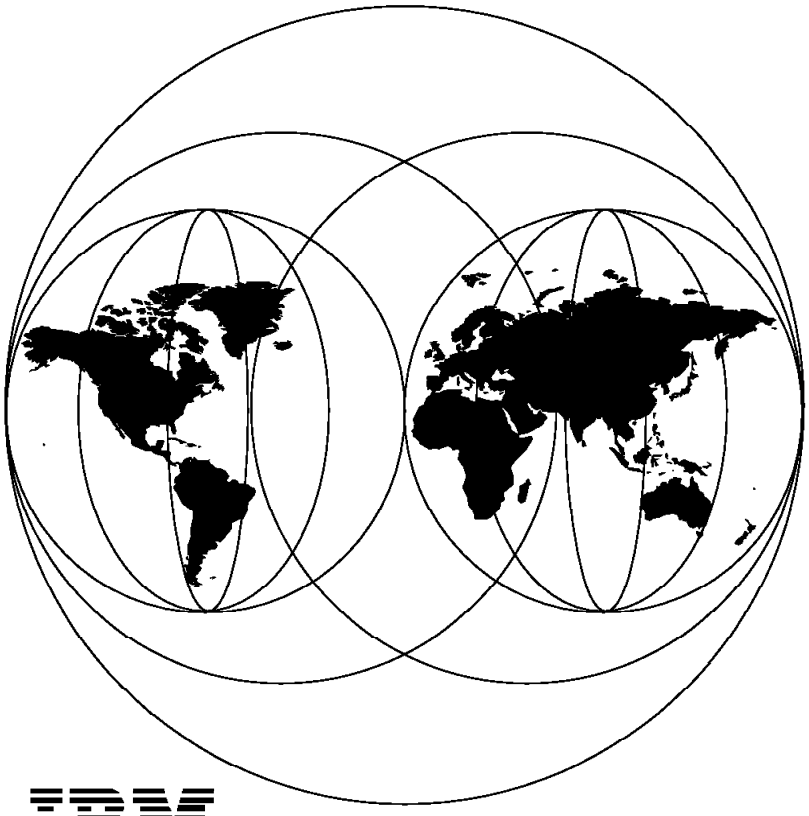
SG24-4819-00

Setting Up a TME 3.0 NT Environment

September 1996



IBM

International Technical Support Organization
Raleigh Center

International Technical Support Organization

SG24-4819-00

Setting Up a TME 3.0 NT Environment

September 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix B, “Special Notices” on page 343.

First Edition (September 1996)

This edition applies to TME 3.0 NT for use with the Windows NT Operating System.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 678
P.O. Box 12195
Research Triangle Park, NC 27709-2195

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.

Contents

Preface

This redbook is unique in its detailed coverage of the Tivoli Management Environment on the NT platform. It provides information about how to install, use and integrate systems management products on the NT Server V3.51 operating system with TME 3.0 NT. In addition, it provides information on built-in tools that come with NT Server V3.51 as well as the NT Resource Kit.

This redbook was written for systems managers who work on the NT platform and wish to know more about TME 3.0 NT and other NT systems management functions.

Several practical examples are presented to demonstrate the integration of the NT operating system with TME 3.0 NT and the NT Resource Kit. Some knowledge of NT Server and TME 3.0 on any platform is assumed.

How This Redbook Is Organized

The redbook is organized as follows:

- Chapter 1, “Introduction”

  This chapter provides an overview of TME 3.0 and describes the hardward and software environment that was used for this book.

- Chapter 2, “TMR and Client Installation”

  This chapter describes how to install the Tivoli Management Region (TMR) and how to install its clients.

- Chapter 3, “Connecting Tivoli Management Regions - TMRs”

  This chapter describes the various ways that you can connect TMRs. It also shows how to check the status of those connections.

- Chapter 4, “Microsoft NT Server Environment”

  This chapter provides details on the systems management functions that are provided with NT Server V3.51.

- Chapter 5, “Administration with Tivoli TME 3.0 for Windows NT 3.51”

  This chapter describes how to administer the Tivoli Management Environment (TME) on the NT platform.

- Chapter 6, “Policy Management”

  This chapter describes policy management and shows examples of the end user interface for the policy manager.

- Chapter 7, “Configuration Management”

  This chapter provides examples of setting up profiles and using Tivoli Sentry.

- Chapter 8, “Working with Tasks, Jobs and the Scheduler”

  This chapter describes how to use the task library and the job library function that is supplied the the TME.

- Chapter 9, “Integration”

  This chapter provides examples of how to integrate TME 10 NetFinity and NetView for AIX with TME 3.0 NT.

- Appendix A, “TME Configuration Files”

  This appendix provides samples of the configuration files that were used in this project on the NT and OS/2 platforms.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the Systems Management and Networking ITSO Center, Raleigh.

Barry D. Nusbaum is a Senior International Technical support representative at the Systems Management and Networking ITSO Center, Raleigh. He writes extensively and teaches IBM classes worldwide on all areas of TME systems management on the NT and AIX platform. Before joining the ITSO 4 years ago, Barry D. Nusbaum worked in Professional Services in the United States as the National Communications Specialist.

Jason Forsyth is a systems specialist in the UK. He has several years of experience in systems management.

Patrick Wenz is a systems specialist in Germany. He has several years of experience in systems management.

Thanks to the following people for their invaluable contributions to this project:

Rob Macgregor
Systems Management and Networking ITSO Center, Raleigh

David Young, Mike Testa and Karl Gottschalk
IBM RTP

Tom Bishop, Mike McNally
IBM/Tivoli Austin

Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

Your comments are important to us!

Chapter 1. Introduction

This chapter provides an overview of TME 3.0 (Tivoli Management Environment) on the Windows NT 3.51 platform as well as shows the software and hardware configurations that were used for this project.

1.1 TME 3.0 Overview

Tivoli TME 3.0 is based on an object-oriented framework. This framework allows you to solve the problems of managing distributed applications and systems. Tivoli was founded in 1989. The goal of the company was (and still is today) to develop distributed system management applications, based on an object-oriented framework.

1.1.1 System Management

Before we can start talking about Tivoli TME 3.0 itself, there are some basics that you should know about system management. System management is based on a fundamental of disciplines and rules to manage client/server environments and the underlying network. The goal of system management is to optimize the response times on the network, minimize outages and optimize data handling, especially the storing and updating of configuration data, to interact in a short time. Because of the variety of the requirements a heterogeneous network defines it is not so easy to define the contents of system management. To make this easier IBM, as well as other companies, has developed a framework. The disciplines are defined in the SystemView framework from IBM:

1. Change management
2. Problem management
3. Configuration management
4. Business management
5. Operations management
6. Performance management

The contents of the six disciplines are not discussed here, because part of the merger of IBM and Tivoli was the creation of a new framework for TME 10. This new framework fits together in what today is SystemView and Tivoli TME 3.0. The disciplines are shown in Figure 1 on page 2.

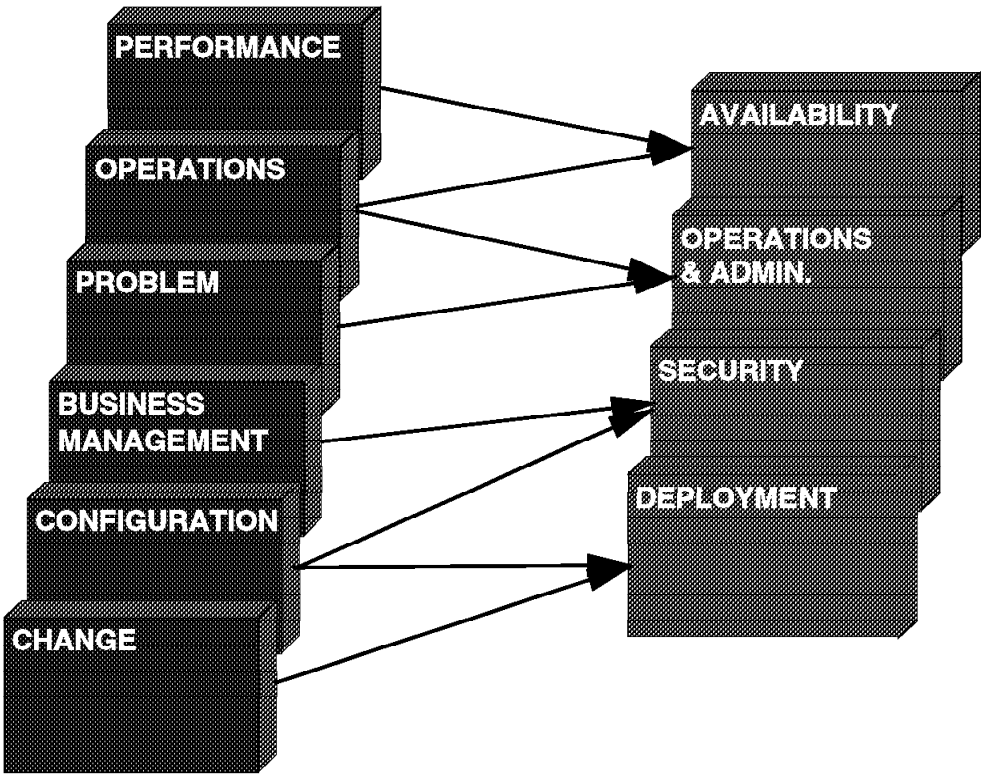


Figure 1. Tivoli/SystemView Merger

The functions of the SystemView disciplines and the Tivoli management processes relate together the following way:

Deployment -> Change, Configuration

Availability -> Performance, Operations

Security -> Business, Configuration

Operations and Administration -> Operations, Problem

The new disciplines are:

- Deployment management

  This discipline includes the following functions:

  – Software distribution (core)

  – Inventory management (core)

  – Asset management and discovery agents (extended)

  Deployment management focuses on the automation of enterprise-wide configuration and change management activities for all of the frequently changing components.

- Security management

  Security management is partitioned into:

  – User administration (core)

  – Third-party directory/name services (extended)

  – Audit analysis (extended)

These disciplines ensure that users have access to the applications and the data that they need to do their job, and ensures the security of corporate information assets.

- Availability management

  Availability management includes the following parts:

  - Performance management (core)
  - Performance analysis/reports (extended)
  - Event automation servers (core)
  - Mid-level manager (core)
  - Network management (core)

  This discipline includes functions that are gathering, collecting and routing information regarding the status of the network.

- Operations and administration

  This discipline is also divided in a few sub disciplines (or functions):

  - Job scheduling (core)
  - Remote control (core)
  - Help desk (extended)
  - Backup and restore (extended)
  - Output management (extended)

  Operations and administration takes care of the automation of activities that ensure the integrity and reliability of the network computing environments.

The products listed above are existing products that integrate with SystemView AIX. As a result of the merger with Tivoli, the SystemView and Tivoli products will get integrated together into a more open platform. Some of the old IBM products may be discontinued due to duplicates of function. A road map has been provided at http://www.tivoli.com.

1.1.2 Tivoli Framework

Tivoli is based on a single architecture. The framework that fits SystemView and Tivoli together is called Tivoli Management Framework (TMF). TME (Tivoli Management Environment) is the platform for the management of heterogeneous networks. The whole framework is object-oriented. The TMF serves as the core for TME. TMF includes services for transactions, security, events and alerts, configuration and automation. In the TME architecture, a service is encapsulated and exposed using integration APIs as a common service if it is of general use to other management applications. The TME architecture includes three framework components:

1. Management Server - This server includes the TMF (which includes the services) and a set of management applications.
2. Management Desktop - This console provides administrators and operators with task-oriented management views.
3. Managed Client - The client is the target of the management operations.

To be scalable and flexible the architecture includes the option to partition the physical environment into multiple logical management regions. These regions

are called TMR (Tivoli Managed Regions). The regions can connect together over a one-way or two-way connection.

The TMR connection options are:

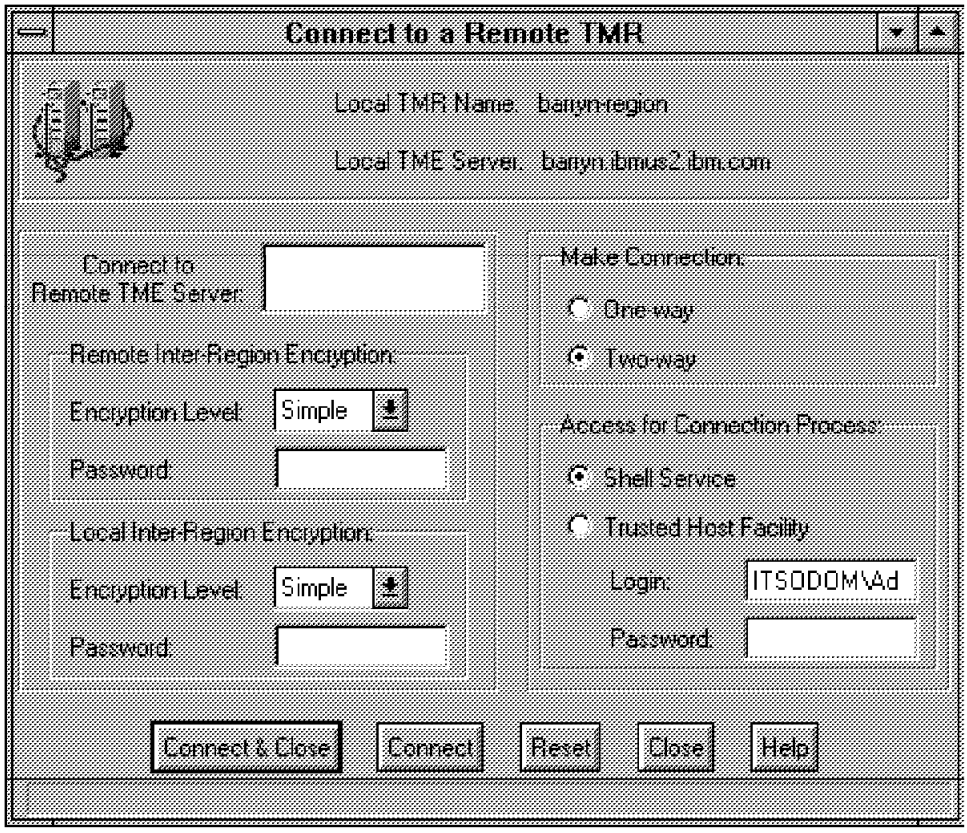


Figure 2. Connecting TMRs

Between two connecting TMRs there is the possibility to update the resources (names and object identifiers) of each region periodically. Therefore, Tivoli integrated a concept called management by subscription. It consists of three components:

- A configuration profile to specify what is to be done
- A profile manager that is a collection of the profiles
- Profile endpoints that subscribe to the profile manager

Another part of TMF is the Policy Regions. They build containers for managed resources. Because of the hierarchical way the regions organize the resources, you can reflect your companies structure in building different Policy Regions which can include subregions.

1.1.3 The Core Environment

The core Tivoli environment consists of the TMF and a set of core applications. Core management applications are organized into the four management processes (deployment, availability, security and operations and administration). This environment resides on every Tivoli management server. There is also a critical interface to third-party products for extended functions.

1.1.4 The Extended Environment

The extended applications are integrated into the four primary core application areas, but they can also perform a specific management function that varies across the parts of a company. Extended applications use supported integration APIs to integrate with TMF or core applications. To integrate the applications there are four tools:

- ADE - Advanced Development Environment
- AEF - Application Extension Facility
- EIF - Event Integration Facility
- AMS - Application Management Specification

The following two figures give an overview of the framework.

