6.1.4.1.2.6.12.4.1.1.14.Name with Name being shown in its ASCII form writing in NetView menus (MIB Browser, ...) and directly displayed as Name in all Systems Monitor menus.

It is important to notice that these new variables will be treated as any other existing MIB variable (of trapgend, of router 6611, of CISCO). They can be graphically displayed (using NetView xnmgraph program), be supervised by the data collection and thresholds Systems Monitor application (SIA Threshold Table), and also analyzed by the SIA Analysis Table. This makes up a very interesting *management chain* built by Systems Monitor automation tables.

On the other hand, the Shell command to be executed in the smSiaCommandGetStringAndParameters field is, in our example, a simple command. Nothing prevents you from putting a much more complex program (shell script, C, ...) in there. Indeed, that field is considered to be an opening on the Korn Shell.

Supervision of Kernel or Shared Memory

Another essential use of the Command Table (illustrated by the entries KERN1, KERN2 and SHARED1 in the sample sysinfod furnished with the product) is the supervision of Kernel Memory or Shared Memory used by a program. To do that, it just takes informing Systems Monitor on which kernel external symbol or shared memory access key and semaphore, offset and memory length it has to run a query.

The /usr/lpp/smsia/original/sharedmem.c C program is given in sample of an application supervised in Shared Memory mode.

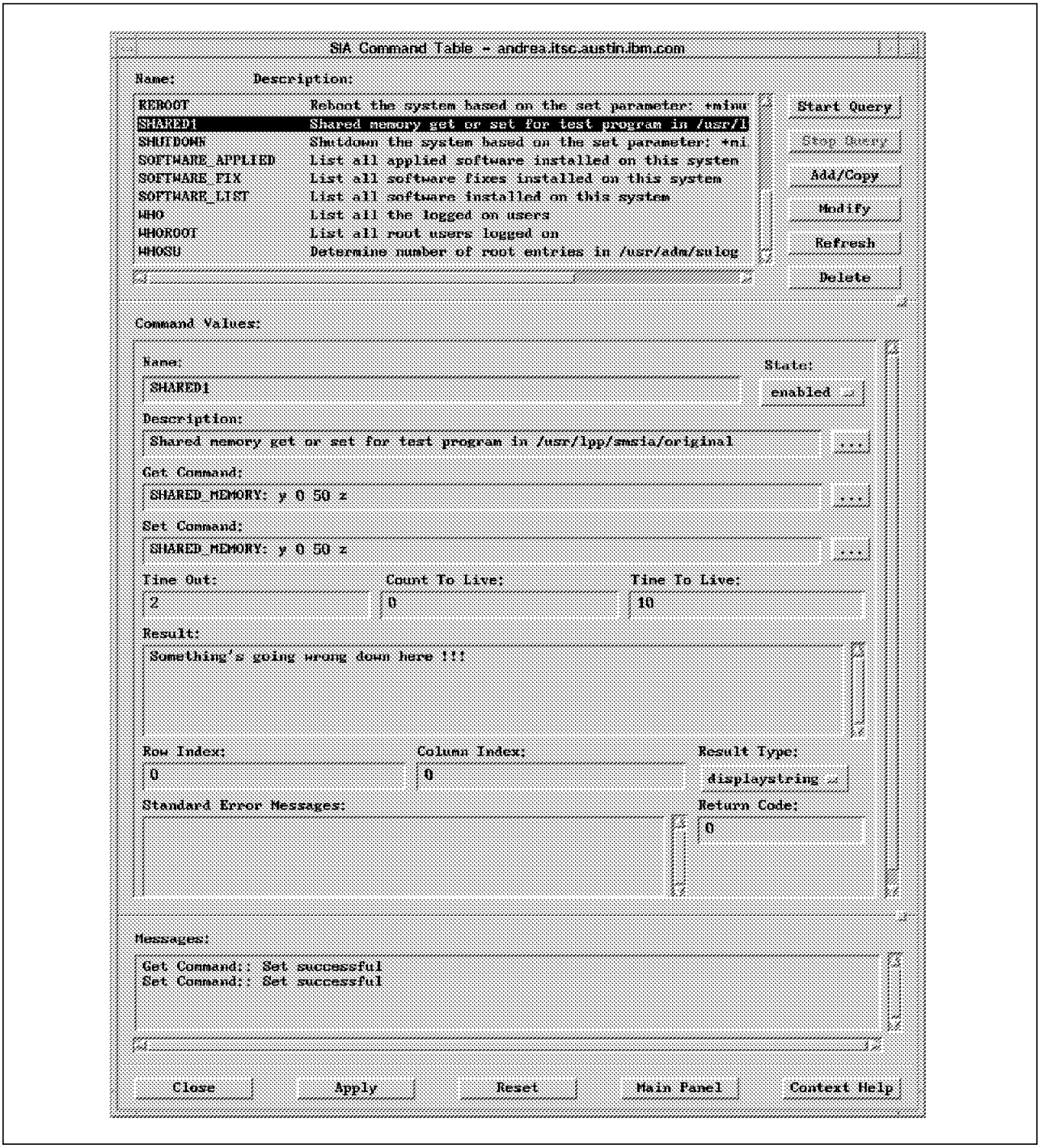

Figure 140. Querying a SharedMemory in Systems Monitor Command Table

When you test it, do not exit (let it run) while you consult the Shared Memory zone in the Command Table (see Figure 140).

You can also update this zone. To do this, you have to fill up the Result field of the SIA Command Table with the new value to put in the Shared Memory. Having written datas in that field will run the command in the smSiaCommandSetStringAndParameters field. Clicking on Modify and then Start Query, updates the Shared Memory.

To supervise a zone in the Kernel, it just takes indicating in the Command Table which external symbol of the kernel (given by the `nm -e /unix` command: for example `vmminfo`) you want to query, as well as the offset and the length (octets number) of the zone.

Note: If you have timeout problems on a Start Query in the automation tables, increase the `smuxtimeout` parameter in the SIA agent /etc/snmpd.conf file in the following way:

```
snmpd      maxpacket=16000 smuxtimeout=1800    # Systems Monitor for AIX
```

Basically, its value is 1 minute (60s). Do not hesitate to increase it considerably. Every Command Table entry disposes of a TimeOut field in the same way. Depending on the complexity of the command, you have to dimension it well:

```
smSiaCommandTimeOutValue =  20 #20 seconds: reasonable maximum not to exceed
```

B.4.3.4 The MLM Analysis Table

This table is used to create new MIB variables, that evaluate arithmetical expressions on other existing MIB variables, local or remote (remote variables belong to nodes defined by their hostname, IP address or alias). The MIB variables can be standard (MIB-II), or private (Systems Monitor, trapgend, cisco, ...). They can also be new variables defined in the SIA Command Table or even the SIA Analysis Table. This is, according to us, very interesting. Possible mathematical operators are multiplication, division, addition, substraction, remainder, unary minus, logical operations (AND, OR), bitwise shift, and grouping (with ()). other functions are accepted, like the sum, difference, average, minimum, maximum, absolute value, delta, value (for counter MIB variables), and random number.

Here is an example:

```
smMlmAnalysisName =  "ObstructionRate"
smMlmAnalysisState =  enabled

smMlmAnalysisLocalRemoteMIBVariableExpression =  "@sum(SubNetwork1:.1.3.6.1.4.1.
2.6.12.4.1.1.14.COLLIS)*100/@sum(SubNetwork1:.1.3.6.1.4.1.2.6.12.4.1.1.14.OPKTS)"

smMlmAnalysisPollTime =  "10m"
smMlmAnalysisResultIndex =  counter
```

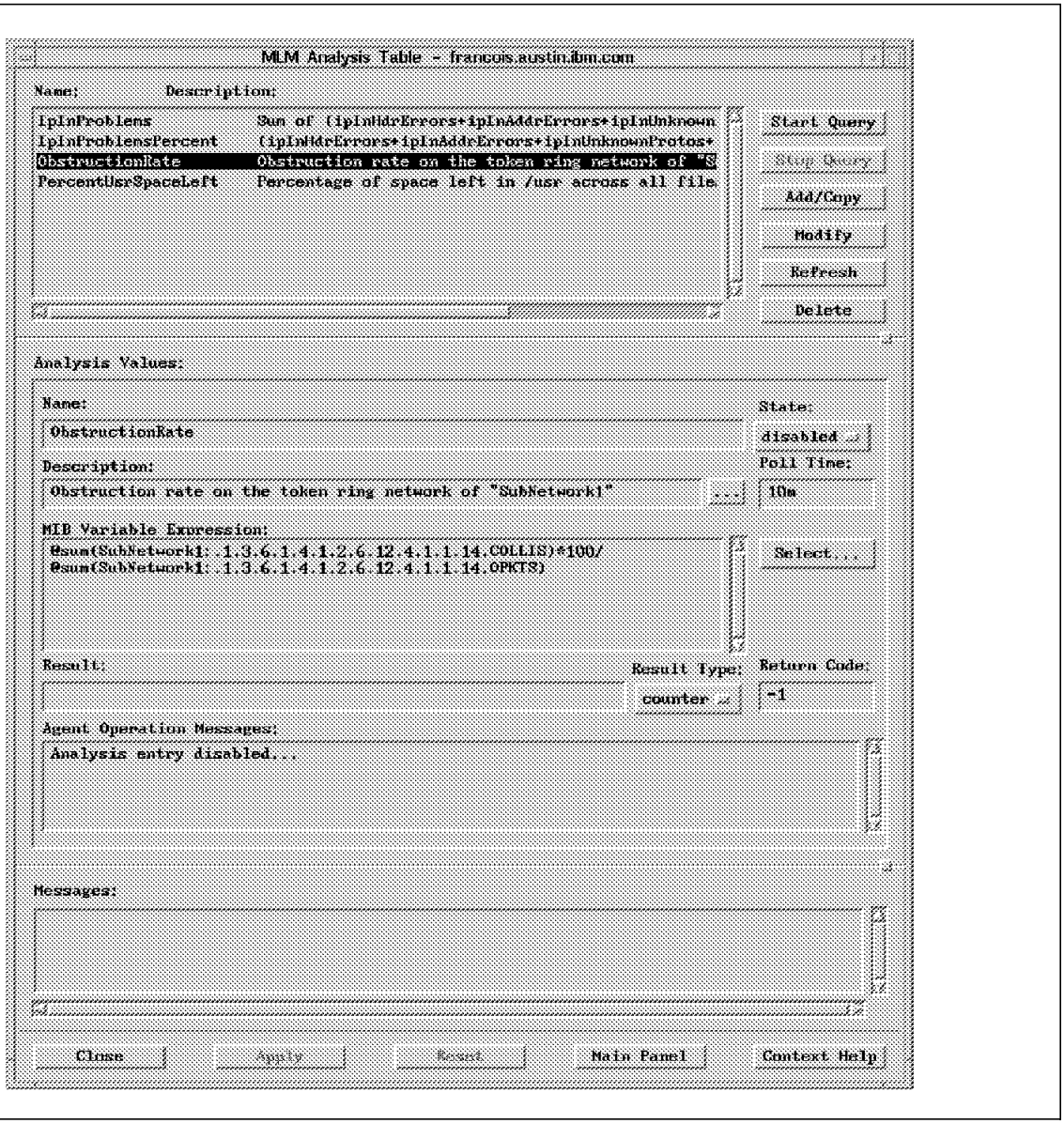

Figure 141. Systems Monitor Analysis Table

In Figure 141, we have created a new MIB variable, named .1.3.6.1.4.1.2.6.12.6.1.1.9.ObstructionRate (see explanation of this SNMP writing in the section B.4.3.3, “The SIA Command Table” on page 217). It is calculated every ten minutes, and polling is done on Command Table variables belonging to nodes with SubNetwork1 alias.