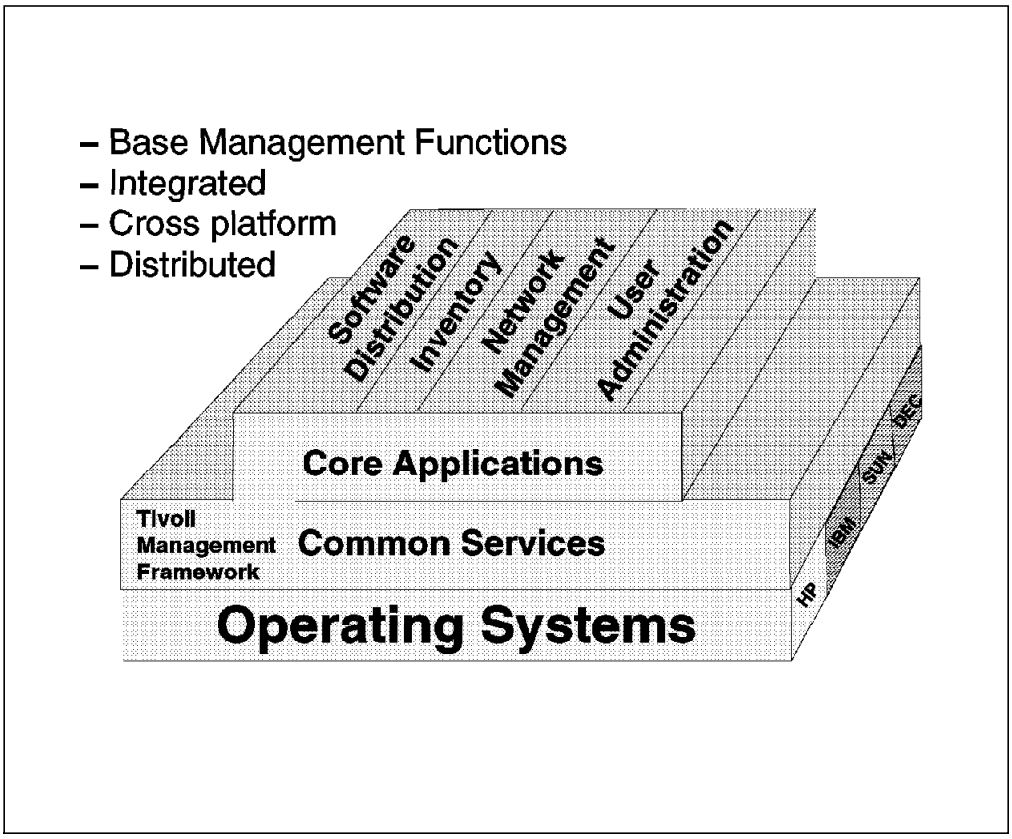


Figure 3. Framework I

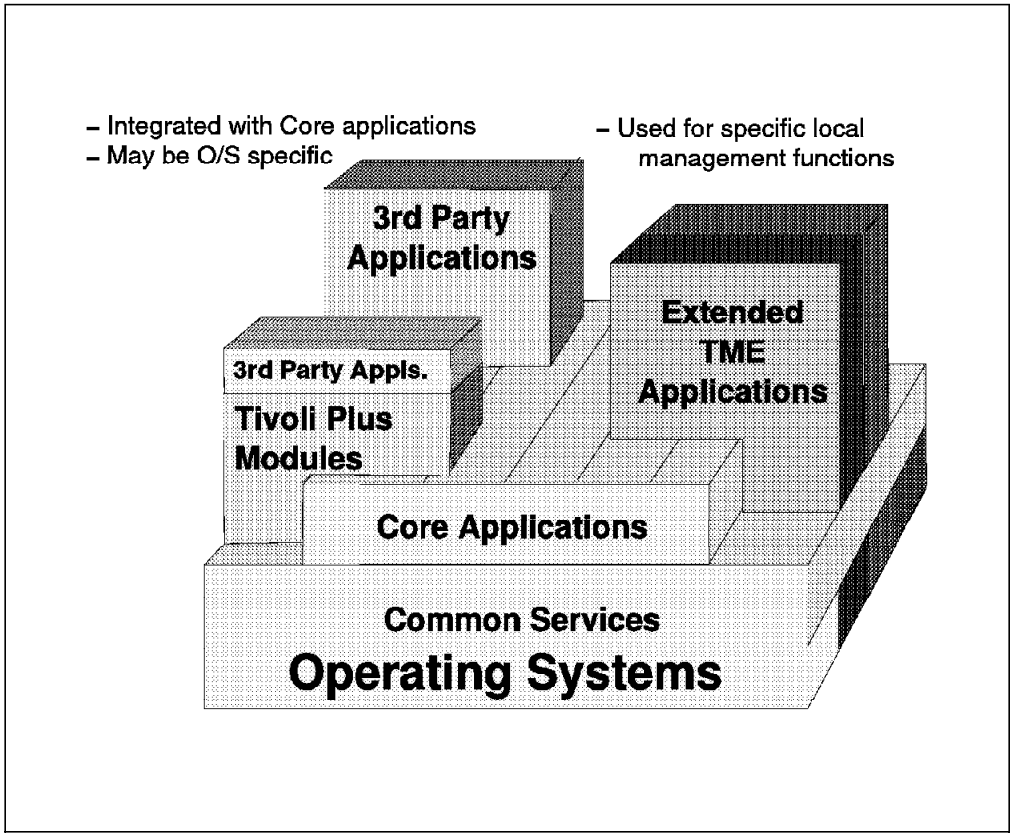


Figure 4. Framework II

To fulfill the functions required for the framework there are lot of products. Some are from IBM and Tivoli, and some from the Tivoli Plus group.

1.1.4.1 Tivoli Products

There are different products from Tivoli that are based on the framework:

- Tivoli Courier - Automates the software distribution process to clients and servers throughout the enterprise. It allows you to install and update applications and software in a coordinated, consistent manner across UNIX and PC platforms. Courier is part of the core applications. It is based in the deployment discipline of the framework.
- Tivoli Sentry - Performs intelligent local monitoring of system resources, initiates corrective actions and alerts administrators of potential problems. It lets you set or change monitoring parameters for hundreds of related systems distributed across remote sites. Sentry is part of the core applications and is placed in the availability discipline.
- Tivoli FSM - Provides a comprehensive solution to the difficult problems of managing access to distributed file systems across a heterogeneous system environment. It is part of the core applications.
- Tivoli Enterprise Console - This is a management application collecting, processing and automatically initiating corrective actions to system, application, network and database events. Built-in event correlation allows you to efficiently pinpoint and focus on problems or critical event information.
- Tivoli Admin - Simplifies the process of user and system administration. It offers efficient automated management of user and system configuration parameters, secure delegation of administrative tasks and centralized

control of UNIX and PC systems across distributed enterprise. It is part of the core applications and resides in the security discipline.

- Tivoli Print - Simplifies and automates common printer management tasks, such as the assignment and queuing of print jobs throughout the network. It is part of the core applications and the operations and administration discipline.
- Tivoli Plus - Module for integrating non-core applications with the TME. An example of this is the ADSM Plus module.

1.2 TME 3.0 NT Environment

The TME 3.0 NT environment used for this project is depicted in the following diagram. We then go on to outline the platform and component configurations used.

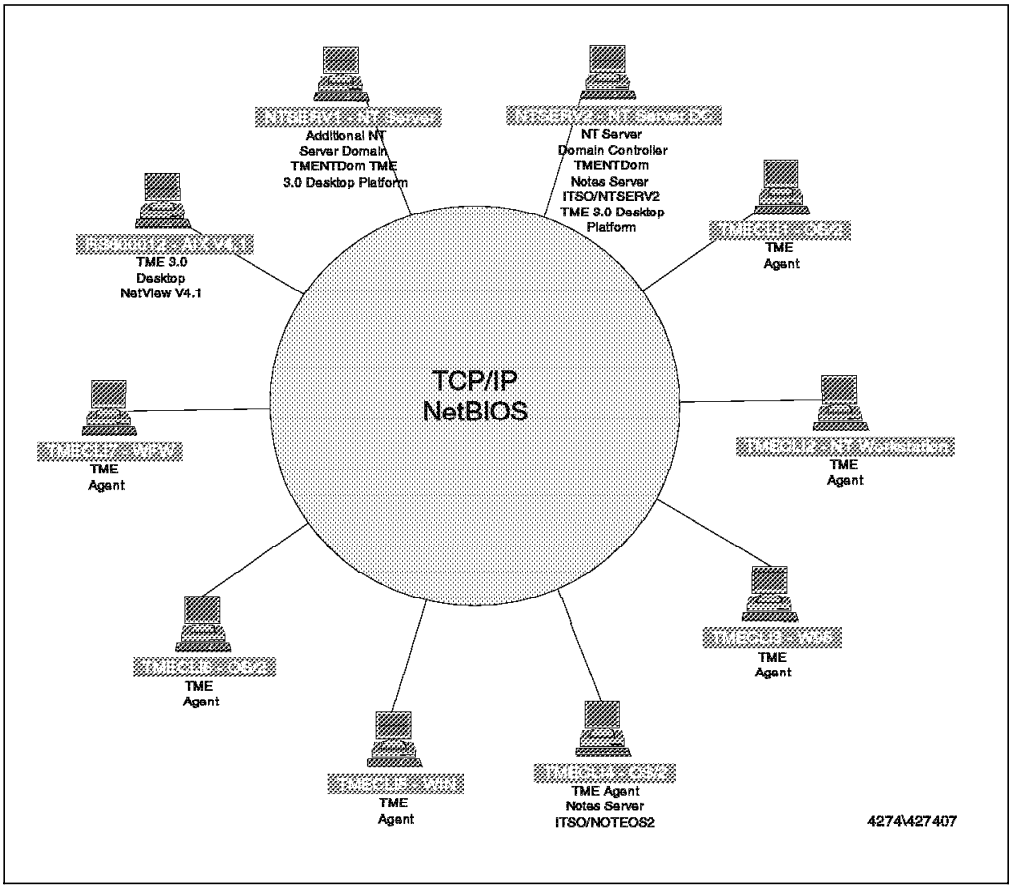


Figure 5. TME 3.0 NT Environment

1.2.1 Windows NT Server

Platform: Microsoft Windows NT Server 3.51 (Service Pack 4)

Hardware: IBM PC 350 P-133 / 1.4 GB / 80 MB

Protocols: TCP / IP, NetBIOS

Software: TME 3.0 Platform / Desktop / Agent, TME 10 NetFinity, Lotus Notes 4.1 Client.

Address: 9.24.104.112

Hostname: NTSERV1

1.2.2 Windows NT Server

Platform: Microsoft Windows NT Server 3.51 (Service Pack 4)

Hardware: IBM PC 350 P-133 / 1.4 GB / 80 MB

Protocols: TCP / IP, NetBIOS

Software: TME 3.0 Platform / Desktop / Agent, SMS Server 1.1, SQL Server 6.0, TME 10 NetFinity, Lotus Notes 4.1 Server

Address: 9.24.104.113

Hostname: NTSERV2

1.2.3 OS/2 Warp Connect Setup

Platform: IBM OS/2 3.0 Warp Connect

Hardware: Model 77i / 1.0 GB / 32 MB

Protocols: TCP / IP, NetBIOS

Software: TME 3.0 Agent, TME 10 NetFinity Server, Lotus Notes 4.1 Client

Address: 9.24.104.114

Hostname: TMECLI1

1.2.4 Windows NT Workstation

Platform: Microsoft Windows NT Workstation

Hardware: Model 9595 / 1.0 GB / 32 MB

Protocols: TCP / IP, NetBIOS

Software: TME 3.0 Agent

Address: 9.24.104.115

Hostname: TMECLI2

1.2.5 Windows 95

Platform: Microsoft Windows 95

Hardware: 9595 / 1.0 GB / 32 MB

Protocols: TCP / IP, NetBIOS

Software: TME 3.0 Agent

Address: 9.24.104.99

Hostname: TMECLI3

1.2.6 OS/2 Warp Connect

Platform: IBM OS/2 Warp Connect

Hardware: Model 77i / 0.5 GB / 32 MB

Protocols: TCP / IP, NetBIOS

Software: TME 3.0 Agent, Lotus Notes 4.1 Server

Address: 9.24.104.80

Hostname: TMECLI4

1.2.7 Windows 3.1

Platform: Microsoft Windows 3.1

Hardware: Model 80 / 150 MB / 16 MB

Protocols: TCP / IP, NetBIOS

Software: TME 3.0 Agent

Address: 9.24.104.87

Hostname: TMECLI5

1.2.8 OS/2 Warp Connect

Platform: IBM OS/2 Warp Connect

Hardware: 9595 / 1 GB / 32 MB

Protocols: TCP / IP, NetBIOS

Software: TME 3.0 Agent

Address: 9.24.104.68

Hostname: TMECLI6

1.2.9 Windows for Workgroups

Platform: Microsoft Windows for Workgroups 3.11

Hardware: 9595 / 1 GB / 32 MB

Protocols: TCP / IP, NetBIOS

Software: TME 3.0 Agent

Address: 9.24.104.69

Hostname: TMECLI7

1.2.10 AIX Setup

Platform: IBM AIX 4.1.4

Hardware: RS6000

Protocols: TCP / IP

Software: NetView for AIX V4.1, TME 3.0 Platform and Desktop

Address: 9.24.104.124

Hostname: rs600012

For all machines:

Subnet mask - 255.255.255.0

Gateway - 9.24.104.1

DNS - 9.24.104.108

Domain ID - itso.ral.ibm.com

Chapter 2. TMR and Client Installation

This chapter shows the installation process for TME 3.0 NT, as well as the installation of a client on the NT platform. In addition, we show the installation of Sentry and the TME 10 NetFinity Client for NT. We also show how to install and customize the TME desktop and agent functions.

2.1 TME V3.0 Platform Installation for Windows NT Server V3.51

The following steps should be performed when installing the TME NT platform:

1. From the Tivoli TME 3.0 CD, run setup.exe from the root of the CD or from a LAN-connected drive.
2. You are presented with the welcome panel. Please note the use of the Next and Cancel buttons. These will be used throughout the installation of Tivoli products.

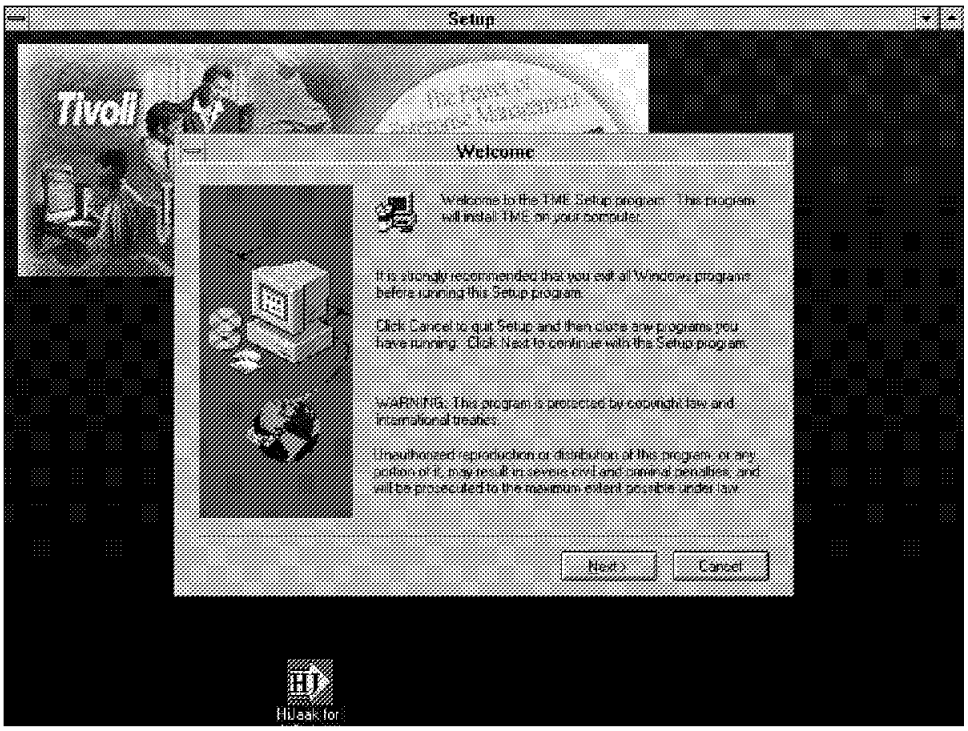


Figure 6. TME 3.0 Platform Installation - Welcome Panel

3. The next panel asks for an administrator to be set up.

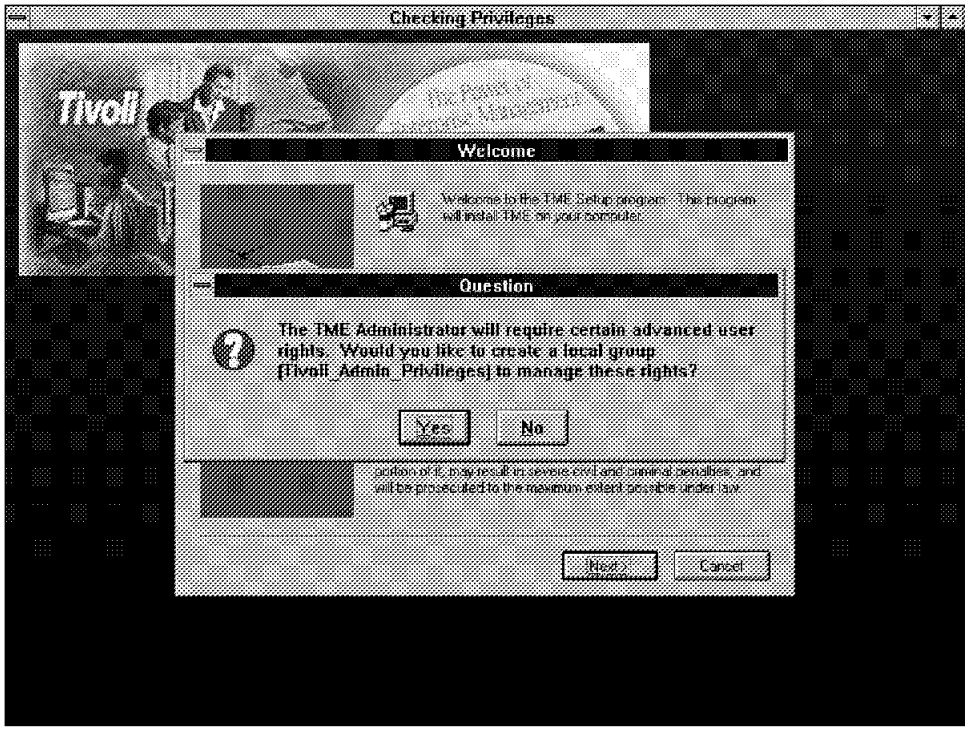


Figure 7. TME 3.0 Platform Installation - Administrator Pop-Up Window

Choosing **Yes** checks your current rights, logs you off and creates the administrator. During the installation Tivoli creates a new local user group under NT called Tivoli_Admin_Privileges. You can see it in the User Manager.

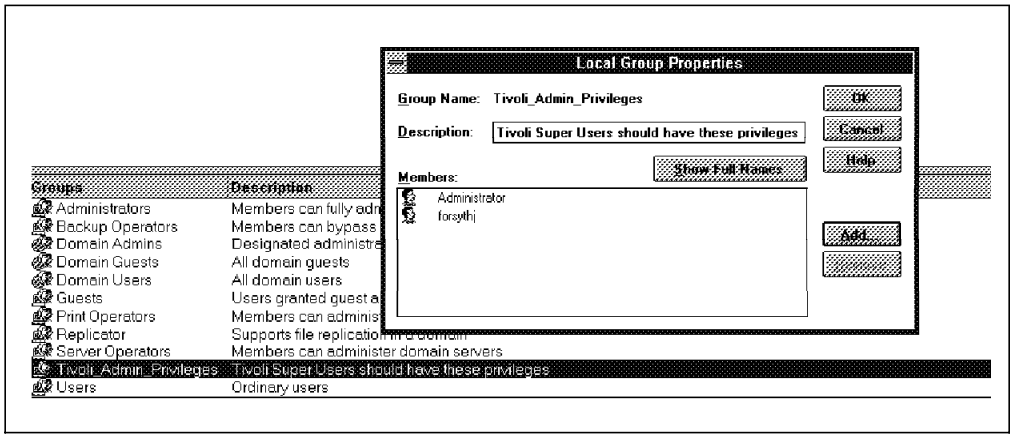


Figure 8. User Rights under NT

It also adds a new user called tmesrvd. The functions of the user are discussed in Chapter 4, “Microsoft NT Server Environment” on page 81.

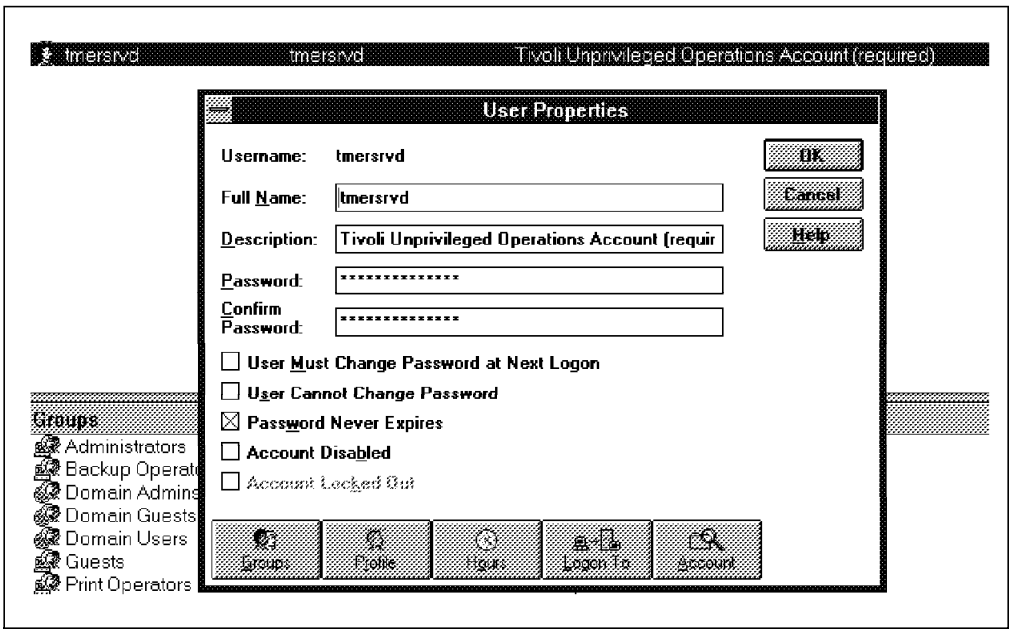


Figure 9. User Rights under NT

If there is no user defined, the installation asks you to log off to add the user.

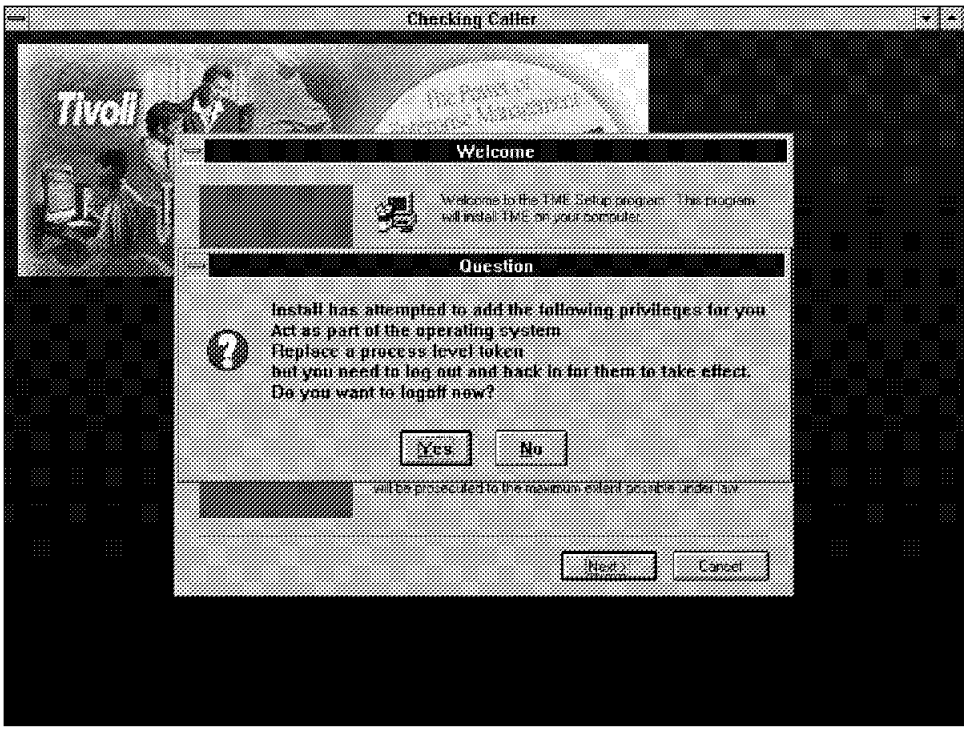


Figure 10. TME 3.0 Platform Installation - Logoff Prompt Window

You have to restart the setup program once the new administrator account is established.

4. Once restarted the Welcome panel is again presented. The next page asks for information regarding your name and company. This information is not required for anything relevant to the setup of the TME environment.

Figure 11. TME 3.0 Platform Installation - Information Screen Panel

5. The next panel asks for an installation password. This password will be used during all of the client installations from this TMR.

Figure 12. TME 3.0 Platform Installation - Password Screen Panel

6. The next panel asks for a remote user to be defined to allow connections to remote network drives. This enables remote installation of TME agents across the network, by using a valid user ID and password.

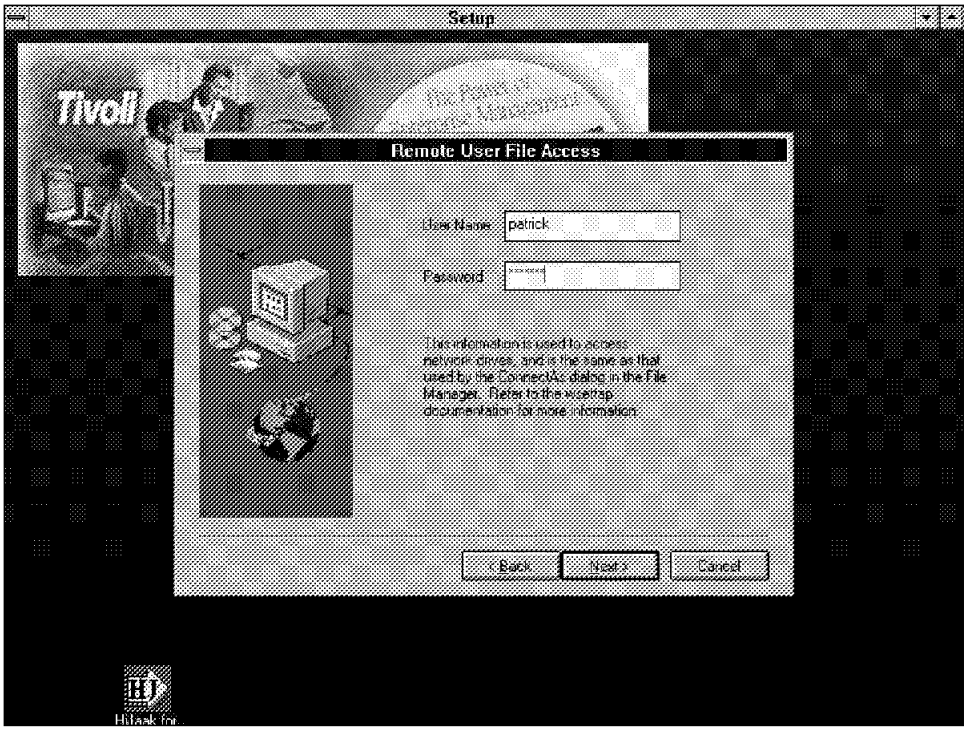


Figure 13. TME 3.0 Platform Installation - Remote User Definition Panel

7. The next panel asks for the installation path to follow. You are presented with three options: Typical, Custom and Compact. You are also asked for a destination directory.

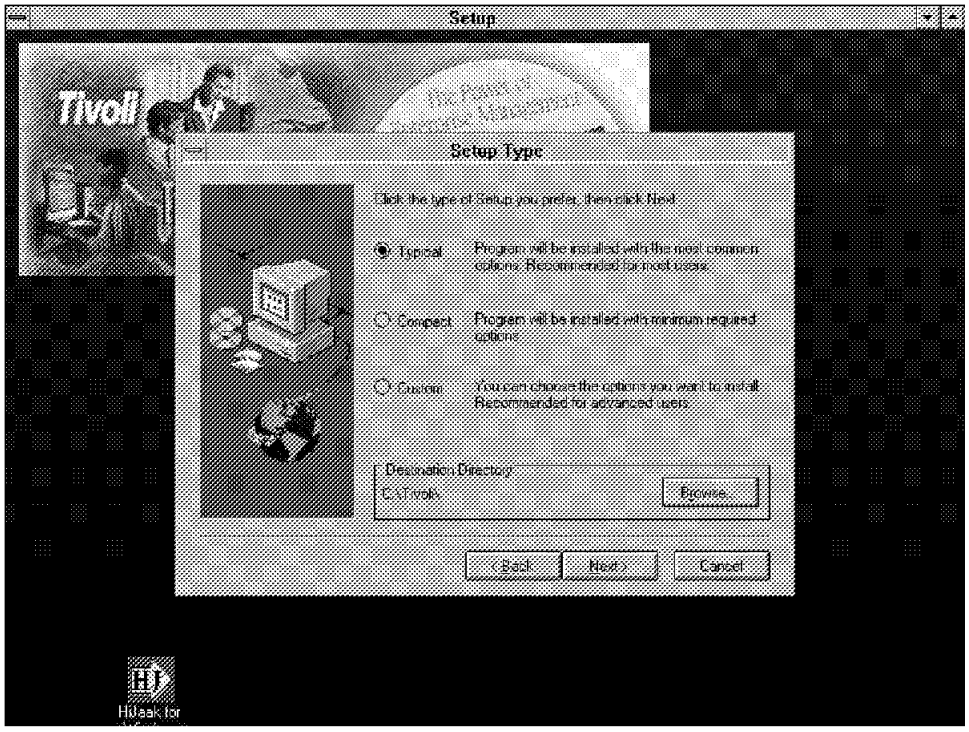


Figure 14. TME 3.0 Platform Installation - Setup Type Panel

You can choose between three types of installation:

- Typical - Installs the most common parts of Tivoli.
- Compact - Installs the minimum parts of Tivoli.
- Custom - You can choose the parts of Tivoli to install.

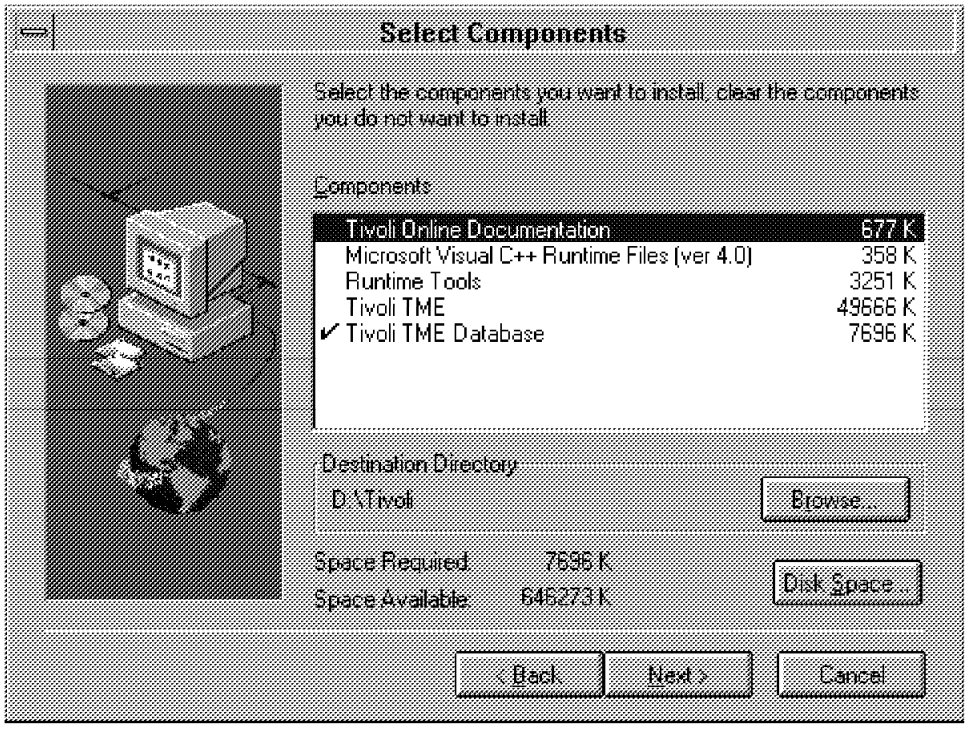


Figure 15. Custom Installation Panel

8. The next panel prompts you for the license key, and is followed by a panel asking for the database destination directory.

> **Note**
>
> The installation process stores the license key and other installation-related information in the NT registry. If you have to re-install you will find some of the fields already filled in for you. In our experience, we found it cleaner and easier to go into the registry, using the registry editor tool (regedt32), and delete all of the Tivoli-related entries.