The variable we just created gives the obstruction rate on a token-ring network. It is evaluated by the total of collisions on all tr0 interfaces over the total of output packets on the same interfaces in SubNetwork1. The amount of output packets on the tr0 interface of a node is given by the OPKTS Command Table entry:

```
smSiaCommandName =  "OPKTS"
smSiaCommandGetStringAndParameters =  "netstat -i | grep tr0 | grep -v Link | awk
'{print $7}'"
smSiaCommandOutputResultIndex =  counter
```

B.4.3.5 The MLM Trap Destination Table

The Trap Destination Table is used to configure one or many network managers (NetView or Systems Monitor Mid-Level manager) to receive traps from SIA nodes. The manager (represented by a hostname, an IP address or an alias) listens on the UDP port 162 by default. You can specify another protocol (TCP), as well as another port number. You can also use a GLOBAL filter mask (254, which in hexadecimal equals to fe -see snmpd.conf, line trap-, sends all traps). The trap destination table is mainly used with the MLM Filter Table. Activating a filter entry in this table overtakes the filter mask configured here.

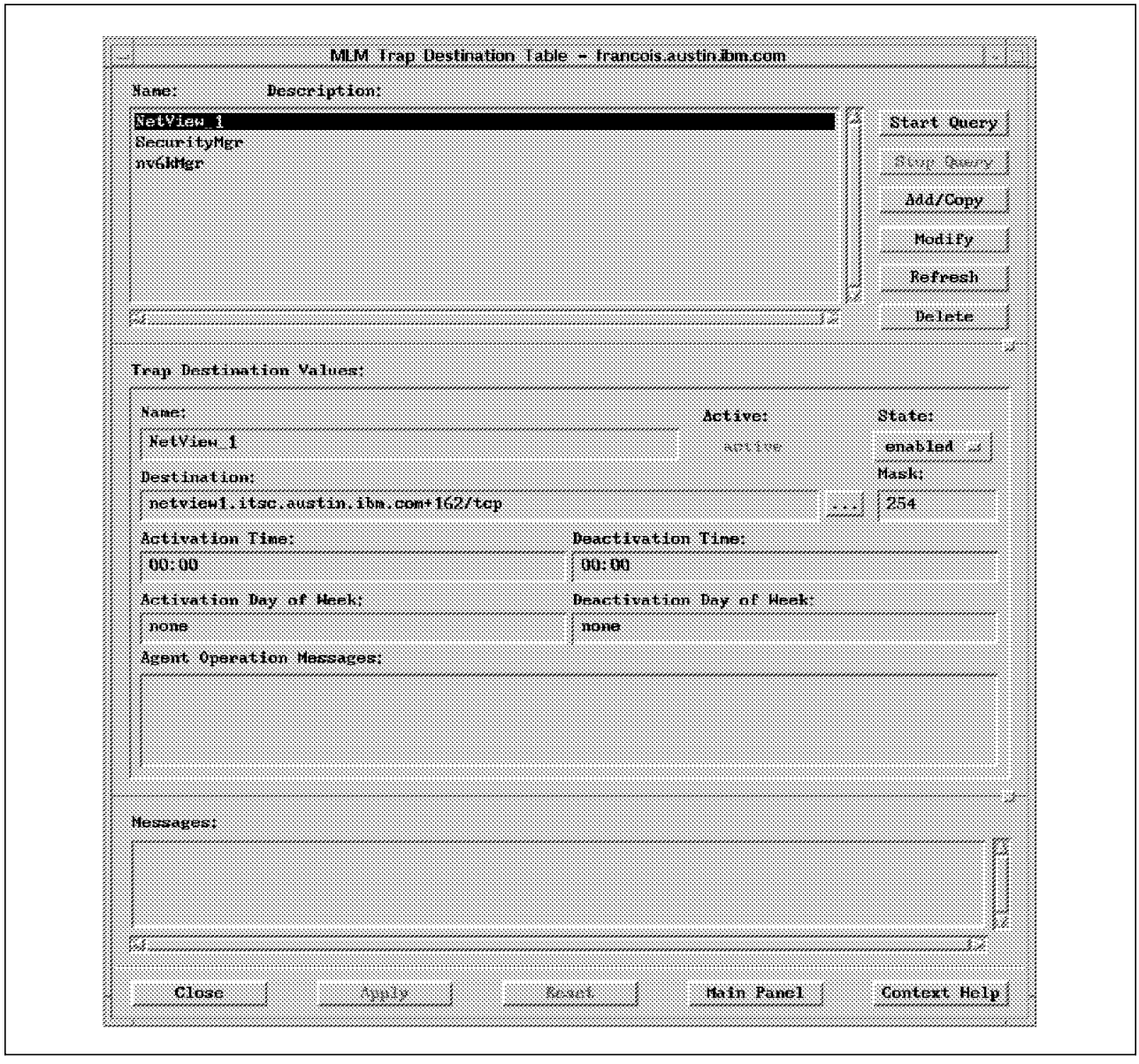

Figure 142. Systems Monitor Trap Destination Table

Figure 142 shows an example of the configuration.

B.4.3.6 The MLM Threshold and Collection Table

Thresholding provides a means for managing applications and systems at the node level.

Implementation

This table processes data collection on a MIB variable (standard MIB-II or any private MIB: Systems Monitor, trapgend, router 6611, ...) on a polling interval you set. The variable is local or remote (on other nodes, for distributed management). To indicate to whom it belongs, it just takes preceeding it with an alias, a hostname or an IP address.

You can decide to operate in the Store mode (data collection is stored in a file), in the Threshold mode (thresholds exceeded detection), or both. It is obviously possible to exploit collected data using the `xnmgraph` graphical analysis tool of NetView. This application can also compute some statistics on the data.

At a threshold detection, you can send a trap (every enterpriseId is possible) to the Mid-Level manager or to NetView for AIX. You also can run automatic actions (C program, shell script, simple or complex) that will be executed on the local SIA node.

So, there are two possible levels of automated actions, one is to be implemented *centrally* on NetView (`Options→Event Configuration→Trap Customization` menu), and another *locally* on the SIA node. Obviously, the local automation driven by the Mid-Level manager can also be applied to other nodes it manages. It just takes running remote commands in the shell script. Thus, the function of automated actions is considerably enriched by Systems Monitor.

Lets consider the following example:

```
smMlmThresholdName =  "TokenRingObstruction"
smMlmThresholdState =  disabled

smMlmThresholdLocalRemoteMIBVariable = ".1.3.6.1.4.1.2.6.12.6.1.1.9.ObstructionRate"

smMlmThresholdCondition =  "delta >"
smMlmThresholdValue =  "2000"
smMlmThresholdPollTime =  "15m"

smMlmThresholdTrapDescription =  "Network obstruction rate exceeded on "SubNetwork1""
smMlmThresholdArmEnterprise =  ".1.3.6.1.4.1.2.6.12.5.1"
smMlmThresholdSpecificTrap =  100
smMlmThresholdCommandToExecute =  "/usr/OV/procs/shell1"

smMlmThresholdReArmCondition =  "delta <"
smMlmThresholdReArmValue =  "1000"
```

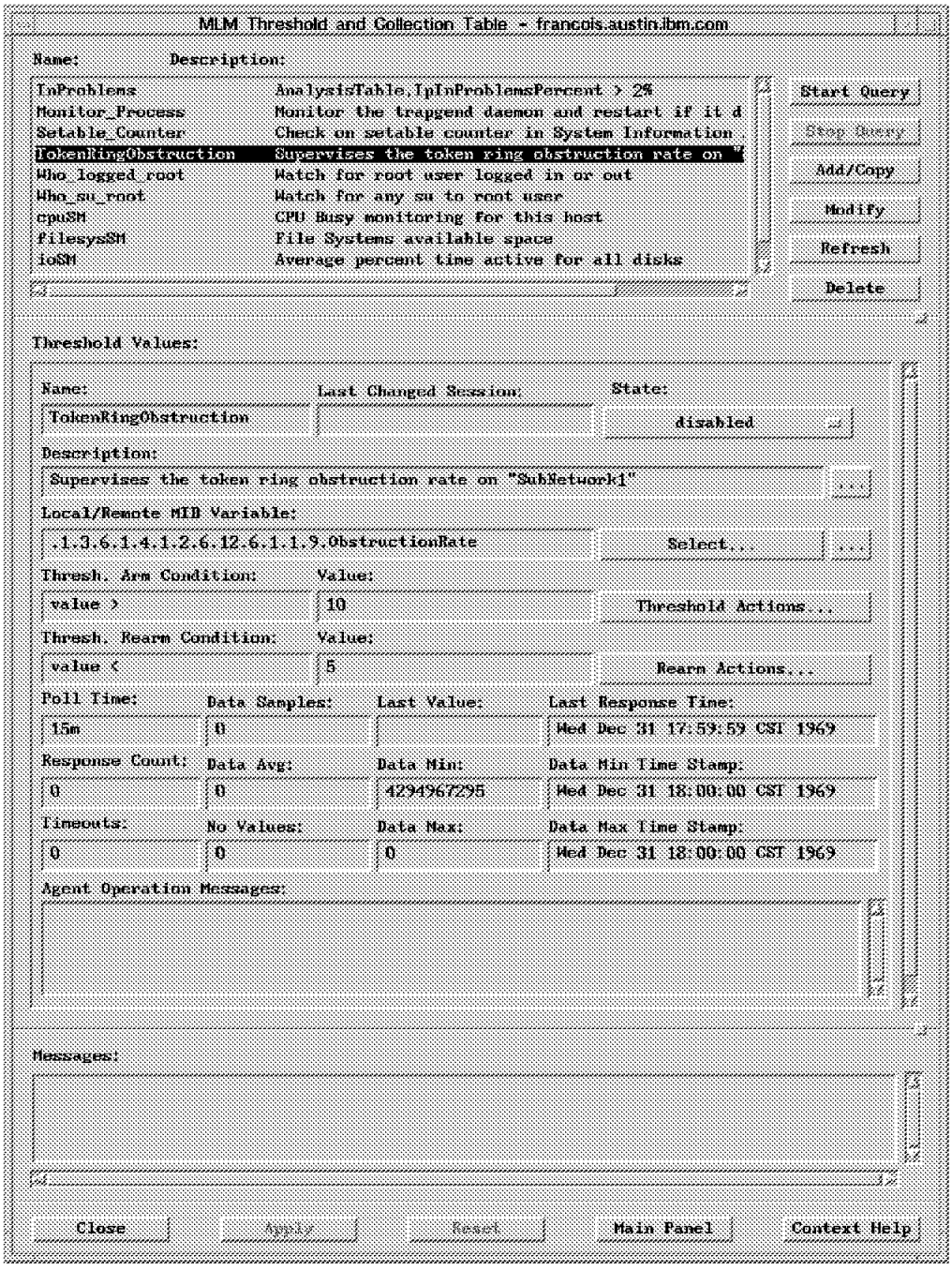

Figure 143. Systems Monitor Threshold and Data Collection Table

In Figure 143, if the smMlmThresholdState field is set to enabledStoreOnly, data will be collected every 15 minutes into the /usr/adm/smv2/collect/midmand.col file, concerning the ObstructionRate variable we created in the MLM Analysis Table.