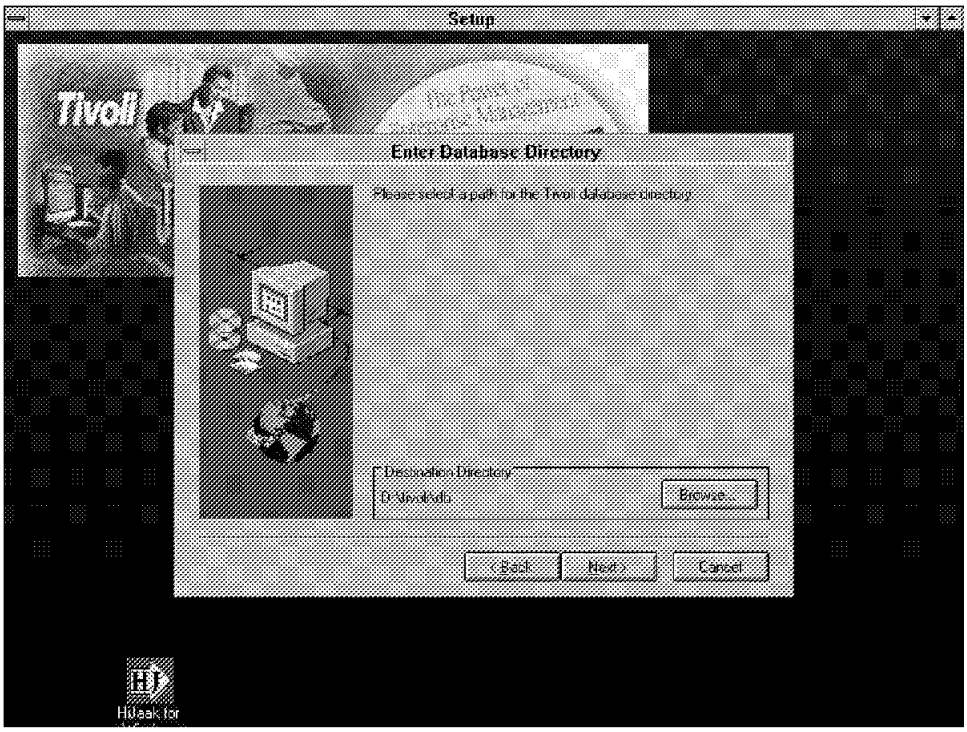


Figure 16. TME 3.0 Platform Installation - Database Directory Panel

9. The TME 3.0 Platform installation now begins.

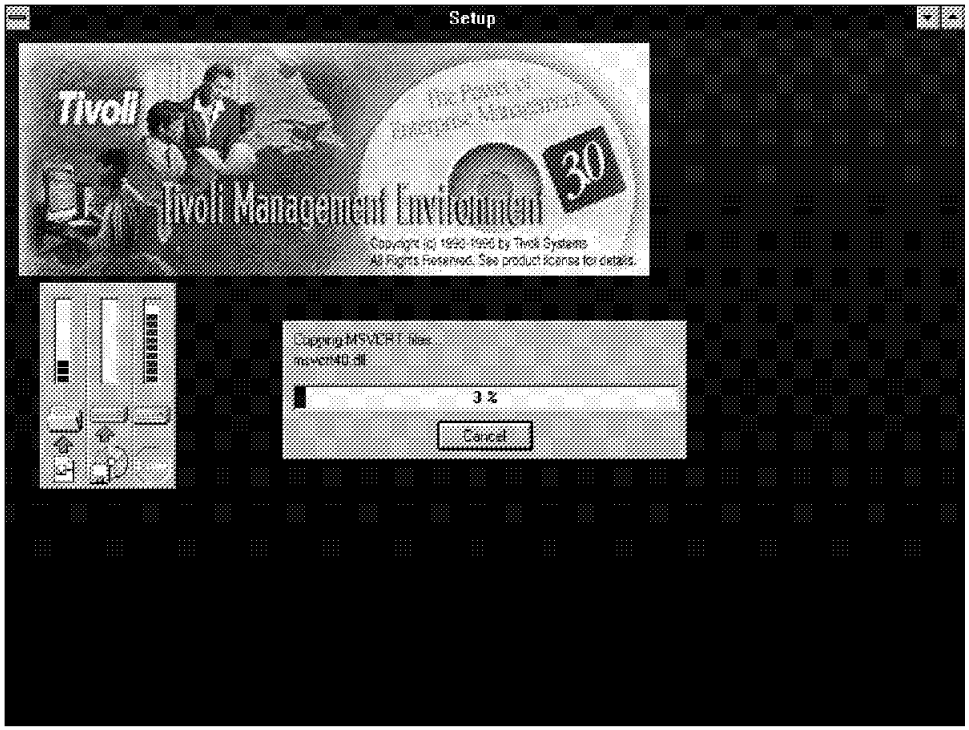


Figure 17. TME 3.0 Platform Installation - Installation Panel

10. During the installation a command prompt appears, describing the Tivoli database being initialized. The database is built in a hierarchical way. That means every object in the database (and on the desktop) has one single identifier. Every object has one parent it belongs to. It can have several sons, that belong to it. Through the inheritance process every object gets the behaviors of the parent object, but they can be changed by the user.

    The root of the database is always the TMR. The TMR is identified through one unique number. You can see this number using the odadmin command. This number builds the first part of the identifier. Each object in this TMR gets its own number or dispatcher. For example, the TME desktop has the dispatcher 1. This number builds the second part of the identifier. For example, if 2114718921 is the region identifier, then the identifier for the TME server is 2114718921.1. Every object that belongs to the server has its own number. For example, 2114718921.1.104 is a Managed Node. The TME server always has the number .0.0.

    If you want to work with the database you can use the following commands from the \Tivoli\db\xxxxx.db directory, or include the *-k* parameter to point to the base directory:

    - **odbls -a -l -i -m -O -k c:\tivoli\db\xxxxx.db**

      Before you use this command you have to shut down the TME desktop and stop the oserv daemon:

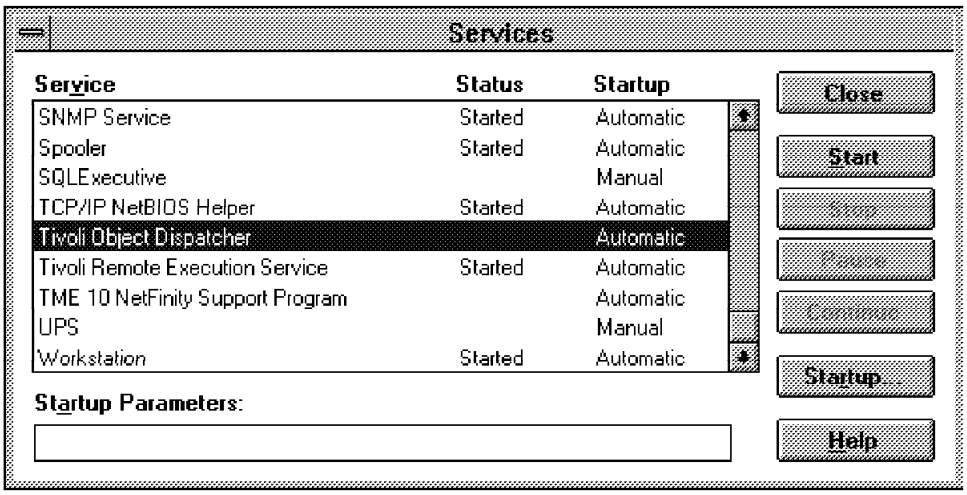


Figure 18. Stop Oserv to Work With the Database Using the odbls Command

This command lists the contents of the database. The parameters are:

- -a displays the attributes in the database.
- -I walks through the inheritance list.
- -i displays the inheritance trees in the object database (only for the TME server).
- -m displays the methods headers (only for TME server).
- -O default, walks through the database.

For the following commands you will need to restart oserv.

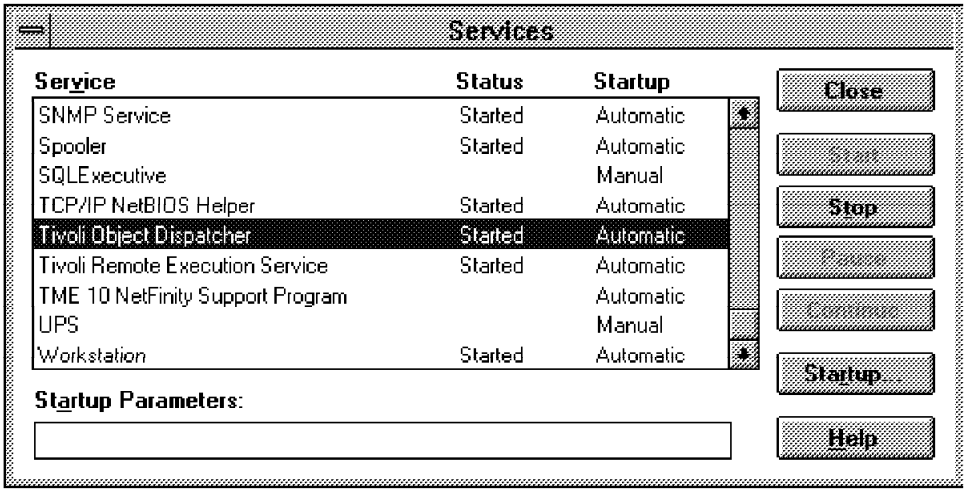


Figure 19. Start oserv to Work with the Database

- **wls**

  This command lists the members of the current collection from the current location in the database tree.

- **wcd**

  Changes the current working collection. It can be used to go to the leaves of the database tree.

- **wpwd**

  Shows the current collection.

We show some outputs for the commands to get a feeling for the database structure.

Extraction of the output of the odbls command

```
2114718921.0.0
    attributes:
2114718921.1.0
    attributes:
 _HostLocation "2114718921.1.322#TMF_ManagedNode::Managed_Node#"
 _NameRegistry "2114718921.1.26"
 _baselist "2114718921.1.0 "
 _fileioRef <binary, 39 bytes: d\x84\x00\x00\x00!\x01\x01\x00\x1b\x1c2114
 _master_base_oid "2114718921.1.0"
 _oserv "2114718921.1.2"
 _security_objid "2114718921.0.0"
 _skeleton "2114718921.1.1"
 iflag.oserv "<No Data>"
 iflag.security_objid "<No Data>"
2114718921.1.1
    attributes:
2114718921.1.10
    attributes:
 __BOA_id <binary, 12 bytes: a\x84\x00\x00\x00\x06\x02\x04\x00\x00\x00>
 _class_objid <binary, 57 bytes: d\x84\x00\x00\x003\x01\x01\x00\x1b.21147
 _collections <binary, 75 bytes: a\x84\x00\x00\x00E\x02\x04\x00\x00\x00\x
 _label <binary, 17 bytes: d\x84\x00\x00\x00\x0b\x01\x01\x00\x1b\x06trito
 _pres_object <binary, 21 bytes: d\x84\x00\x00\x00\x0f\x01\x01\x00\x1b\x0
 _pro <binary, 21 bytes: d\x84\x00\x00\x00\x0f\x01\x01\x00\x1b\x0aOBJECT_
 _pro_name <binary, 11 bytes: d\x84\x00\x00\x00\x05\x01\x01\x00\x1b>
 _resource_host <binary, 21 bytes: d\x84\x00\x00\x00\x0f\x01\x01\x00\x1b\
```

Output of the wls -l command

```
2114718921.1.179#TMF_BBoard::GUI# Notices
2114718921.1.168#TMF_Administrator::Collection_GUI# Administrators
2114718921.1.195#TMF_PolicyRegion::GUI# ntserv1-region
2114718921.1.641#TMF_TGC::CollectionGUI# test
```

Output of the wcd Administrators command and the wls command

```
2114718921.1.178#TMF_Administrator::Configuration_GUI# Root_ntserv1-regio
1149853629.1.178#TMF_Administrator::Configuration_GUI# Root_ntserv2-regio
```

Another interesting function might be to use the grep command to search through the database for a specific string. For example:

```
odbls -a -l -i -m -O -k c:\tivoli\db\xxxxx.db > c:\temp\search.out
grep -i -e monit c:\temp\search.out|more
```

That will search for all occurences of the word monit in the database.

This command changes the directories. You are now in the administrators directory of the database. You can change the directory to the Root_ntserv1-region by using the wcd command again.

Continuing with the installation process:

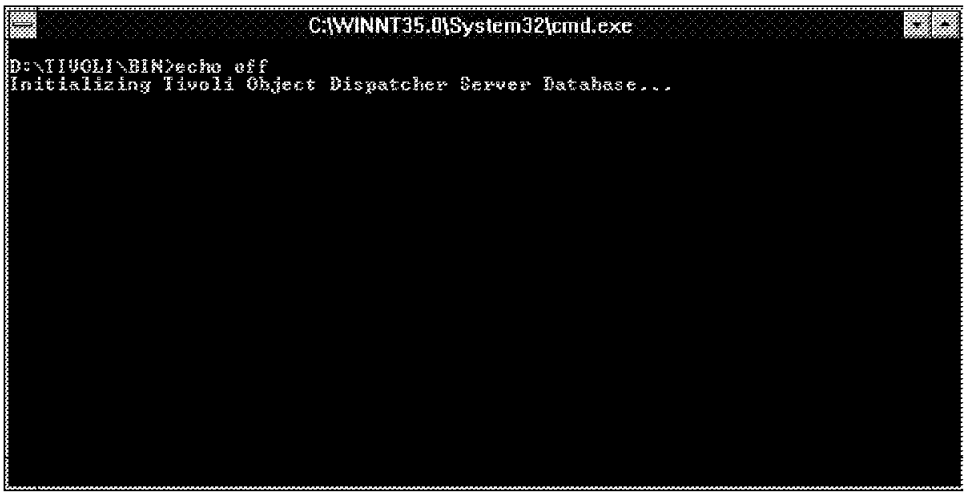


Figure 20. TME 3.0 Platform Installation - Database Initialization Panel

Confirmation is required when the database initialization ends.

Figure 21. TME 3.0 Platform Installation - Initialization Confirmation Panel

11. Once initialization completes you are prompted to reboot.

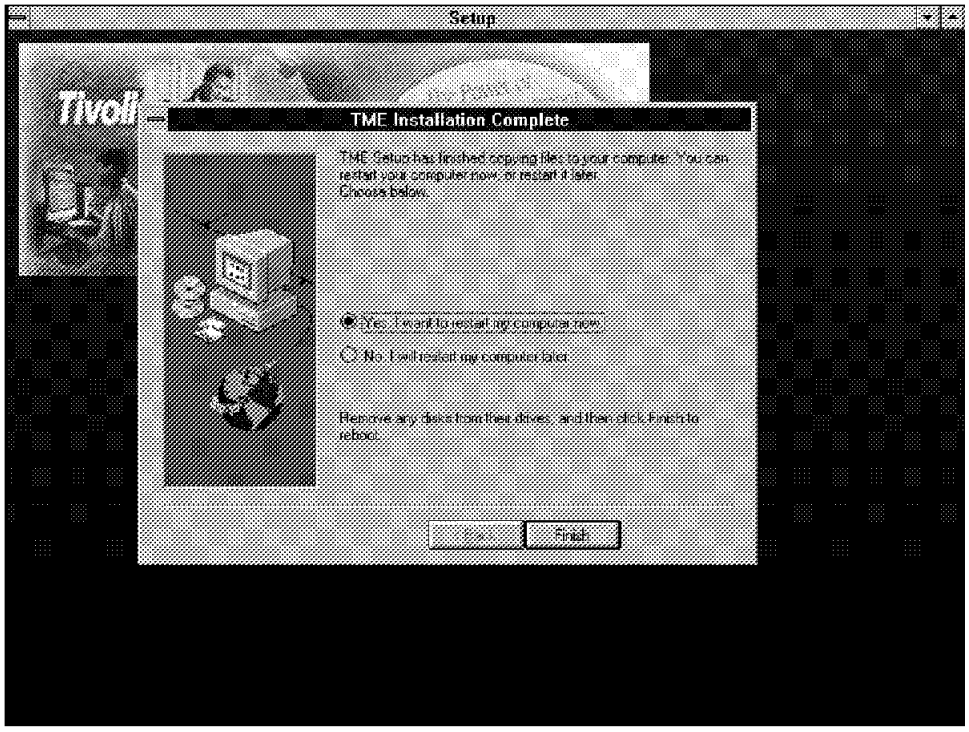


Figure 22. TME 3.0 Platform Installation - Installation Complete Panel

After the system restarts there is no folder or application for TME on the desktop, only the TME 3.0 Platform has been installed. We then had to install the TME 3.0 Desktop.

2.2 TME V3.0 Desktop Installation for Windows NT Server V3.51

The following steps should be performed when installing the TME NT Desktop:

1. From the Tivoli TME 3.0 CD or from a LAN drive run the following:

   `\pc\desktop\disk1\setup.exe`.

   a. The initial screen asks for an installation path.

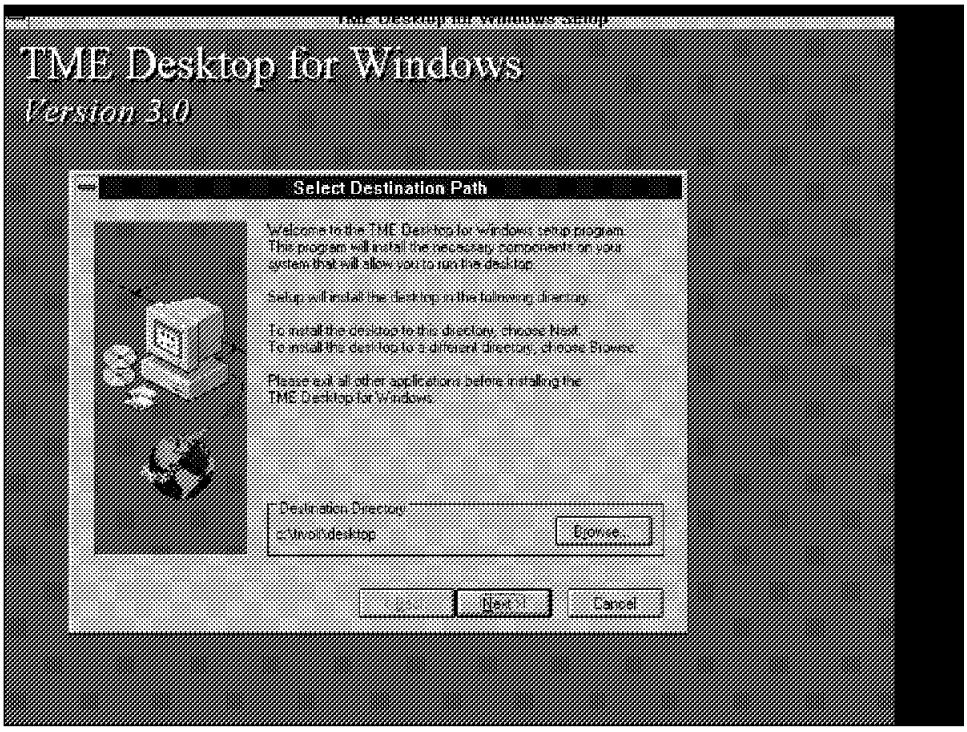


Figure 23. TME 3.0 Desktop Installation - Destination Path Panel

b. The following panel asks for an NT desktop group name.