If the smMlmThresholdState field is set to enabledThresholdOnly, the MLM will, every 15 minutes, look for exceeded thresholds (threshold here has fixed value of 2000).

The enabledThresholdStore mode implements simultaneous data collection and threshold detection.

The threshold value is 2000: the MLM measures the variation (delta) of the variable ObstructionRate. If you want to measure the absolute value, you have to put `value` instead of `delta` in the ThresholdCondition and ReArmCondition fields.
Because the ObstructionRate variable is of type counter, you can afford not specifying `delta` in those fields.
The condition operators are: =, <, <=, >, >=, !, & (AND), | (OR), changes, doesNotChange, exits and doesNotExist.

When threshold is exceeded, the /usr/OV/procs/shell1 shell script is executed locally (local SIA node), and the following trap is sent back to the manager (or managers) configured to be the MLM trap destination in the trap destination table (do not forget to *enable* it):

```
enterpriseId = ".1.3.6.1.4.1.2.6.12.5.1"
generic trap = 6
specific trap = 100
parameters passed in the ThresholdTrapDescription field
```

Anyway, before sending the trap, the MLM Filter Table is checked, to know whether the trap will be blocked, forwarded to NetView, or kept waiting for a certain condition (mode `throttleTraps` in the Filter Table) before forward is launched.
As always, the forwarded trap has to be recognized on NetView. It has to be preliminarily configured in the `Options→Event Configuration→Trap Customization` menu (Event Log Message, Severity, Command for Automatic Action, ...).

If you have defined a rearm condition in the Threshold Table (in our sample, obstruction rate goes under the value 1000), a rearm trap can be sent, as well as another shell script can be executed.

Supervision of Applications

The Threshold Table can be of a great use in applications supervision because it can follow particularly sensitive data in an application (processes, filesystems, shared memory, etc...). Either this data already exists in the standard Systems Monitor MIB, or it is created via the Command/Analysis Table.

You then create an *entire supervision mechanism* with:

1. Sensitive variables polling
2. Exceeded thresholds detection and rearm
3. Trap forwarding to one or more Mid-Level Manager and to one or more NetView
4. Actions automation in local, on other nodes of the network, and/or on NetView

An automation in central on NetView that may be interesting to implement would be to create objects in the NetView topology map representing sensitive data (process chain, filesystems, sharedmemory, ...). Depending on the forwarded trap, NetView will change the object status (the color) using the EUI APIS with

the wteuiap6 package (see section B.1.2.13, “Programming the Graphical Interface: Use of EUI APIs” on page 189).

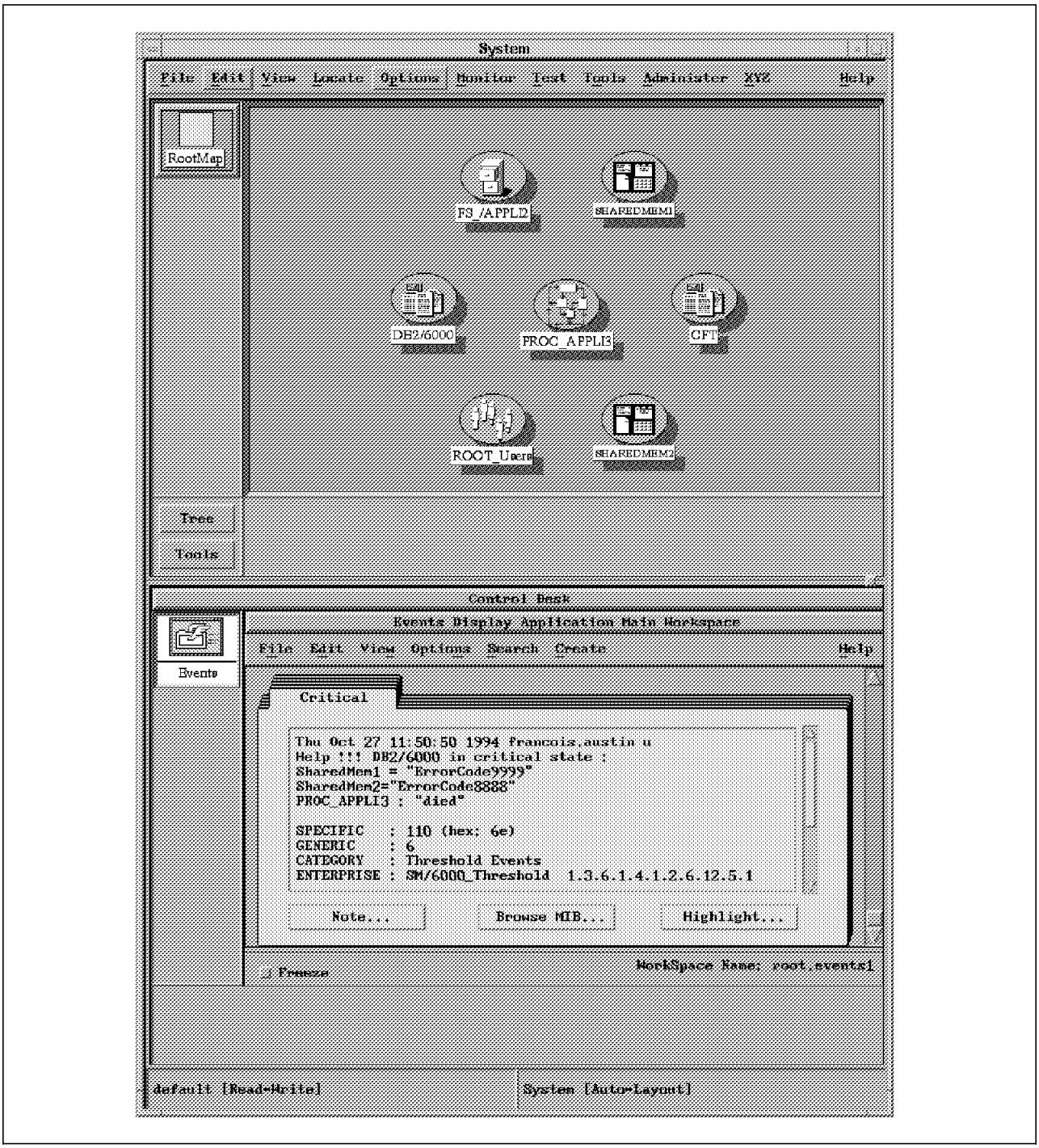

Figure 144. Supervising an Application Using Systems Monitor

GraphicallyFigure 144 shows what we can obtain in that perspective.

B.4.3.7 The MLM Filter Table

The Filter Table defines another essential function in the distributed network management between NetView and the Mid-Level manager. The MLM can be configured to block incoming traps from its managed nodes, forward them to another manager (NetView or another MLM), or wait for a certain condition before forwarding (throttleTraps mode).

A MLM that wants to do filtering will, for example, have the following configuration in its Filter Table:

```
smMlmFilterDefaultAction =  sendTraps
smMlmFilterUdpTrapReception =  enabled
smMlmFilterTcpTrapReception =  enabled

smMlmFilterName =  "ObstructionTrap"
smMlmFilterState =  enabled

smMlmFilterAction =  throttleTraps
smMlmFilterEntryEnterpriseExpression =  ".1.3.6.1.4.1.2.6.12.5.1"
smMlmFilterGenericExpression =  "6"
smMlmFilterSpecificExpression =  "100"

smMlmFilterAgentAddrExpression =  "SubNetwork1"

smMlmFilterActivationDayOfWeek =  "weekdays"
smMlmFilterActivationTime =  "7:00"
smMlmFilterDeactivationTime =  "20:00"

smMlmFilterThrottleType =  sendAfterN
smMlmFilterThrottleArmTrapCount =  4
smMlmFilterThrottleDisarmTimer =  "10m"
smMlmFilterThrottleDisarmTrapCount =  2
smMlmFilterThrottleArmedCommand =  "/usr/OV/procs/shell2"

smMlmFilterTrapDestinations =  "mlm2.austin.ibm.com"
```

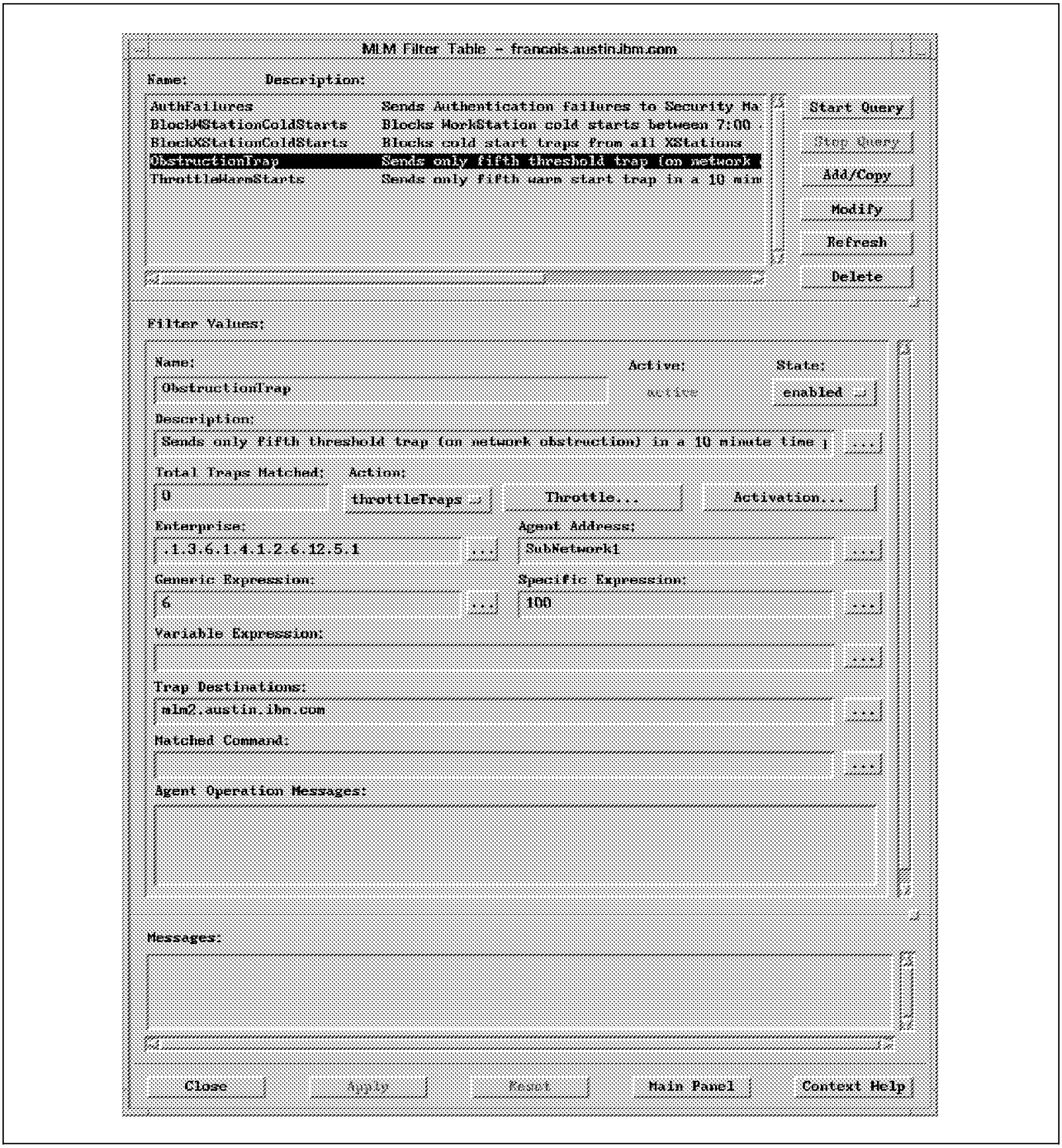

Figure 145. Systems Monitor Filter Table

The filter rule is customized to be active only on weekdays, from 07:00 am to 08:00 pm (see Figure 145). It applies to the threshold trap we defined precedently in the Threshold Table (ObstructionRate entry). Only SIAs belonging to the SubNetwork1 alias are concerned.