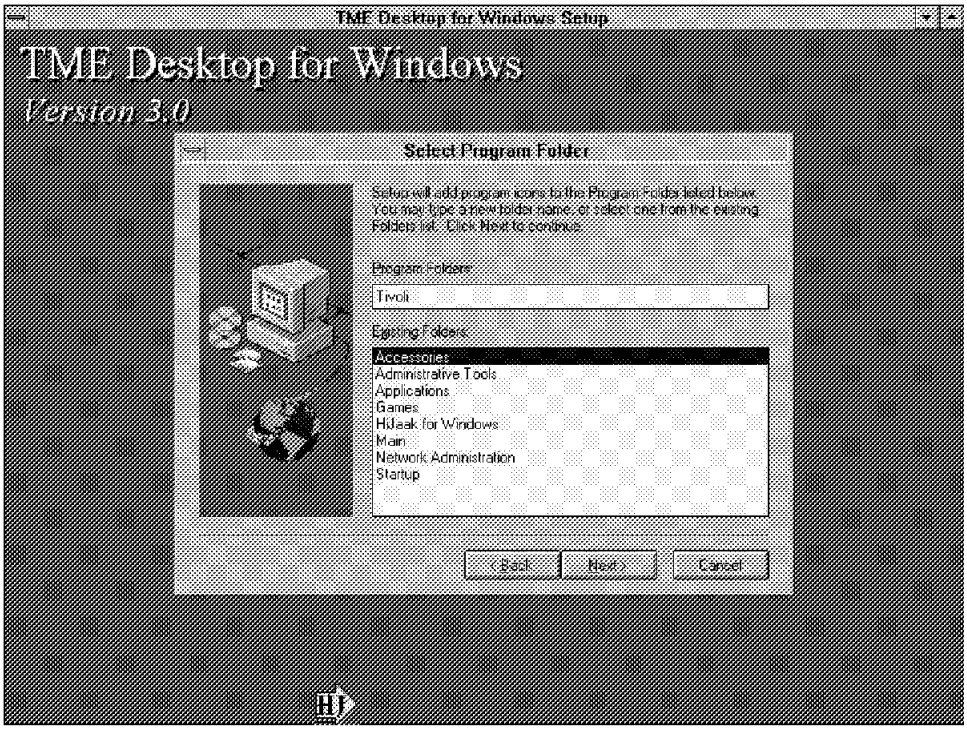


Figure 24. TME 3.0 Desktop Installation - NT Desktop Group Panel

c. The files are then transferred to the NT server, and the TME 3.0 Desktop installation is complete.

2.2.1 TME Desktop Directory Structure

The directory structure of the Tivoli platform, when installed under NT, resembles the following:

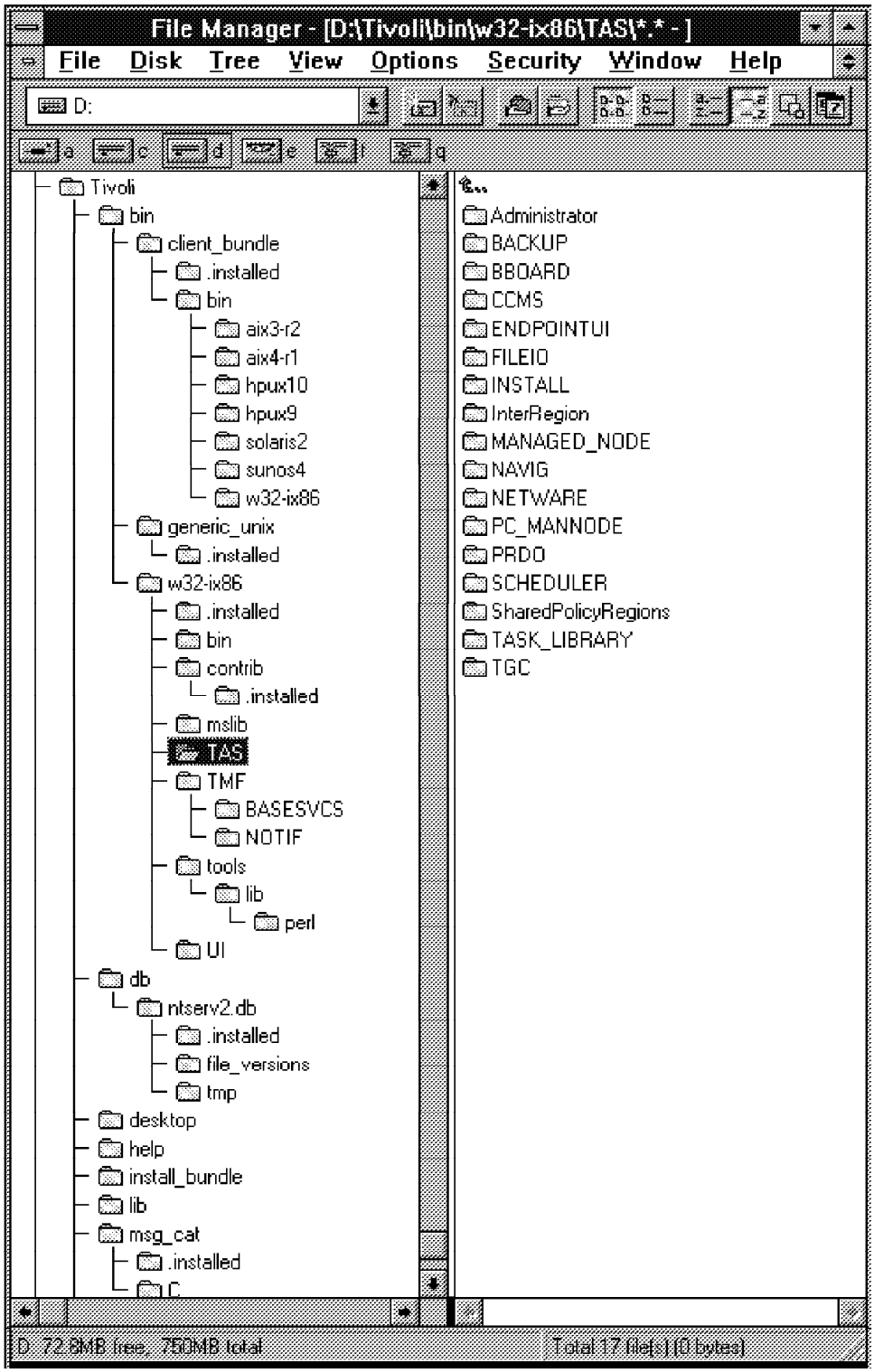


Figure 25. TME 3.0 Desktop Installation - NT Directory Structure

- **Tivoli\bin** - Contains all the executables required for the Tivoli platform and application services. Also held here are the client bundles for different platforms.
- **Tivoli\db** - Contains the database files for the TME server, in this case for ntserv2.
- **Tivoli\desktop** - Contains the database and executables to run the TME desktop.
- **Tivoli\help** - Contains the help files.
- **Tivoli\install_bundle**- Contains scripts for installed software.
- **Tivoli-lib** - Contains library files.
- **Tivoli\msg_cat** - Contains all the possible message errors and catalog files required for the TME platform.

2.3 TME V3.0 Platform and Desktop Installation for AIX V4.1.4

The following steps should be performed when installing the TME AIX platform:

1. From the source directory run the script wpreinst.sh, followed by:

   ```
   wserver -c /<path to Tivoli Software>
   ```

2. Ensure there is enough space in the /var and /usr file systems before installing.
3. From the Primary Install panel, select **Install Options** to select the directory destinations for the different component parts. Check the boxes to permit the creation of the new directories. In addition, select the option to permit the daemon to start up after the reboot of NT as well as to allow the Tivoli daemon to be remotely started.

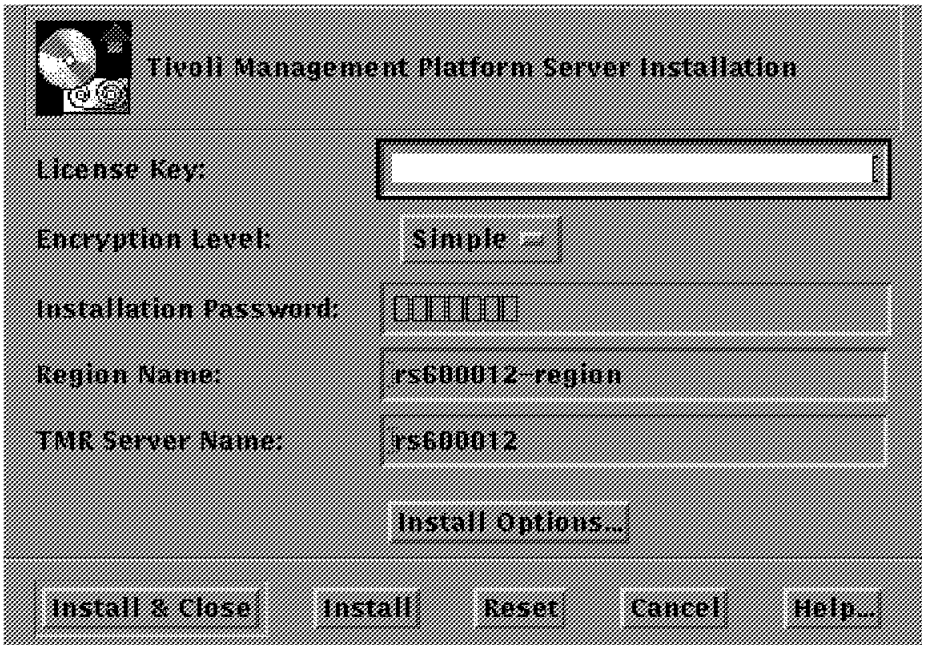


Figure 26. TME 3.0 Desktop AIX Installation - Primary Panel

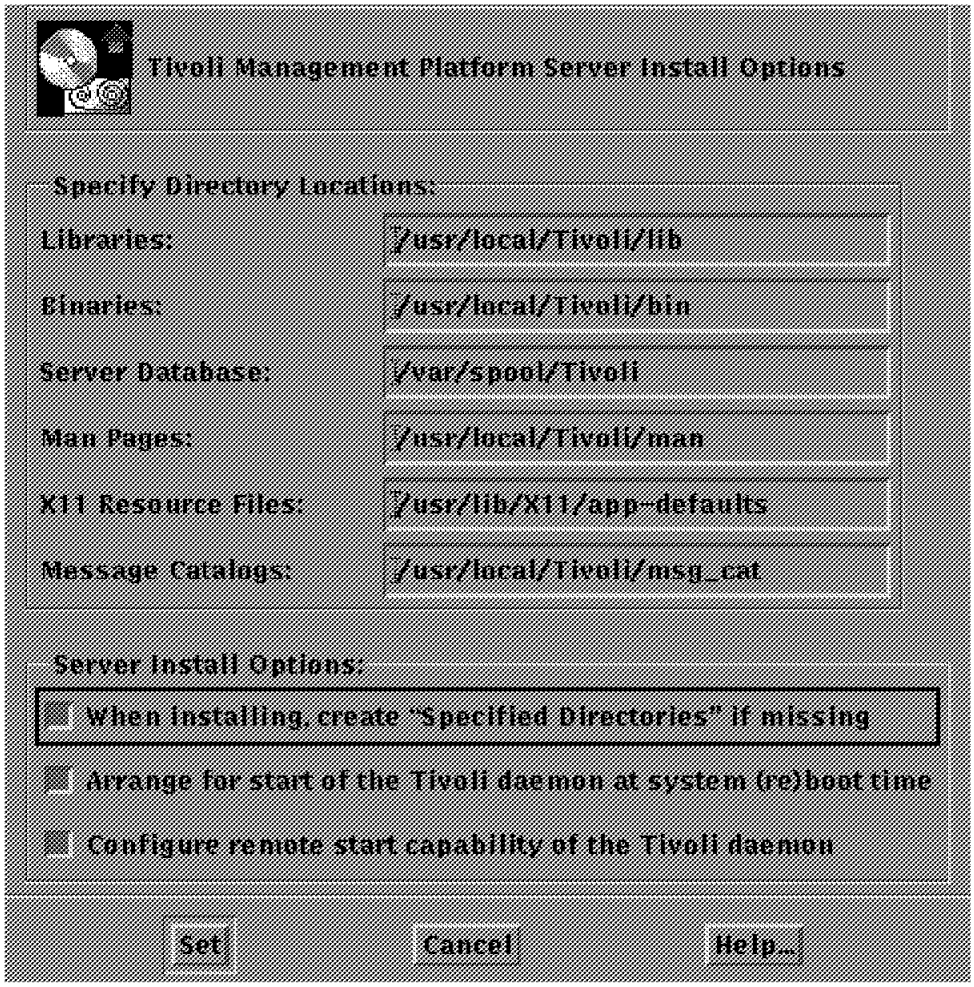


Figure 27. TME 3.0 Desktop AIX Installation - Install Options Panel

4. Once the install options are set, along with a valid license key, encryption level, installation password, region and server name, the install process can begin. The first stage then presents a confirm panel before continuing the process.

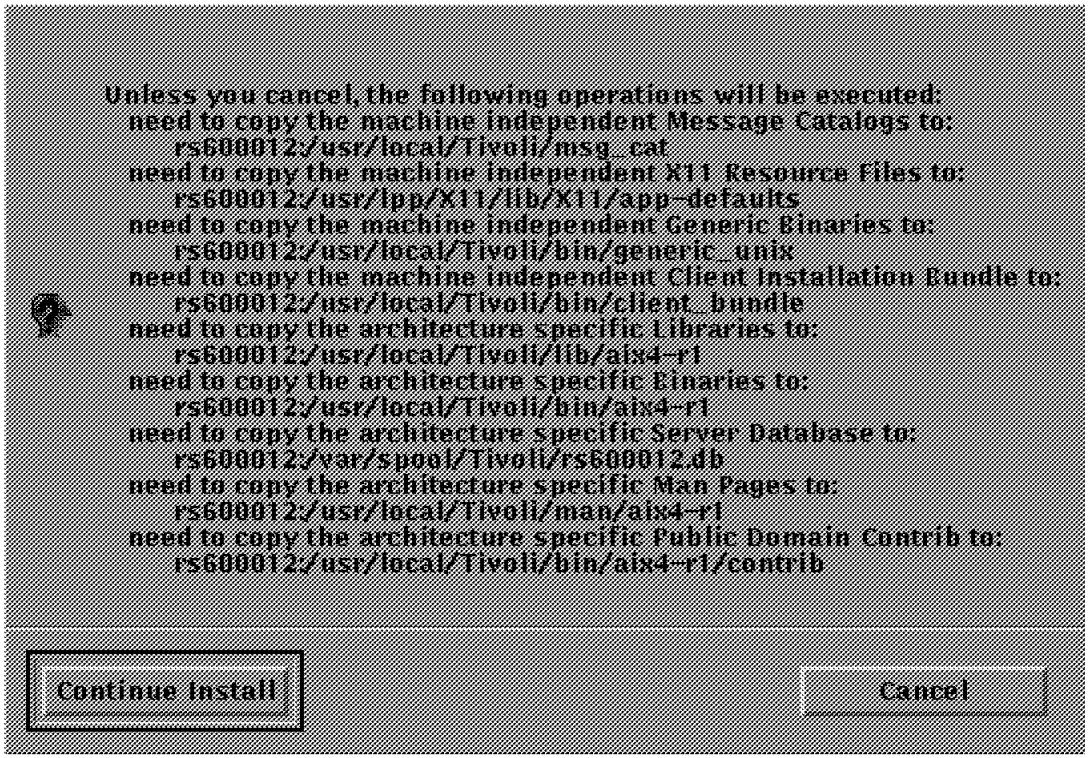


Figure 28. TME 3.0 Desktop AIX Installation - Continue Installation Panel

5. The install procedure then begins the transfer of files to the different areas defined in the Install Options panel.

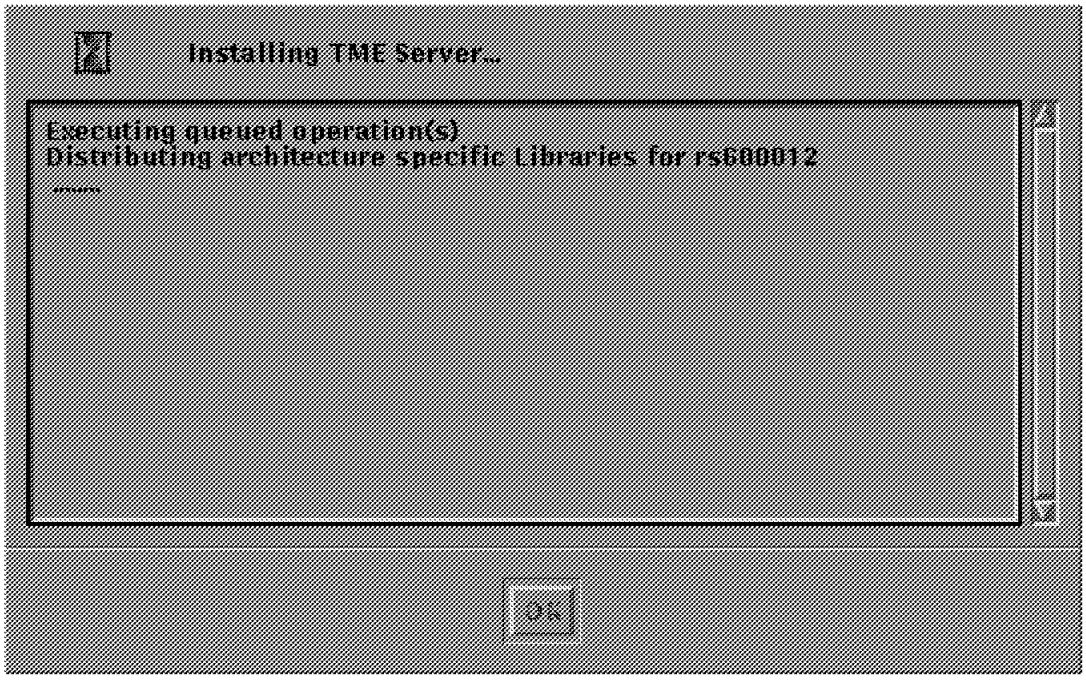


Figure 29. TME 3.0 Desktop AIX Installation - Progress Indicator Panel

6. Once the install is completed, the Tivoli desktop starts.

As you can see in Figure 30 on page 28, the desktop interface is the same as the one that you saw on NT.

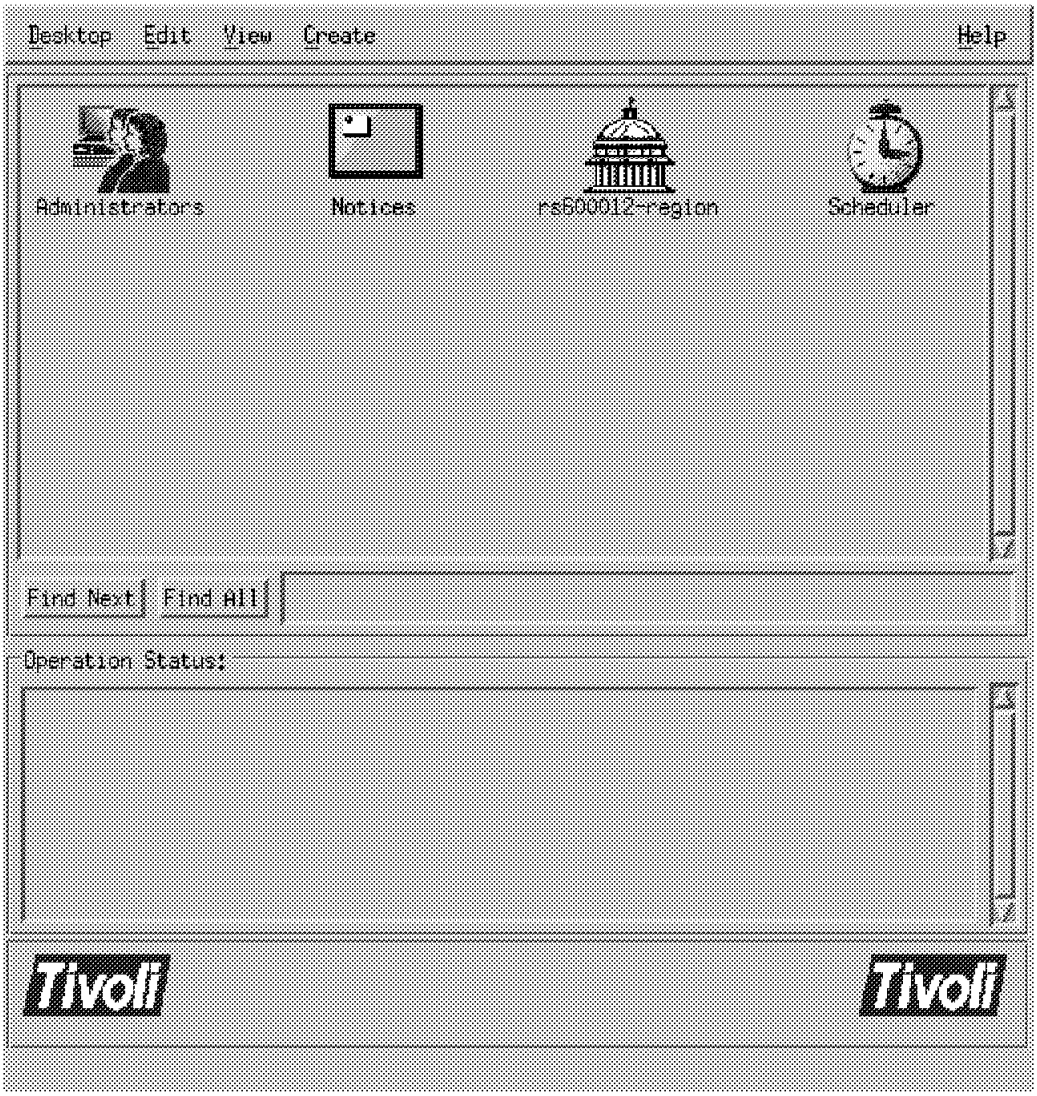


Figure 30. TME 3.0 Desktop AIX Installation - Desktop Panel

7. Finally, the progress indicator panel notifies of the completion of the TME Desktop installation for AIX. Click on **OK**.

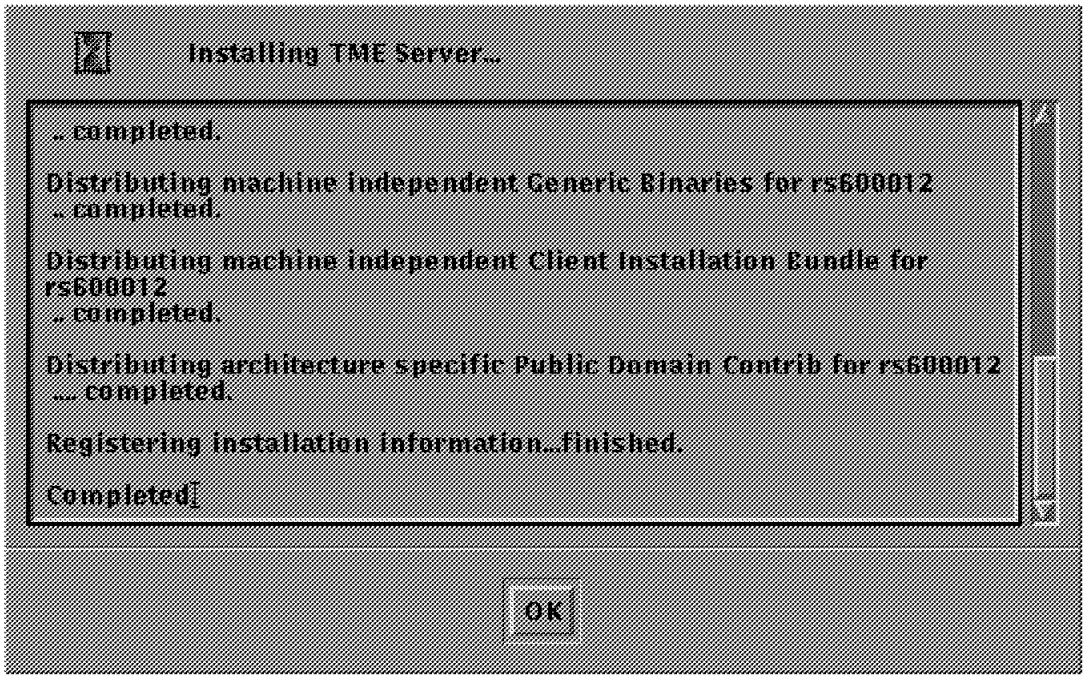


Figure 31. TME 3.0 Desktop AIX Installation

2.4 TME V3.0 Agent Installation

The agent portion for TME 3.0 NT is available on many different platforms. These platforms vary from Intel-based x86 platforms to different flavors of UNIX. We discuss the installation process for a number of these platforms and mention the other platforms available.

2.4.1 Agent Installation on Multiple Operating System Platforms

Agents were installed on the following platforms:

1. Microsoft Windows NT Server 3.51
2. Microsoft Windows NT Workstation 3.51
3. Microsoft Windows 95
4. Microsoft Windows for Workgroups 3.11
5. Microsoft Windows 3.1
6. IBM OS/2 Warp Connect
7. AIX V4.1.4

The installation procedure for all of the Intel X86 operating system platforms is almost identical. The minor differences are pointed out in the following figures.

2.4.2 TME 3.0 Agent Installation

1. From the Tivoli 3.0 CD, run setup.exe from the \PC\TCPAGENT\CD directory. The first panel that will appear is the Welcome panel.

Figure 32. TME 3.0 Agent Installation - Welcome Panel

2. The following screen presents you with several options. These options refer to which platform you are installing the agent on. Choose the one relevant to your operating system.

   Note: The difference between NT(Console) and NT(Service) is that the *console* agent writes information to a console window such as OS/2, NetWare and DOS, whereas the *service* window writes to the Windows NT applications event registry.

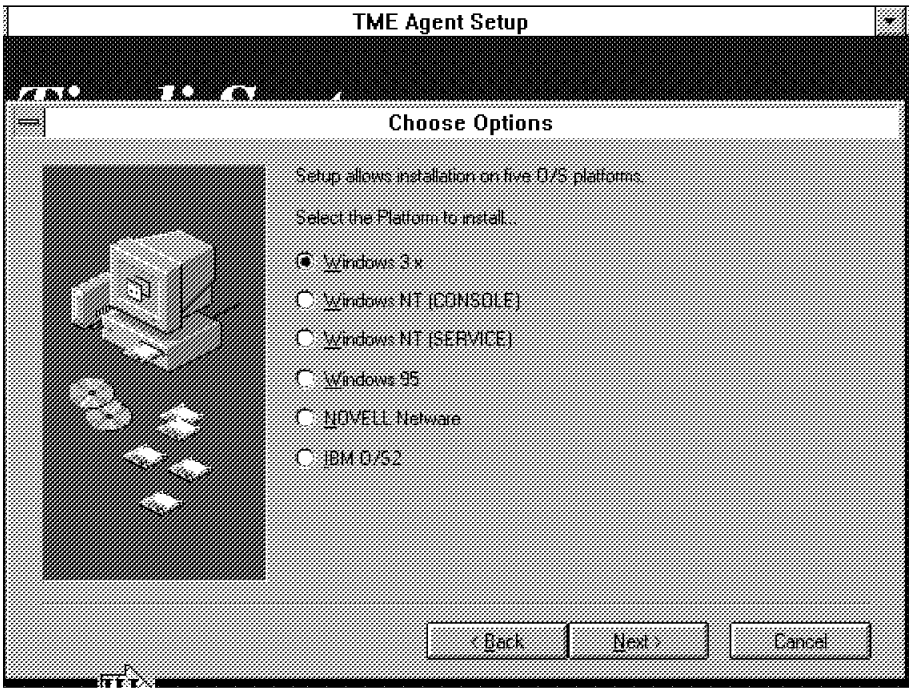


Figure 33. TME 3.0 Agent Installation - Choose Options Panel

3. You are then given an opportunity to read the information page relevant to the install being performed.

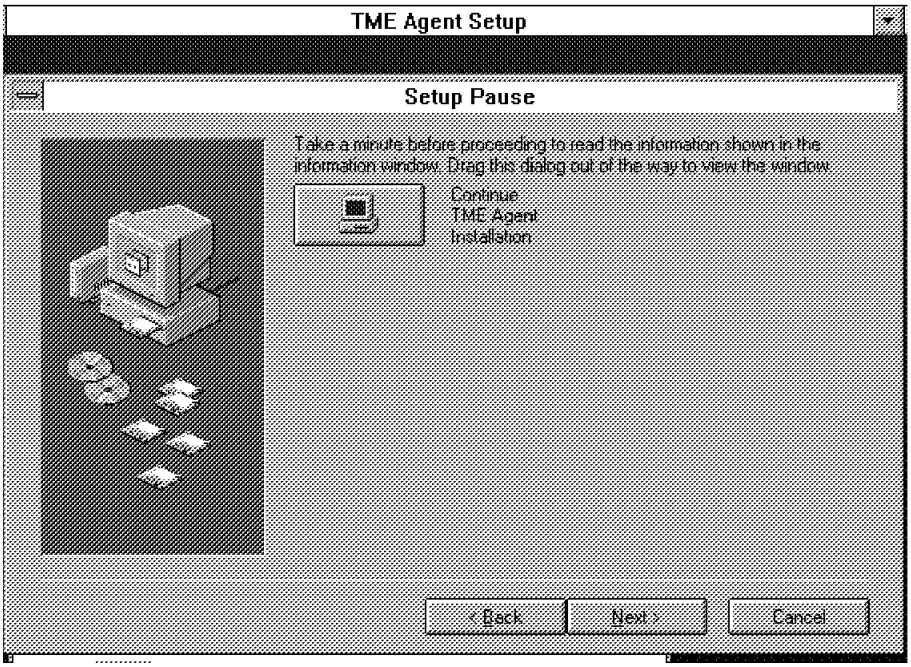


Figure 34. TME 3.0 Agent Installation - Setup Pause Panel

4. This is followed by a prompt for a destination path for the installation. It is recommended that the destination drive have enough space to accommodate the installation.

Figure 35. TME 3.0 Agent Installation - Drive Destination Path Panel

5. A check is then performed for previously installed Tivoli products.

Figure 36. TME 3.0 Agent Installation - Installation Check

6. Following this, a path is requested to install the agent.

Figure 37. TME 3.0 Agent Installation - Destination Path Panel

7. A panel is then presented to confirm the paths chosen.

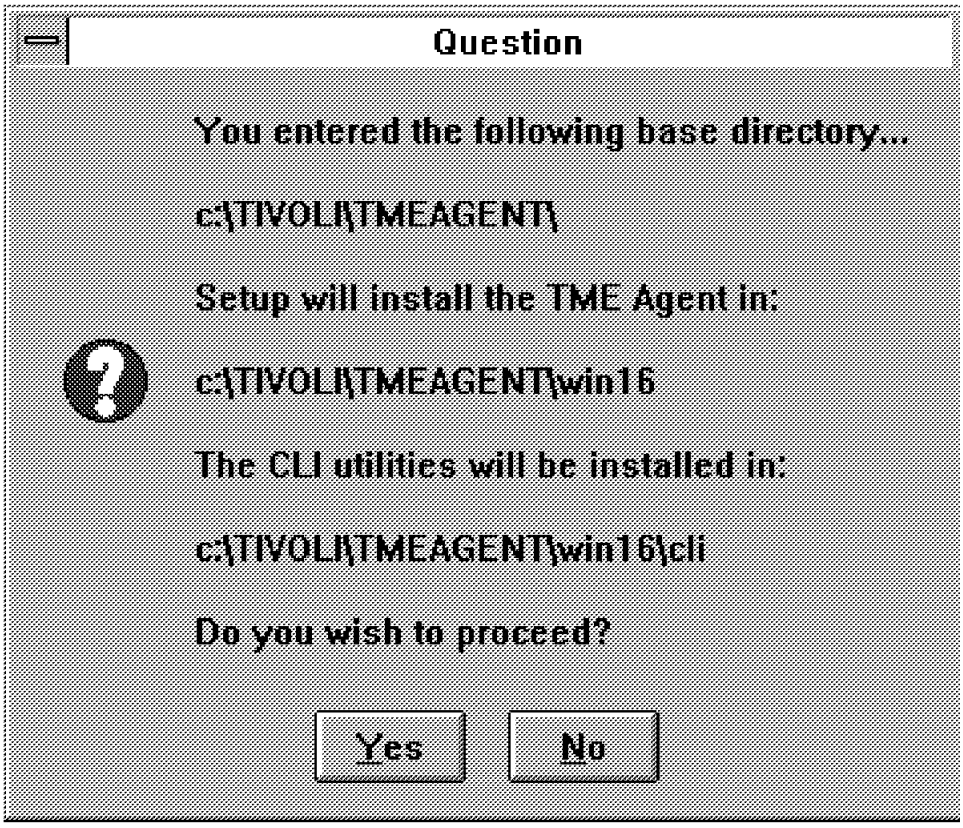


Figure 38. TME 3.0 Agent Installation - Installation Confirmation Panel

8. The next stage asks for some installation options for the agent.

- Install Agent (This option is needed at all times.)
- Start Automatically (adds the Tivoli agent to your startup).
  - DOS - Modifies the autoexec.bat file
  - NT - Adds the agent icon to the startup and starts a new service
  - NetWare - Modifies the NetWare servers autoexec.ncf
  - OS/2 - Modifies the tcpexit.cmd
  - Windows / WFW / W95 - Adds the agent icon to the startup group
- Start after install.