The throttleTraps mode is used (the other possible modes are sendTraps and blockTraps). In our case, only the fifth and sixth traps coming in a 10 minute time period (since the first trap) are forwarded to a second MLM, mlm2.austin.ibm.com. The smMlmFilterTrapDestinations field here overtakes on the MLM Trap Destination Table.

You can also automate actions in the Filter Table, at different levels:

1. smMlmFilterMatchedCommand: Command executed when an incoming trap sastisfies to the filter rule
2. smMlmFilterThrottleArmedCommand: Command executed when an incoming trap satisfies to a throttle condition (here, it will be the fifth trap), for example: /usr/OV/procs/shell2
3. smMlmFilterThrottleDisarmedCommand: Command executed when an incoming trap disarms throttling (it will be the case of the sixth trap)

The /usr/adm/smv2/log/midmand.log logfile reports details of all the filtering steps (and also analysis, data collection ... steps). Do not hesitate to look at it, it can give interesting indications, especially in case of configuration problems.

B.4.3.8 The SIA File Monitor Table

With the File Monitor Table, Systems Monitor provides you the ability to monitor files. The table scans a specified file for changes or existence:

- Text strings
- Changes to file characteristics (such as owner, group or permissions)
- Existence of a file
- Changes to the file size

It can issue a command before monitoring and after if the monitored condition is met. A trap can also be sent to the NetView for AIX manager or to a MLM trap filter. There are five possible traps of Enterprise ID .1.3.6.1.4.1.2.6.12 and of generic trap 6. consecutive to file monitoring:

- specific trap 21: String not found
- specific trap 22: File data modified
- specific trap 23: File status changed
- specific trap 24: File does not exist
- specific trap 25: File exists

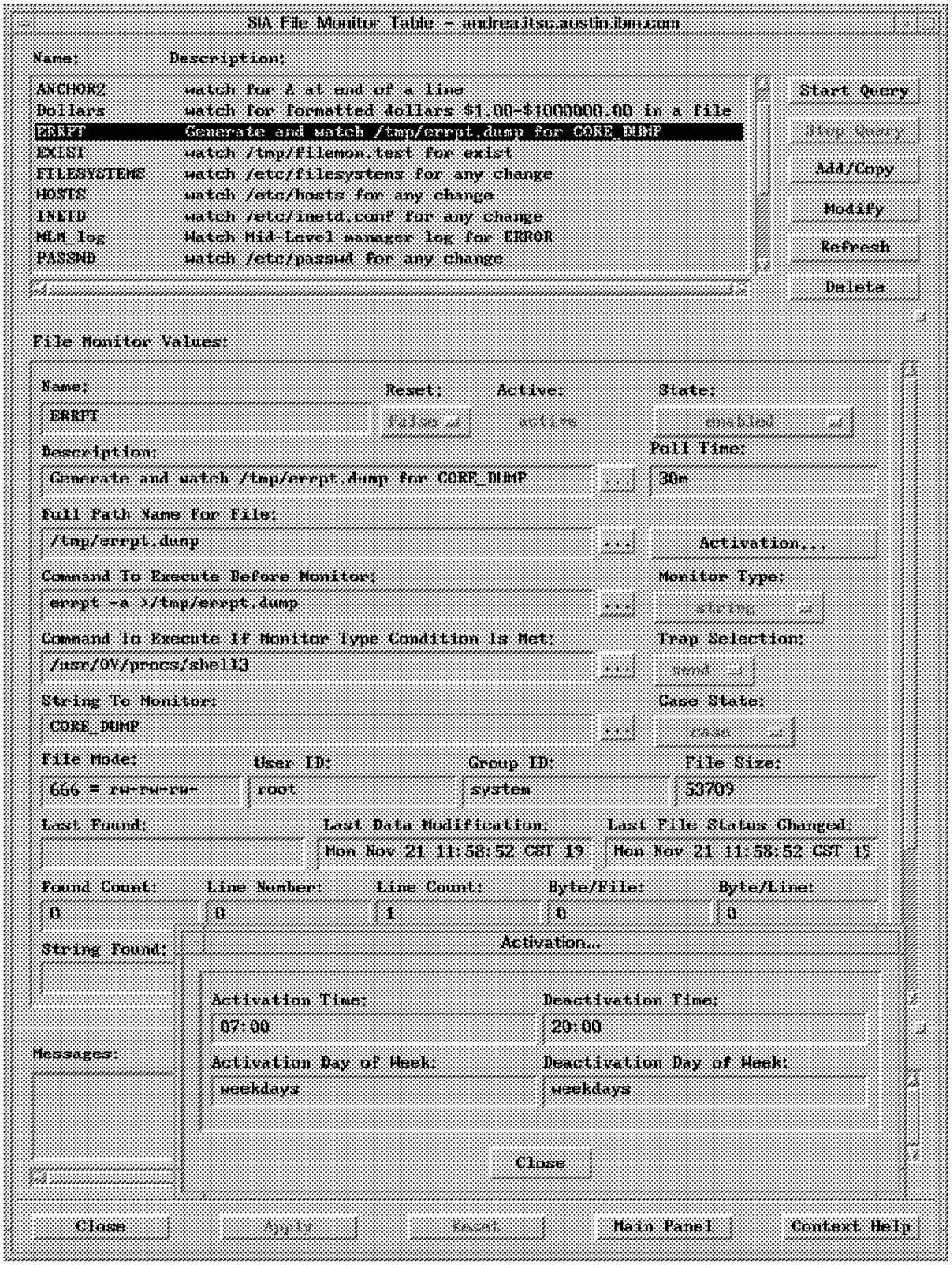

Figure 146. Systems Monitor File Monitor Table

The Systems Monitor file monitoring function is useful, particularly for system files such as those in the /etc directory, to know all attempts to change file characteristics (as read/write permissions or owner). It is also useful to monitor application output files, in order to be notified of errors, return codes or job completion status (see the example given in Figure 146).

B.5 Trouble Ticket/6000 V3R1

This section develops a cookbook for the Trouble Ticket/6000 (TT/6000) V3R1 product.

B.5.1 Installation

Installation is done using SMIT `installp` on a V3.25 AIX system with X11R5. It needs a minimum size of 32MB of memory, as well as 100MB minimum of disk space on /usr. The directory is /usr/lpp/tt6000. PTF's U409865 and U402887 are to be applied to fix a known AIX X-server display bug which causes some character fonts to be partially displayed (edit /usr/lpp/tt6000/README for more details).

The PTF: U433633 is to install on the top of the TT/6000 V3 product to make it operate with NetView for AIX V3. Without this PTF, TT/6000 would be unable to read the new trap format brought by the V3 of NetView, described in the /usr/OV/conf/C/trapd.conf file.

Note: Look out! TT/6000 retrieves a /usr/OV/conf/$LANG/trapd.conf file, so you may have to reassign the LANG variable in your korn shell session.

Implementation of TT/6000 with Ingres** database needs Ingres Release 6.4/01 (rls. US5/02); for Sybase**, you will have to install Sybase SQL Server Release 4.9.1 with a Sybase Open/Client license.

B.5.2 Configuration

After the product installation, there are many necessary configuration steps to process. We give here a summary of the operations, but you should read *Trouble Ticket/6000 V3 User's Guide* to get their details:

1. Setting up Ingres (Ingres has to be already installed before this step), optional
2. Setting up Sybase (Sybase has to be already installed before this step), optional
3. Configuring the server (particularly with `smit TT/6000-1→Configure→Set options for TT/6000`)
4. Setting some optional server environment variables

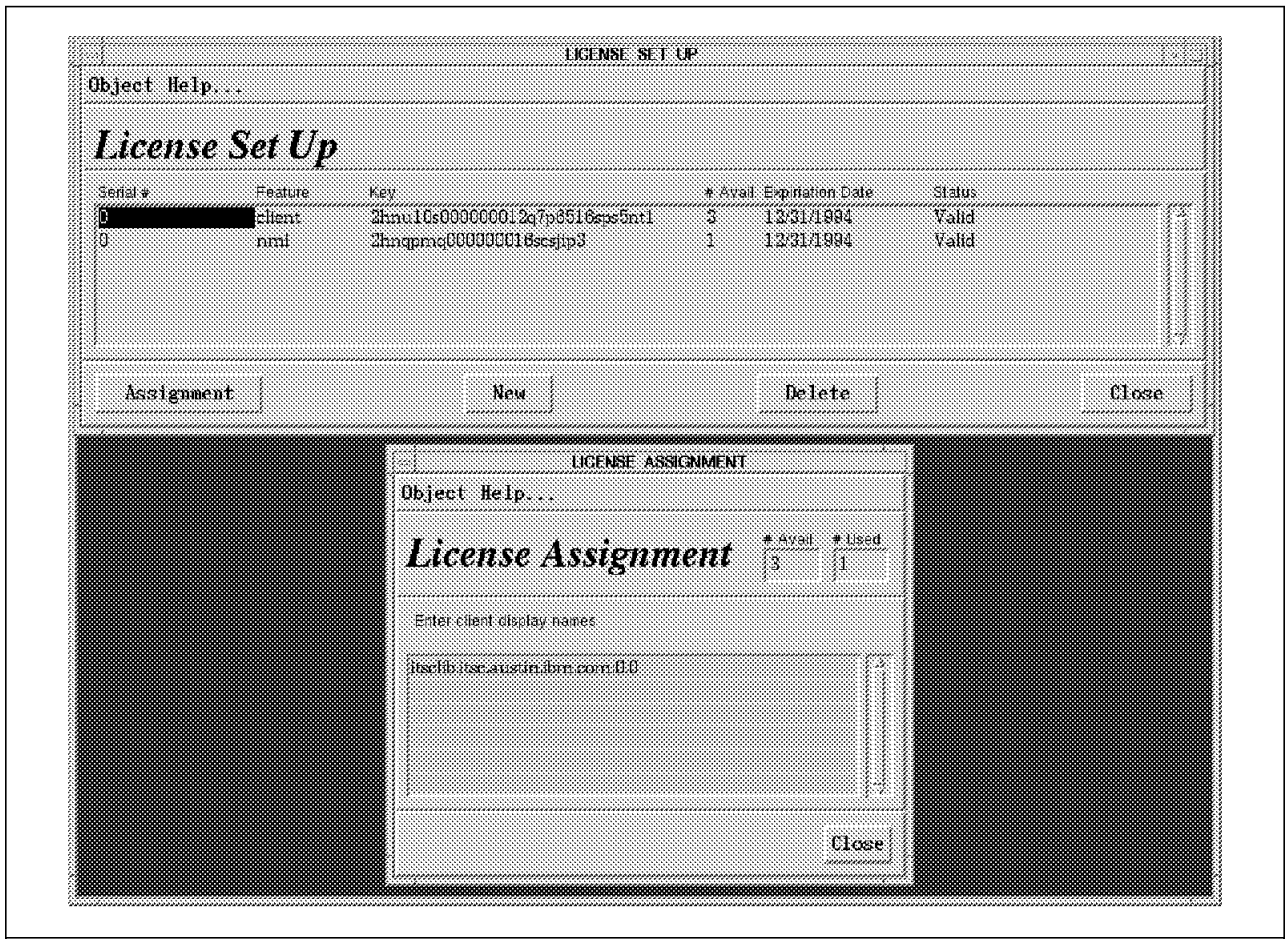

Figure 147. Setting up Licenses on Trouble Ticket/6000

5. Licensing client and nmi (NetView for AIX) licenses. You must license the workstation that holds the TT/6000 server code as a client (see Figure 147). Without that, you can access the Trouble Ticket EUI, but none of its functions.
6. Installing and configuring clients (AIX, DOS/Windows, Sun and HP). In this client-server environment, the TT/6000 server stores the database information, the client stores the EUI information. The client needs to interact with the server only when database information is being shared. They can perform all the non-administrative functions TT/6000 offers. In this configuration, work is offloaded from the server workstation and network traffic is reduced.

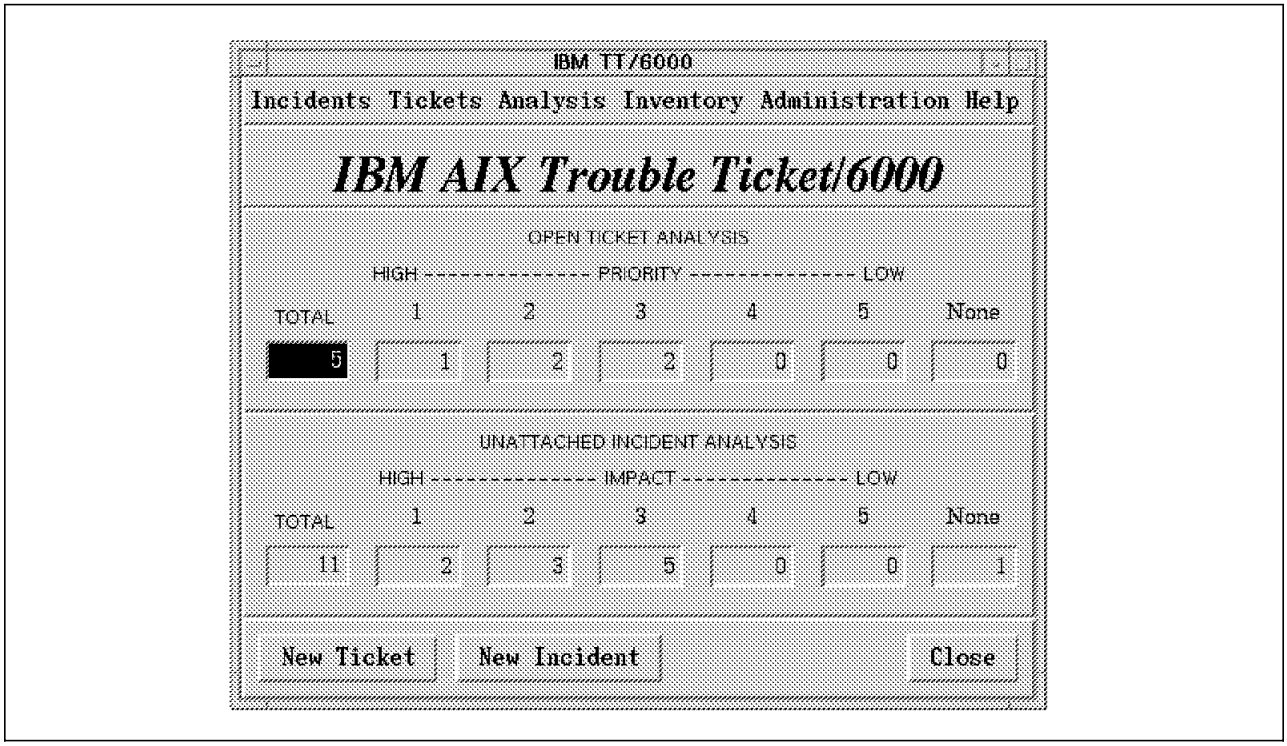

Figure 148. Trouble Ticket/6000 Graphical User Interface

Now, you can run TT/6000 invoking it from:

- The TT/6000 server using `smit TT/6000-1-Control-Run TT/6000`, or `tt6000` from the AIX command line (see Figure 148). Be sure the application daemons are all started. If not, start them using the `nx_init` command.
- NetView menu bar, context menu or tool palette (except for the TT/6000 main menu, you have to select a node first to call all those menus).
- The AIX client, running `tt6000` at the AIX shell prompt.
- The Microsoft Windows client by selecting `TT/6000 Windows Client` from the TT/6000 program group.
- The Sun client, running `tt6000` from the command line.
- The HP client, running `tt6000` from the command line.
- The IBM Xstation, typing `tt6000`.
- An X-terminal emulator, typing `/usr/bin/tt6000`.

B.5.3 Creating an Inventory Base for Trouble Ticketing

TT/6000 works with an inventory database reflecting your network environment to perform trouble ticketing. These are the tables of the relational database that it automatically loads (but you can also populate/modify these tables):

- Resource classes (Workstation, Mainframe, Gateway, Hub, Modem, Novell Server LAN Analyzer, PBX/PABX, ...) corresponding to NetView resource types
- Models (IBM 6611, IBM 8250, DEC station, IBM RS/6000, HP Printer, ...) corresponding to what we can find in /usr/OV/conf/oid-to-type file

- Reference table data (on inventory codes: Contact types (Help Desk, Operator, Technician, ...), ...; on troublet ticket codes: Ticket Status (Open, fixed, closed), ...)
- Network resources (but you have to run the `Administration→Import/Verify Managed Elements` menu to interface with NetView topology/object database to retrieve all discovered nodes)

These are the tables that you have to populate (we advise you to do it in the following order):

- Sites
- Organizations
- Vendors
- Locations
- Contacts

Figure 149. Trouble Ticket/6000 Contacts Table

To create and use trouble tickets more effectively, you should populate *at least* the Contacts (see Figure 149) and Network Resources tables.

The tables are loaded, either using directly the Inventory TT/6000 menu or editing the Contacts.dat, files, so that you then load with the nx_load command:

```
"nx_load -v -f XXX.dat", for the first load
"nx_load -a -v -f XXX.dat", for all following populations
```

The Contacts.dat and other samples are available in the /usr/lpp/tt6000/samples/data directory. Put all the XXX.dat table files that you create in the /usr/lpp/tt6000/data directory.

On the other hand, using the Administration→Import/Verify Managed Elements menu, the table Network Resources is populated with all the network nodes discovered by NetView (TT/6000 particularly loads *MIB-II datas* for all the SNMP nodes of your network).

B.5.4 Automatically collecting and filtering NetView incidents

By setting the Collect NetView/6000 events option to yes in smit TT/6000-1→Configure→Set options for TT/6000, you configure TT/6000 to automatically generate Incident Reports (IR for Incident Report, TT for Trouble Ticket) from NetView for AIX traps. It's also necessary to configure NetView trapd daemon in smit nv6000→Configure→Set options for trapd daemon:, enter Yes to the Create socket connection for V1R1 applications field.

You can visualize IRs in the Incident Report List, called by the Incidents→List menu, or by double clicking on the IR number in the impact dialog boxes of TT/6000 main panel.

It is then the operator's responsibility to open an incident ticket (TT: Trouble Ticket) on the incoming IR (or on many of them, having the same or different content, this is called attaching IRs to a ticket). Why the responsibility? Because an open ticket will then be managed and will live different events, from various automatic notifications, level escalations, to database archiving. An archived ticket cannot be deleted from the database. It is then clear that you have to knowingly choose the incidents you want to manage.

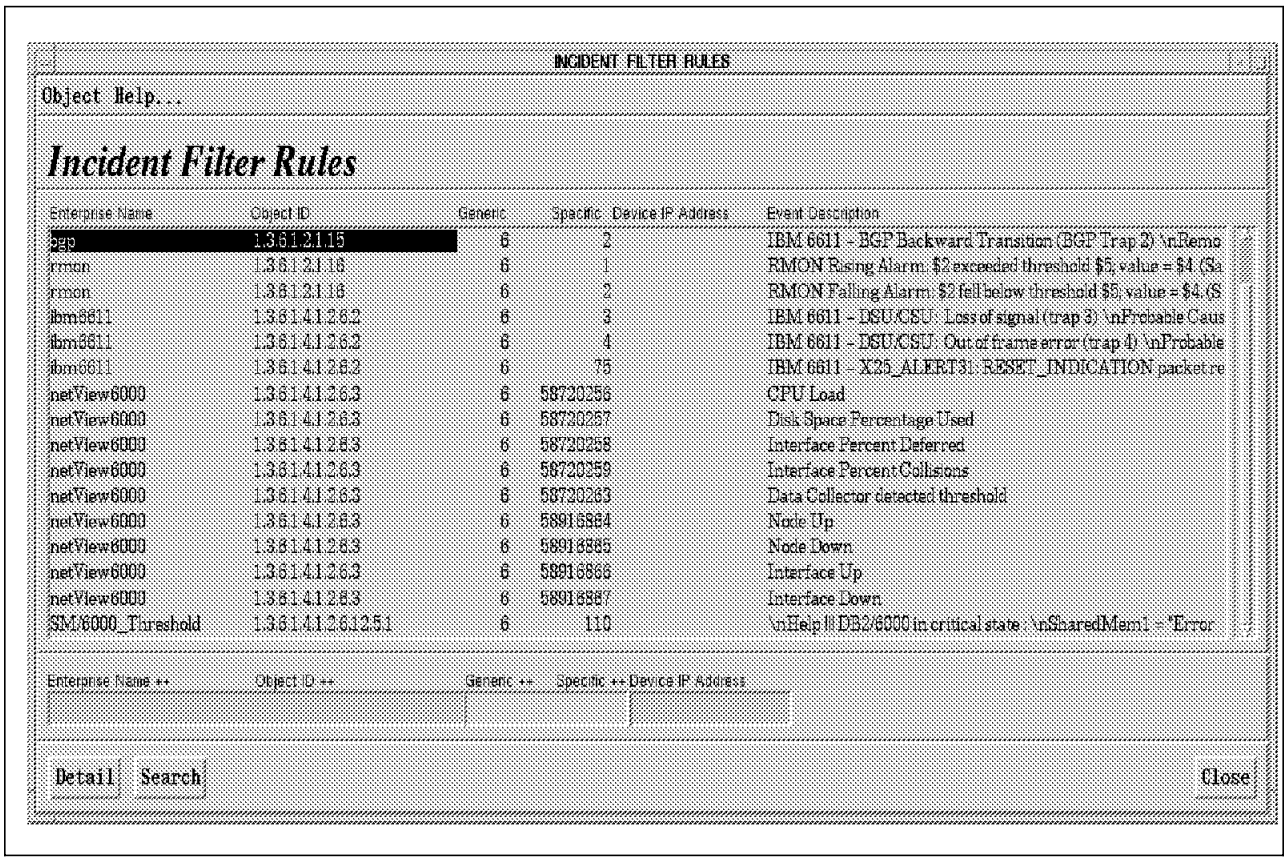

Figure 150. Trouble Ticket/6000 Incident Filter Rules

You can filter which NetView traps will automatically generate IRs by calling the Administration→Incident Filter Rules menu or directly editing the /usr/lpp/tt6000/site/OVfilterspecEVNX.dat file (but we advise you to use TT/6000 user interface). Figure 150 shows you the panel where you define your incident filter rules.