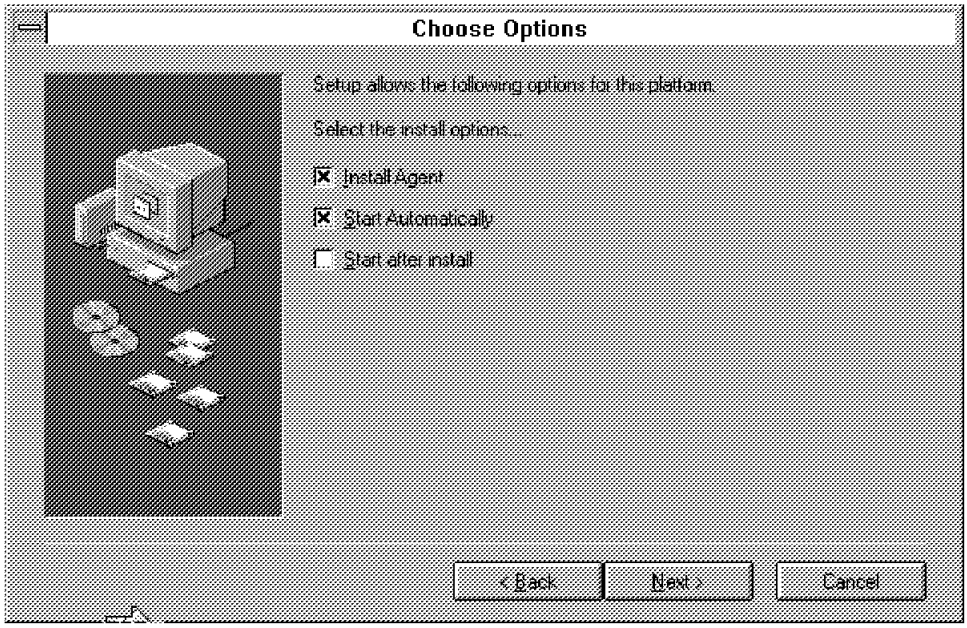


Figure 39. TME 3.0 Agent Installation - Choose Options Panel

9. The next panel asks for the location of a startup file. Under all except for OS/2, it is C:. Under OS/2, the location of the TCPSTART.CMD file is required.

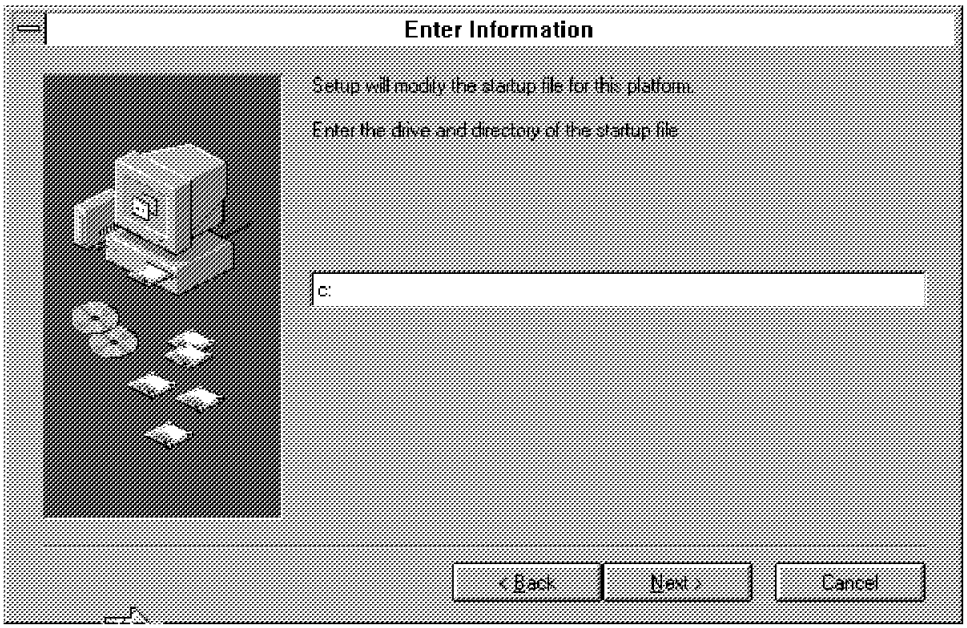


Figure 40. TME 3.0 Agent Installation - Startup File Panel

10. Another pause screen appears, in order to ensure that no Tivoli agent is currently on the system.

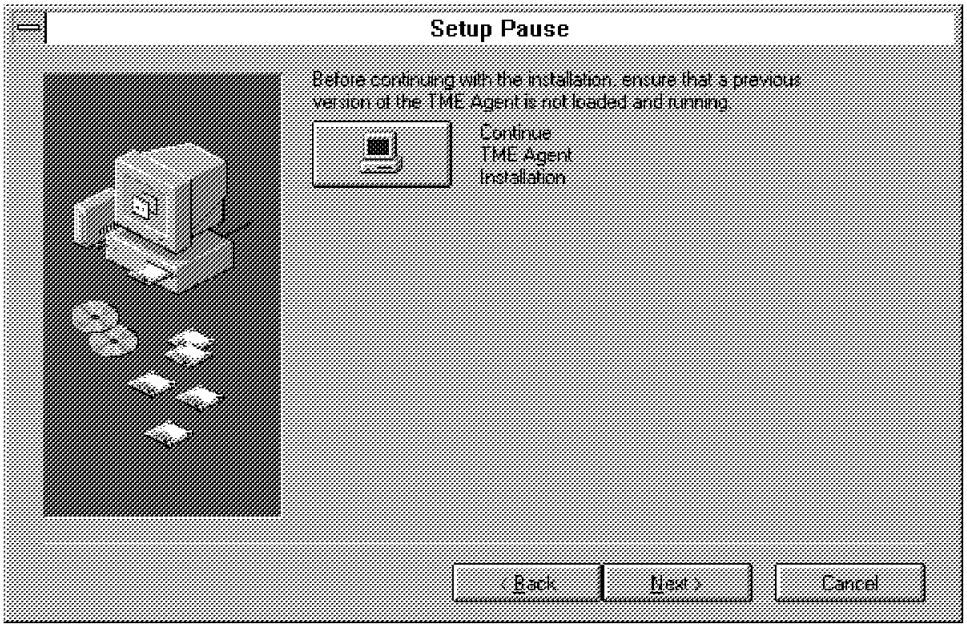


Figure 41. TME 3.0 Agent Installation - Check for Previous Agents

11. The transfer of files begins, followed by a prompt for the machine name that is being installed.

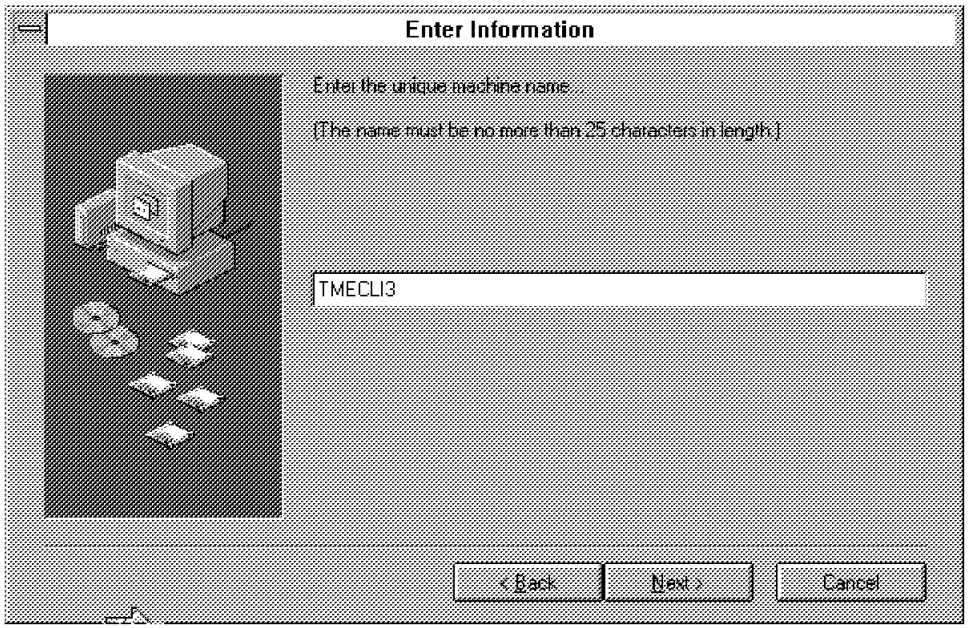


Figure 42. TME 3.0 Agent Installation - Machine Name

12. Next a window appears asking for a remote server to be entered for IP synchronization. Enter any valid TME 3.0 manager.

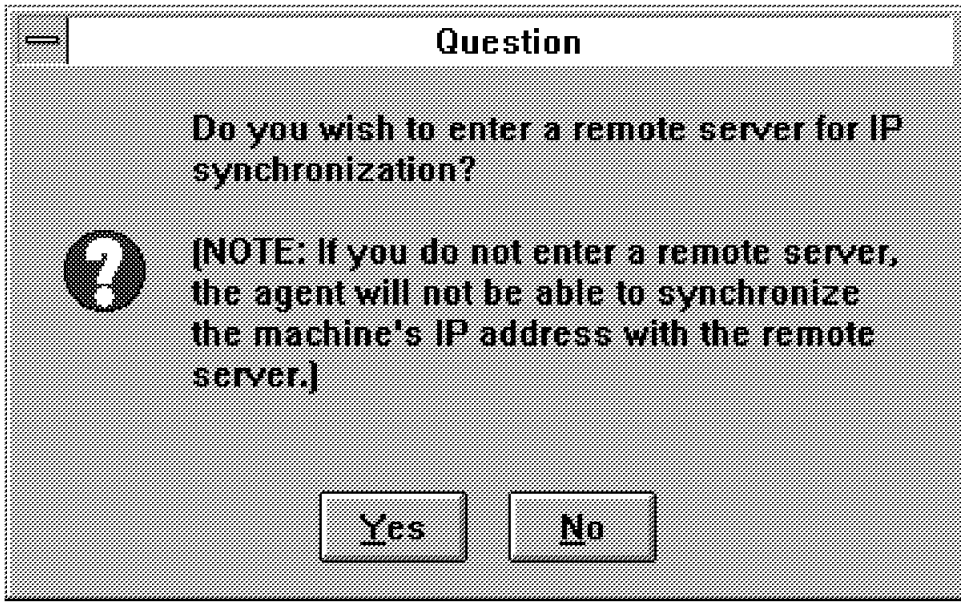


Figure 43. TME 3.0 Agent Installation - IP Synchronization

13. We recommend you enter a remote server. This prompts for information about the remote server.

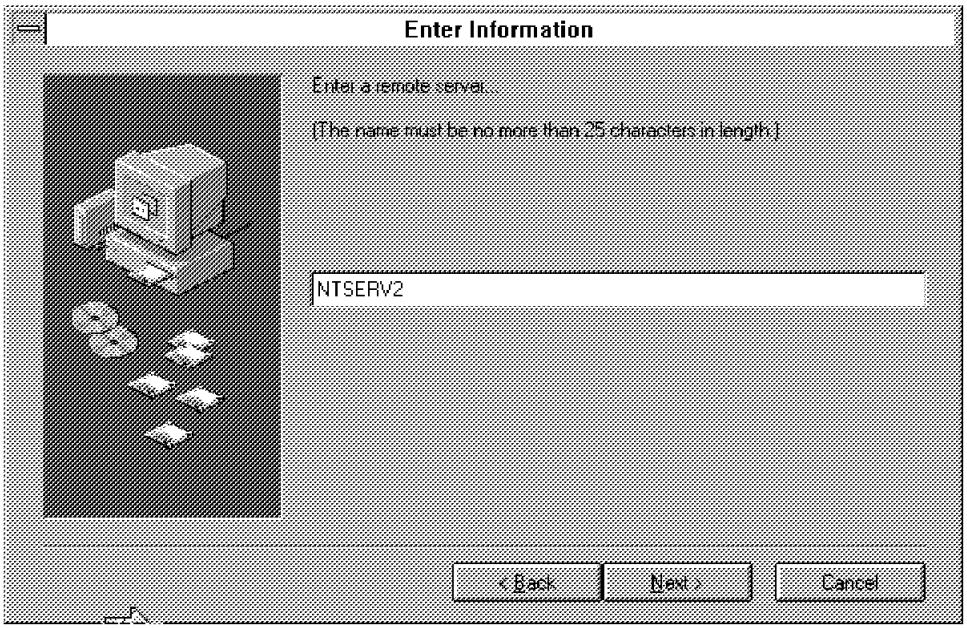


Figure 44. TME 3.0 Agent Installation - Remote Server Information

14. You are also asked whether the agent should log in to the server for IP synchronization at bootup. Click on **Yes**.

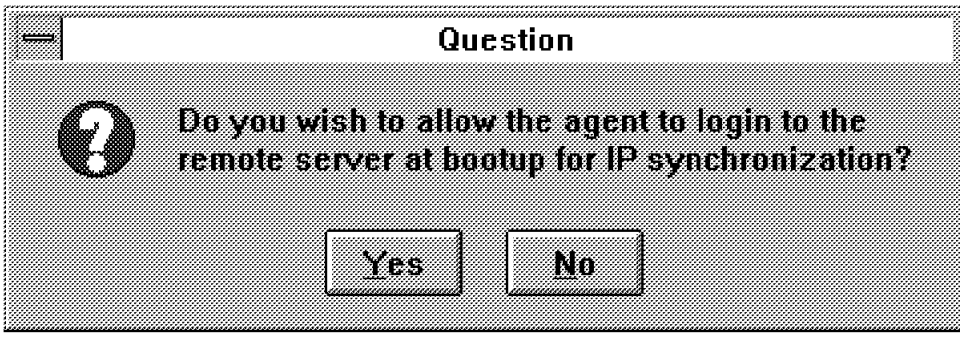


Figure 45. TME 3.0 Agent Installation - Enable Synchronization at Bootup

15. The final screen asks for an update interval for the agent. The default is one day (1440 minutes).

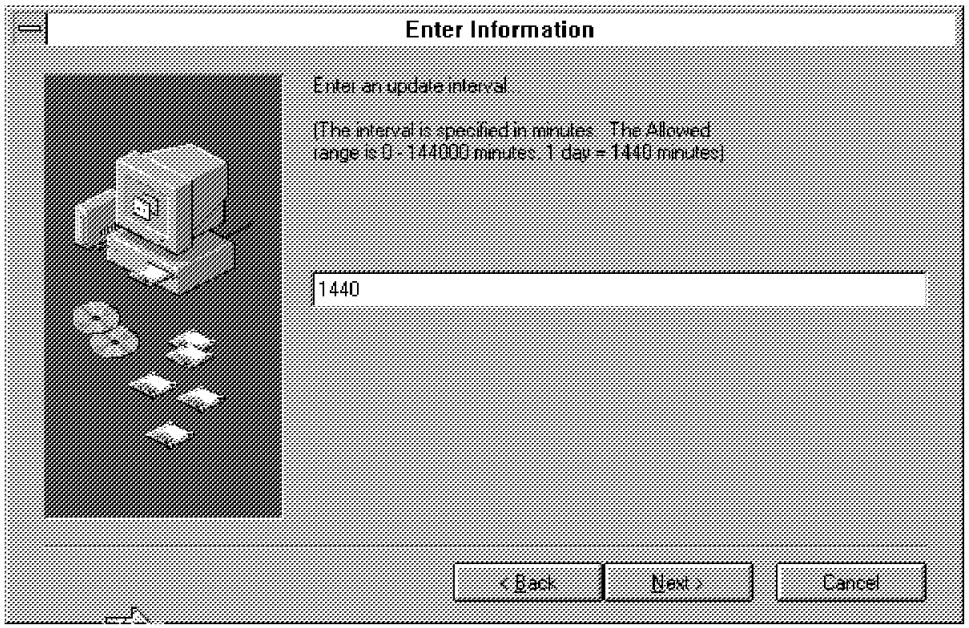


Figure 46. TME 3.0 Agent Installation - Agent Update

16. The installation program now creates the desktop folders on all the Microsoft platforms. Under OS/2 no folder is created. This is a manual process, but can be accessed from the command line. After this the TME 3.0 agent installation is complete.

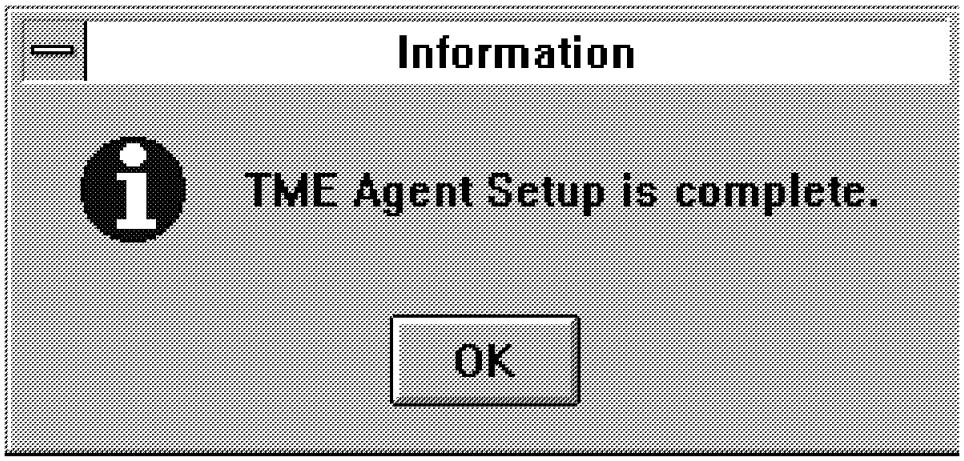


Figure 47. TME 3.0 Agent Installation - Completion Screen

2.5 Installation of Tivoli TME 10 NetFinity Client for NT

For those of you who are a little bit confused about the product names, here is a little history:

- Until May 1996 NetFinity and SystemView were two different products. NetFinity was targeted for small networks and SystemView was for the larger ones.
- SystemView contains NetFinity, some parts of NVDM/6000 and DCAF.