Figure 151. Sample of a Filter Rule

Trouble Ticket/6000 reports NetView for AIX traps along four possible modes:

- ***Report All Events***: Every incoming event from NetView creates an IR
- ***Report Frequent Events***: An event will be reported only after a number of occurences in a time period that you define
- ***Report Long Outages***: Every incoming trap followed by another one that has been configured to be the resetting event, in a time period that you configure, will create an IR (see Figure 151)
- ***Report Long Event Bursts***: The event must have last a long time period to be defined before TT/6000 creates the corresponding IR

Notice that for every reporting mode it is possible to ask TT/6000 to determine, before it creates a new IR, whether the incoming event can be attached to an already existing IR (because it is part of the same problem).

B.5.5 Working on Incident Reports and Trouble Tickets

Incident Reports can be created manually. One reason you might want to create a manual IR is to record the occurrence of a phone call from an affected user who is notifying you of a problem. You want to be sure to record the symptoms the user describes for the technician who is assigned to the problem. You also want to be sure to notify the user when the problem is corrected. Incident reports are also created from NetView automatically collected events along predefined filter rules, as we precedently see.

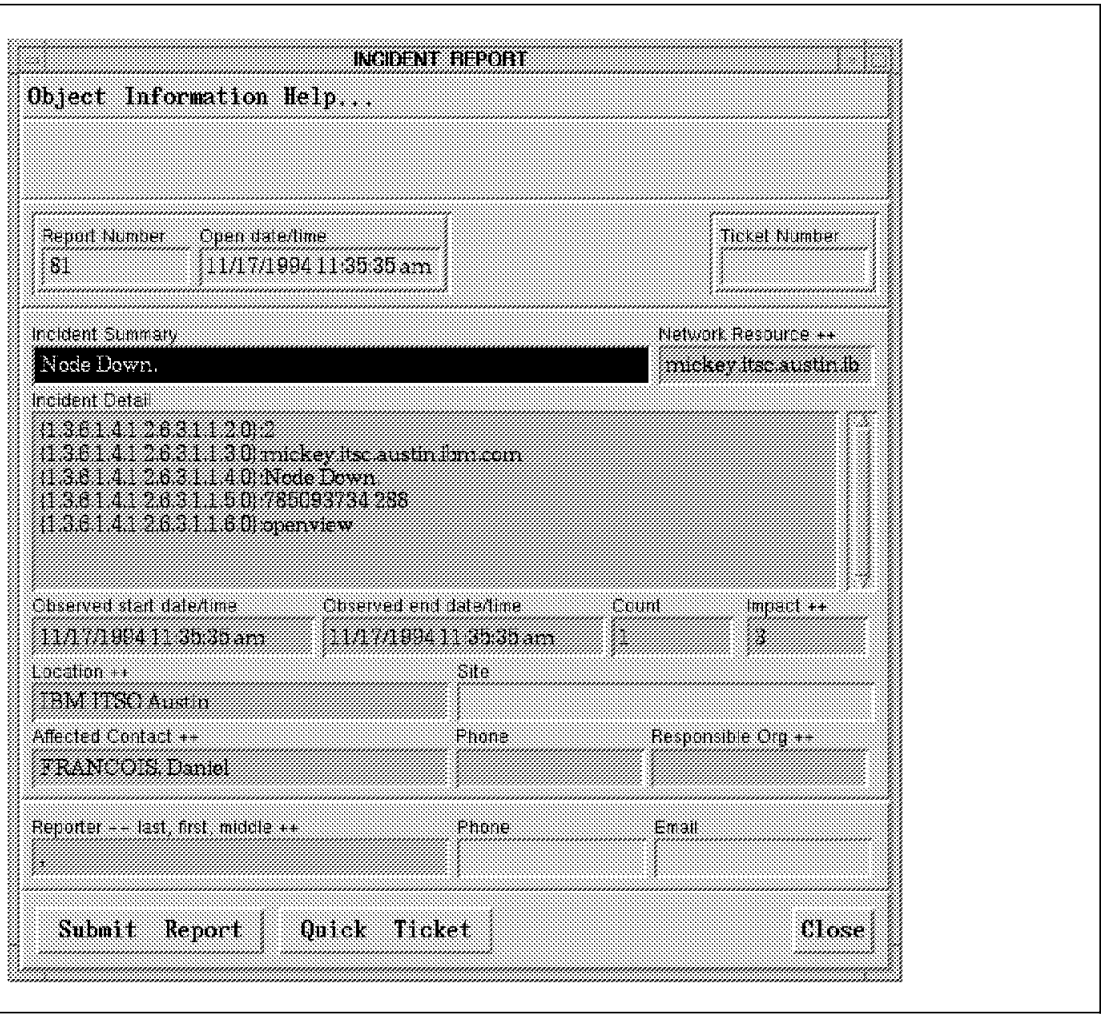

Figure 152. Creating (or Updating) an Incident Report

Activating the Incidents→List→Detail menu on a selected IR, you can fill (or update) the Incident Summary and Incident Detail fields, as well as the Affected Contact, Location, Site and so on inventory fields (see Figure 152). Some of the inventory fields entered here or on the Trouble Ticket dialog box, such as Affected Contact, Assignee, Responsible Organization and Responsible Vendor, will be used for automatic notification, action assignment, and other trouble ticketing operations.

From the Incident Report Detail panel, you can create a trouble ticket by pushing on the Quick Ticket button. You have now to fill up mandatory fields such as Trouble Code, Assignee, Vendor and SLA (Service Level Agreement). You can add your own comments, they will be timestamped and logged with your identification as registrated in the Contact Table. To assist your problem

analysis and diagnostics, you can record possible causes of failure in the Operations→Record Causes menu.

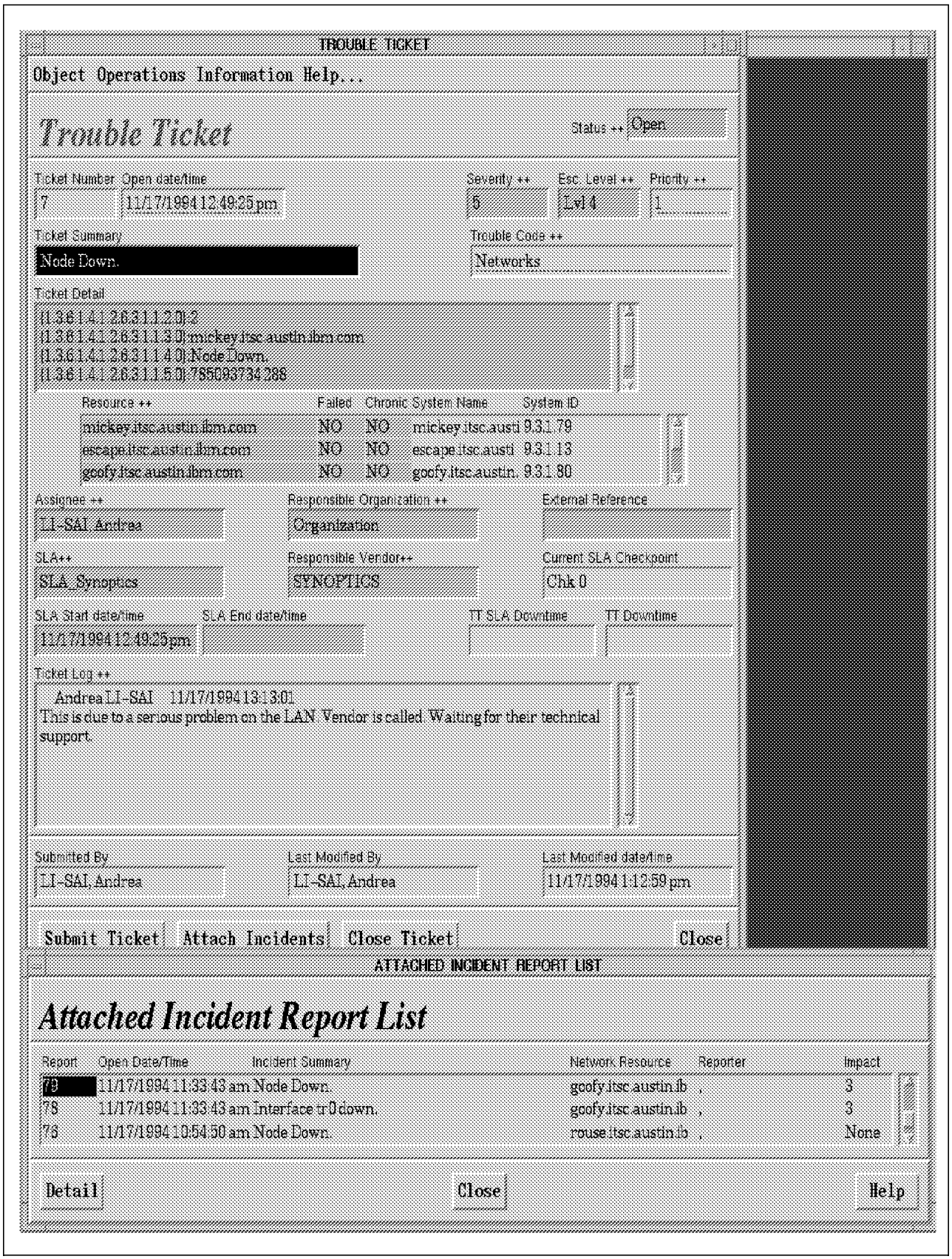

Figure 153. Creating (or Updating) a Trouble Ticket

Incident Reports that you beforehand mark in the Incident Report List panel can be attached to the ticket using the Tickets→List→Detail→Attach Incidents menu. Figure 153 shows a trouble ticket with a list of attached incidents: attaching those incidents filled the Resource field with all the network nodes originator of the IRs. The attached incidents will disappear from the TT/6000 default IR List, as the research criteria which gives the IR List is Attached=NO.

It may be convenient to manipulate the IR and the TT List using the Incidents or Tickets→Search Tool menu : displaying a list responding to the search criteria, you can, for example, delete them in only one time.

B.5.6 Using Action Plans

Often when resolving problems, there are several individuals or groups involved whose activities that must be coordinated. This can be done using Action Plans.

An action is an activity that is assigned to a person in order to correct a network management problem. Each action includes a description, an assignee, a scheduled date and time, and an action log to document the results or process in completing the action.

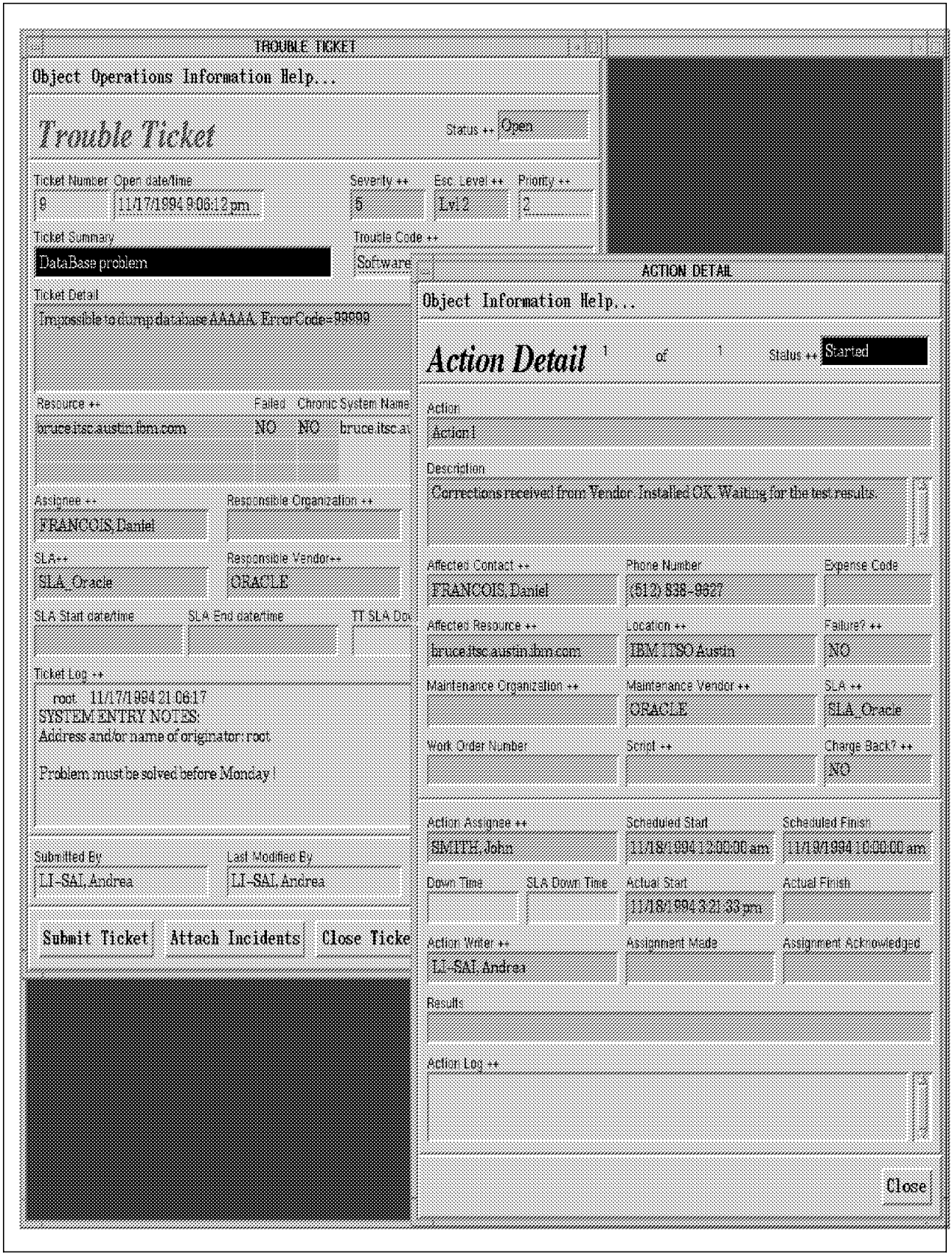

Figure 154. Creating (or Updating) an Action Plan

After you add an action item to a trouble ticket (Operations→Define Actions on the TT detail dialog box), you can configure TT/6000 to automatically notify the action assignees (see Figure 154). Assignees then carry out the corrective actions indicated by the action description, record the results of their actions, and indicate that the activity is completed.

TT/6000 gives you the possibility to load *Action Templates* while defining a new action plan. Effectively in the process of setting up actions for a trouble ticket you will find situations that a very similar set of actions is performed frequently. Grouping these actions into an action template, you will reduce the amount of time required to set up action plans.

How do you create an Action template? After having built the action plan (as in Figure 154 on page 244), select the Object→Save Template menu on the Action List dialog box. To reload it for use in an other action plan, activate the Object→Load Template menu from the same Action List dialog box.

Trouble Ticket/6000 also provides the possibility to enter *Action Delays* on a particular action plan from the Action List dialog box (select the Information→Action Delays menu or the Delays mouse button). In action plans, there are times an action cannot be completed because of a dependency of another action (like waiting for equipment, for redesign or a national holiday). This is often the case when vendors perform an action on a contract basis. In computing their response time performance, it may not be appropriate to include delays incurred because of dependencies outside their control.

B.5.7 Using Service Level Agreements

Service Level Agreements (SLA) identify the arrangements that are made to maintain the operation of various network resources and resource classes. These agreements can be made externally with equipment or maintenance vendors, or internally with maintenance organizations. Action item SLAs are focused on device repair management. For example, on the Trouble Ticket dialog box in Figure 153 on page 242, there are three interesting fields related to the SLA:

- ***SLA Start date/time***: This field is initially populated with the current time when the TT is submited.
- ***SLA End date/time***: This field is initially populated with the current time when the TT status is changed to fixed or closed.
- ***TT SLA Downtime***: Elapsed time calculated when applying the SLA work hours when the TT status is changed to fixed. This is to make the difference with the TT Downtime which is calculated since the open date and time when the TT status is changed to fixed.

For Action Plans, SLAs are naturally used and the Action Detail panel in Figure 154 on page 244, for example, also makes the difference between the SLA Down Time (which is the length of time the action took to be completed, within the SLA work hours and taking into consideration possible action delays) and the Down Time (elapsed time the action took to be completed).

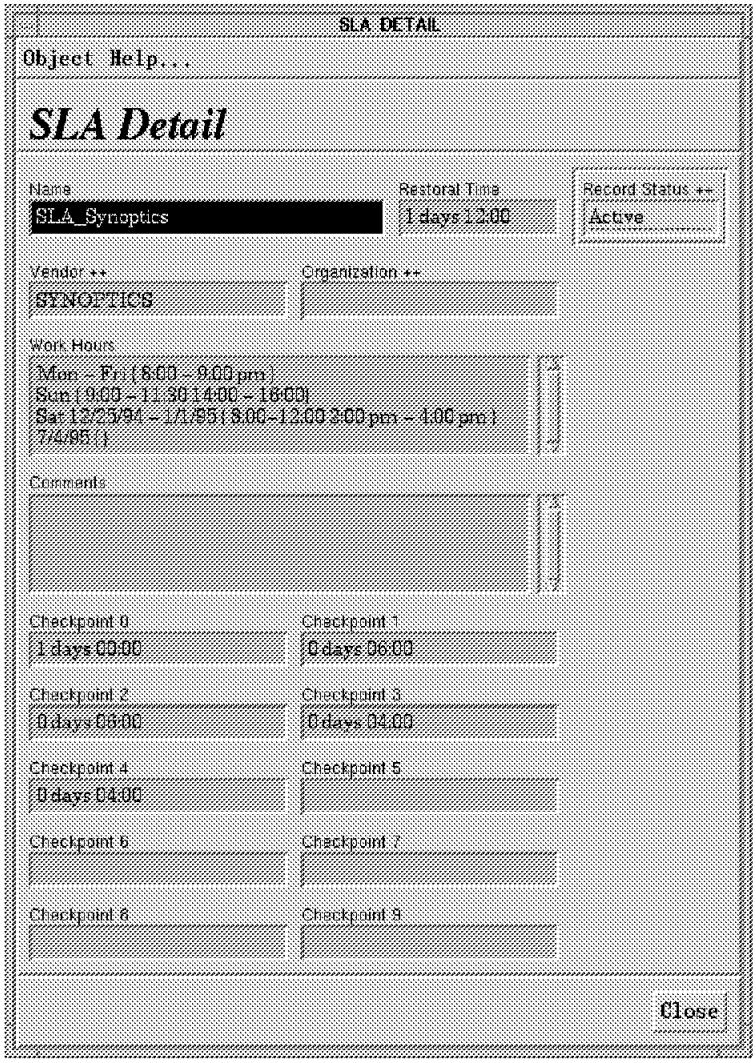

Figure 155. Creating a Service Level Agreement (SLA) Record

Figure 155 shows an example of a SLA record. You notably configure on this panel the Work Hours to be taken into account when analyzing the vendor response time on action plans (activate the Analysis→Reports→Vendor Response Time). You also define the Restoral Time, which is the maximum time you expect for the resource restoration to service. Ten SLA Checkpoints may be set up to indicate the time until the next SLA checkpoint is triggered. These checkpoints are used to track the progress of an action or a ticket. When the checkpoint times occur, TT/6000 automatically sends notifications to the people (managers, technicians) you defined in the Notification Rules panel (developped in the section B.5.10, “Defining Notification rules” on page 248).
In the trouble ticket detail panel, the Current SLA Checkpoint field tells how far the restoration organization is solving the incident.

B.5.8 Trouble Ticketing Using Electronic Mail

You can send electronic mail to your TT/6000 server to submit (create), update or query an Incident Report or a Trouble Ticket. This mail interface may also be used to initiate or update an action. Your local e-mail system interface must be compatible with the BSD Sendmail function, and you must have set up Enable Email and Command Line Entries to YES in the Administration→System Defaults menu. The e-mail userID to use is the one you assigned in the smit TT/6000-1→Configure→Set Options for TT/6000→Email user name field.
A particular mail Subject has to be given:

- ***tt*** or ***ttq***: respectively to submit and to query a trouble ticket
- ***ir*** or ***irq***: respectively to submit and to query an incident report
- ***act*** to create or update an action item

These mail Subjects are *not* case-sensitive.

Here is an example of a trouble ticket creation using Email:

```
mail -s tt tt6k_user@tt_server </usr/lpp/tt6000/actions/mail.tt
```

In the mail.tt file, you give value to fields like Assignee, Description, Detail, Originator, TicketNum, Priority and Trouble Code (Priority and TroubleCode are required fields).

```
Assignee = FRANCOIS,Daniel
Description = "DataBase problem"
Detail = "Impossible to dump database AAAAA. ErrorCode=99999"
Log = "Problem must be solved before Monday !"
Severity = 5
Priority = 2
TroubleCode = Software
```

B.5.9 Trouble Ticketing Using the Command Line Interface

Trouble Ticket/6000 provides a command line interface named cmd_in that enables you to submit and query incident reports and trouble tickets, and to create or update an action item. The command line interface is interesting especially for batch processing multiple requests, which makes cmd_in very powerful.

An example of using the command is:

```
cmd_in /usr/lpp/tt6000/actions/cmd.ir
```

Beforehand, you should have written /usr/lpp/tt6000/bin in the PATH variable of your .profile file. You must also do a export NX_ROOT=/usr/lpp/tt6000 in the same .profile file. For example, to submit a new incident report, you put in the /usr/lpp/tt6000/actions/cmd.ir file the order Incident_Report_Submission with the Client (affected contact from the Contacts Table), ClientPhone, Count, Detail, Impact, Location, Site, Reporter, Resource, Summary and Ticket fields. Summary is a required field.

```
Incident_Report_Submission
{
Summary = "Token Ring Traffic Problems"
Detail = "High response time. Sometimes, unable to communicate"
Impact = 2
Resource = "goofy.itsc.austin.ibm.com"
Count = 3
Ticket = 7
Client = SMITH,John
Reporter = LI-SAI,Andrea
}
```

The other operations using cmd_in are accessible with the function keywords:

- ***Incident_Report_Query***: To request the detail of an incident report
- ***Trouble_Ticket_Submission***: To submit a trouble ticket
- ***Trouble_Ticket_Query***: To request the detail of a trouble ticket
- ***Action_Submission***: To create or update an action item

B.5.10 Defining Notification rules

When an event that has been configured to be significant for trouble ticketing occurs, TT/6000 sets up a notification mechanism which will decide whom should be contacted: a specific contact chosen from the Contact Table, the recipient ID taken from the System Defaults panel (Default Notification Contact field), the contact trouble ticket is assigned to, the responsible organization, the TT originator, a maximum of three potential affected contacts, or the responsible vendor). The notification rule also specifies at which significance of the event the notification mechanism will be launched; significance means Incident Impact for an IR, Ticket Priority and Escalation Level for a TT.

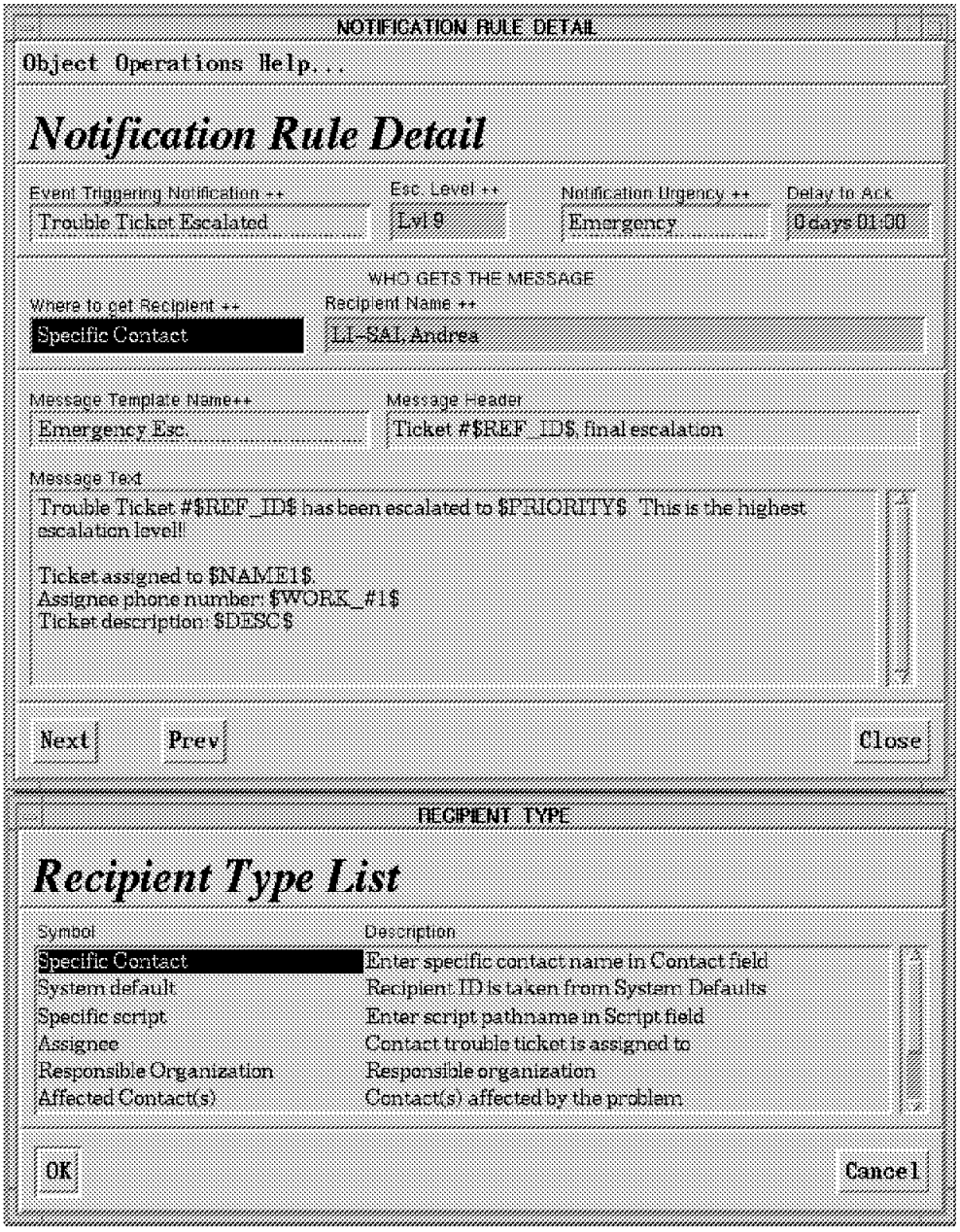

Figure 156. A Trouble Ticket/6000 Notification Rule

Each notification uses a Message Template which includes fields that can be expanded and populated by the triggering notification event message, and forms the message to send to the recipient. You have to tell in the Notification Rule Detail panel (see Figure 156) what the Notification Urgency is. Depending on this urgency, a notification method will be used as e-mail, your personal shell script or a message to a paging system. You may specify if the notification needs an acknowledgement within a certain delay or is just a FYI (For Your Information) message. Here is an example of a notification method which sends a message to a printer located at the Help Desk office:

```
echo "\n
TO:         Help Desk Coordinator
\n
SUBJECT:    TT6000 Notification
\n
A notification of --- $NX_NTF_SUMMARY ---\n
has been sent to userid $NX_NTF_USERID.\n
If this notification has not been acknowledged\n
within 24 hours, please call $NX_NTF_BEEPER_PHONE.\n
\n
When the recipient responds, convey the following message:\n
$NX_NTF_MESSAGE" > /tmp/fic$$
mail -s "TT/6000 message" root < /tmp/fic$$
#rm -f /tmp/fic$$
#$NX_NTF_MESSAGE" | lp -d printer1
```

The printed notification result would be:

```
TO:         Help Desk Coordinator

SUBJECT:    TT6000 Notification

A notification of --- Ticket #7, final escalation ---
has been sent to userid root.
If this notification has not been acknowledged
within 24 hours, please call (512) 838-9999.

When the recipient responds, convey the following message:

Trouble Ticket #7 has been escalated to Lvl 9.
This is the highest escalation level!!

Ticket assigned to LI-SAI, Andrea  .
Assignee phone number: (512) 838-9647
Ticket description: Node Down.
```

We previously talked about a *pager notification* possibility. NetView for AIX provides a sample in the /usr/OV/prg_samples/nnm_examples/beeper directory. The shell script to run is beep_95lx and the way to use it is:

```
USAGE:
      beep_95lx phone_number pager_number [code]

EXAMPLE:
      beep_95lx 9,1-800-555-7243 123456 2001
```

The beep_95lx program executes the cu command accessing a modem connected to the NetView node. The modem calls a paging system, which then sends a callback number to a pager. The callback number is associated to a user-defined message and may so represent an individual alarm.

B.5.11 Defining Escalation Rules

Depending on its priority, a trouble ticket will be automatically escalated from a lower to a higher escalation level as time passes since the TT is created and it is not fixed nor closed yet. Every priority defines the timing of automatic level escalation.

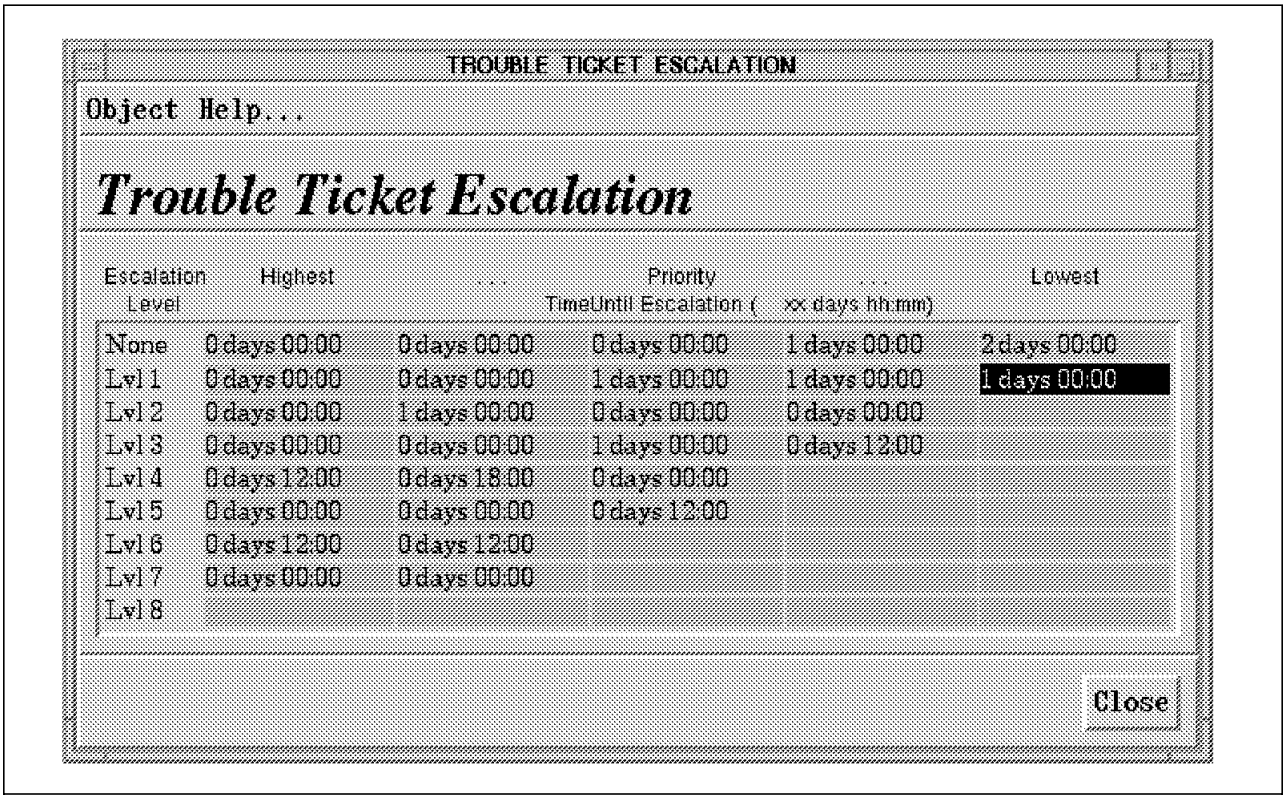

Figure 157. Trouble Ticket/6000 Escalation Rules

For example, lets define from the Administration→Escalation Rules menu the following matrix (see Figure 157):

- Highest priority trouble tickets start at escalation level 4. After 12 hours, if the ticket is still in the Open state, it is escalated to level 6. After 12 hours, it reaches the highest escalation level (level 9). The notification rule we defined in Figure 156 on page 249 will then be applied.
- For tickets of medium priority (priority=3), the escalation level starts at level 1. After one day, level 3 is reached, level 5 after another day. The maximum escalation level here is level 6 after 2.5 days in total the trouble ticket is created.

Features as using Analysis Tools and Reports, customizing Security and redefining Flexible Dialog Boxes are not covered in this cookbook. See the *AIX Trouble Ticket/6000 Version 3 User's Guide* for more information.

Index

© Copyright IBM Corp. 1993,1994

M

N

O

P

R

S

T

U

V

W

ITSO Technical Bulletin Evaluation RED000

Managing One or More
AIX Systems - Overview

Publication No. GG24-4160-01

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 948, Building 821
Internal Zip 2834
11400 BURNET ROAD
AUSTIN TX
USA 78758-3493

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold Along Line

Printed in U.S.A.