- NetFinity contains monitoring, inventorying and scheduling functions and a simple software (or file) distribution facility.
- In May 1996 NetFinity was named PC SystemView. Some weeks later the new name was TME 10 NetFinity. This product is available as a manager platform for OS/2, NT, Windows 95 and Win 3.11. It is available as a service for all the above platforms as well as NetWare.
- SystemView for OS/2 changed its name to TME 10 NetFinity Server. This product is available as a manager (server) only on the OS/2 platform. Clients are available for NT, Windows95, OS/2, Win 3.11 and NetWare.
- TME 10 NetFinity Server includes all features from TME 10 NetFinity Manager plus the old SystemView for OS/2 functions.
- TME 10 NetFinity Manager and Services include the features of NetFinity Manager and Services 3.05 plus Webability and some alert enhancements.

The following description is about the installation of TME 10 NetFinity Client (the partner product for the TME 10 NetFinity Server) for NT.

2.5.1 Installation

1. You have to start the installation with SETUP.EXE in the client directory or at the first diskette.
2. The first screen is the Welcome screen. It contains the rules of where to install the product.

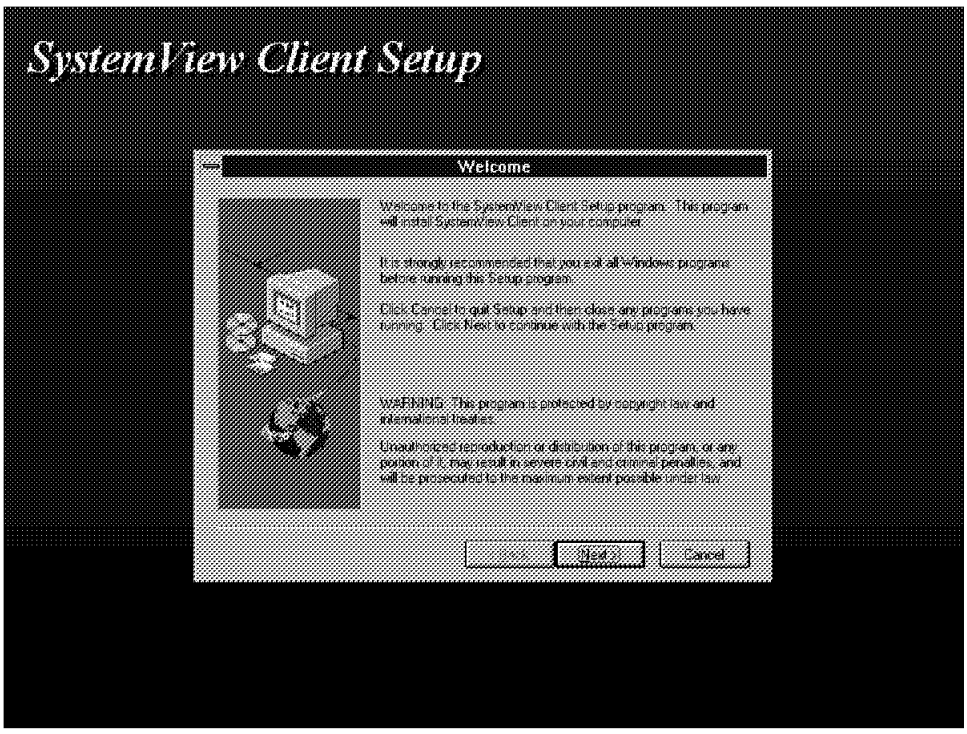


Figure 48. Welcome Screen

3. Selecting the **Next** button leads you to the Destination Location screen. On this screen you have to choose the path for the installation.

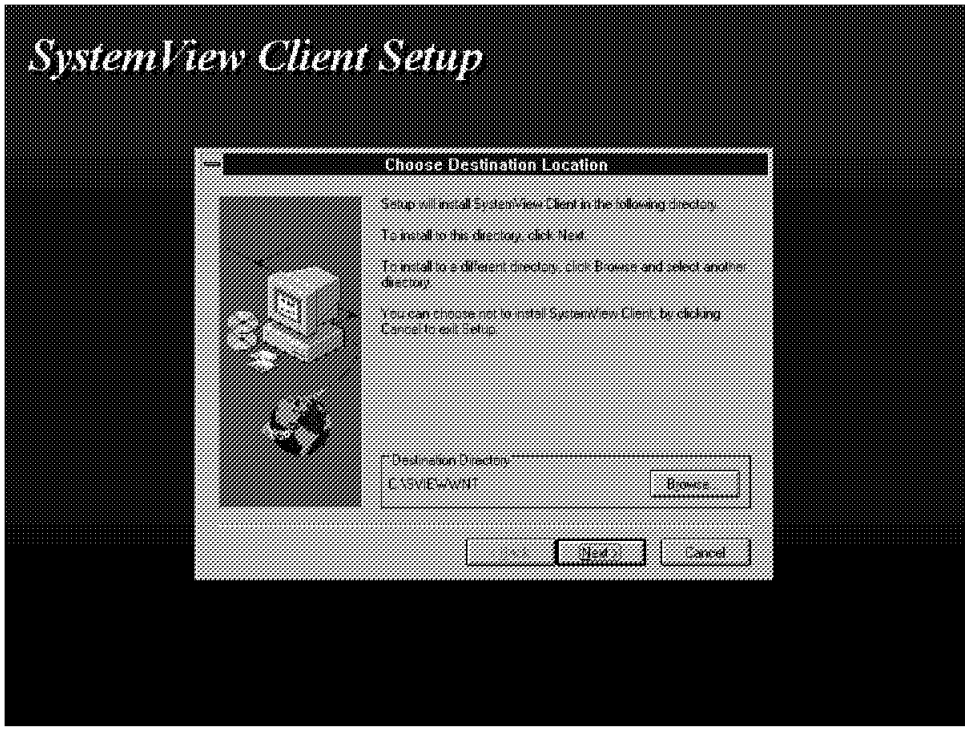


Figure 49. Destination Path

4. The next screen shows the three kinds of installation types:
   - Typical - The most common functions.
   - Compact - The minimum functions.
   - Custom - You select the functions you need.

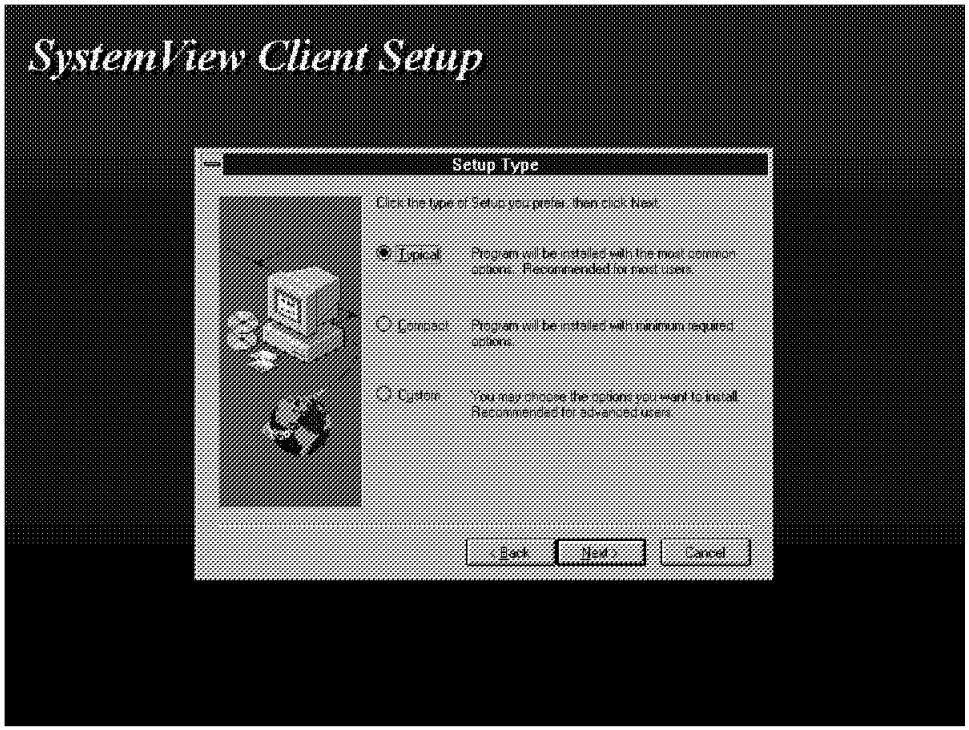


Figure 50. Installation Types

You have to make a selection from one of the installation types.

5. After that, a new screen comes up. At this screen you have to select an existing program folder for the TME 10 NetFinity or the program will create a new one.

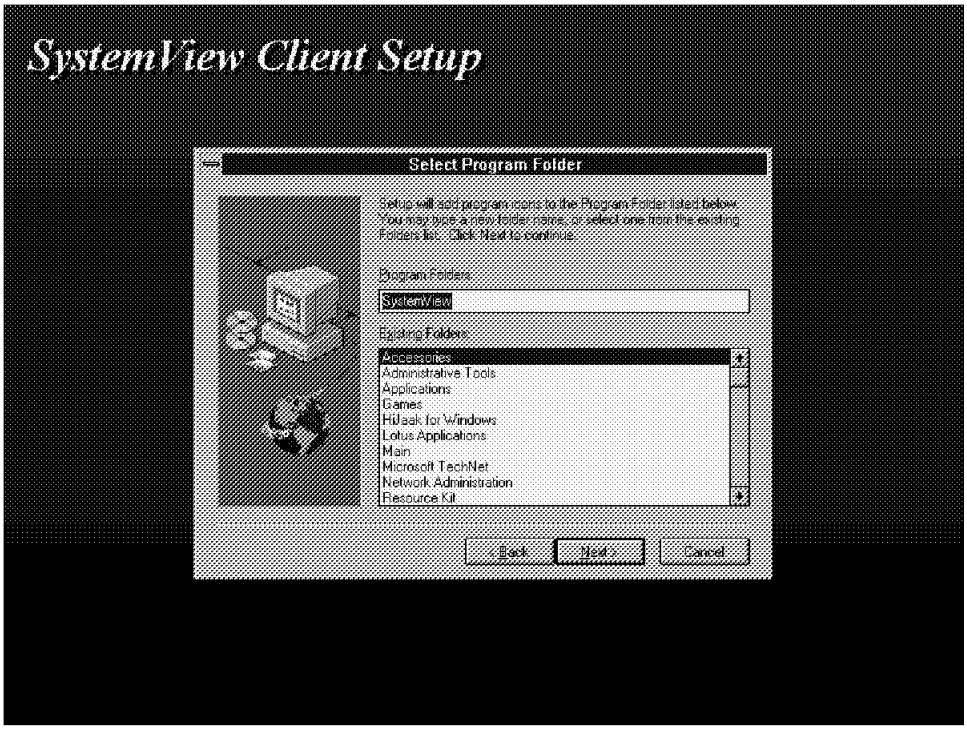


Figure 51. Folder Creation

6. It is possible that the installation program will detect a DLL file that has the same name as the one that it is trying to install. It will always check with you to confirm that it is OK to overwrite the old DLL.

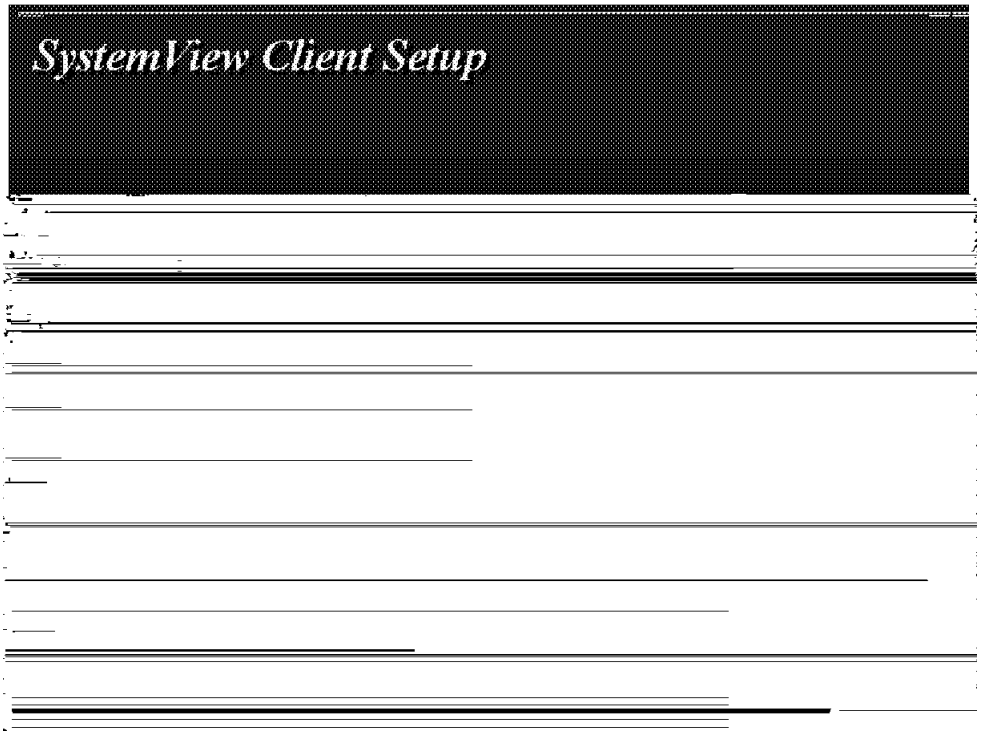


Figure 52. DLL - Pop-up

7. After this pop-up, the installation starts.

Figure 53. Installation Window

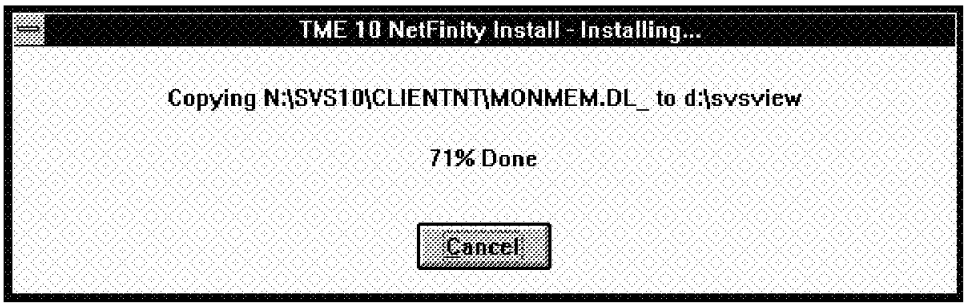


Figure 54. Installation Pop-Up

8. The next step in the installation process is a configuration notebook. There are several fields in this notebook to fill in information about your system:
   - The system name.
   - The network drivers. Be sure to enable the ones you need.
   - System keywords.
   - Options.

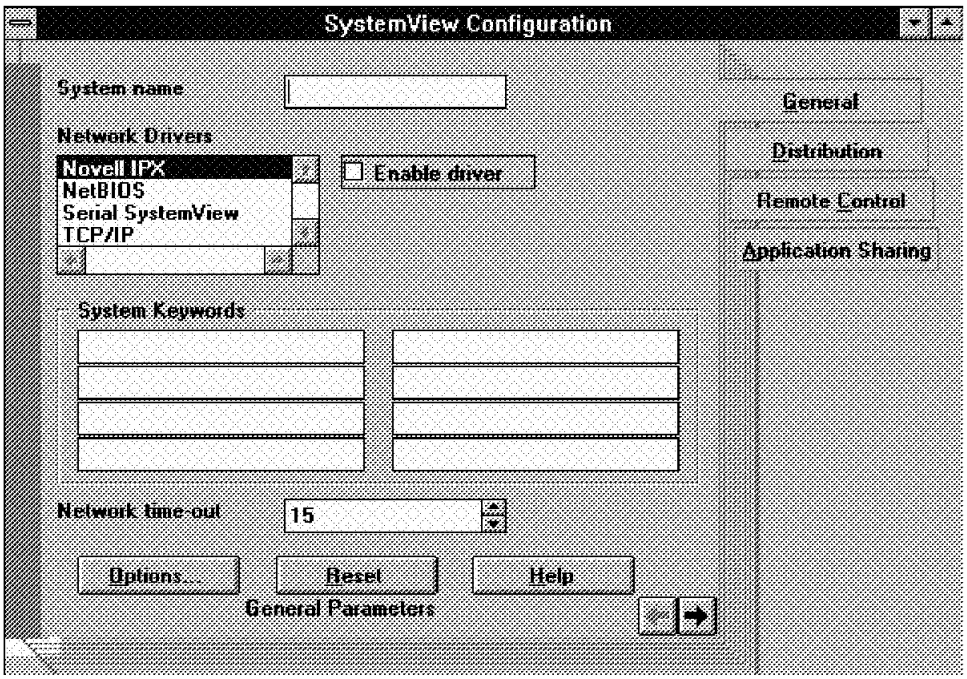


Figure 55. Configuration Notebook - Installation Types

The notebook has four pages. The first page is described above. On the second page you have to fill in some information about directories and the TCP/IP network. Often the default directories are OK. The next page is for remote control and the last page is for application sharing. If you enable application sharing you have to know about the network operating system and remote names and drives.

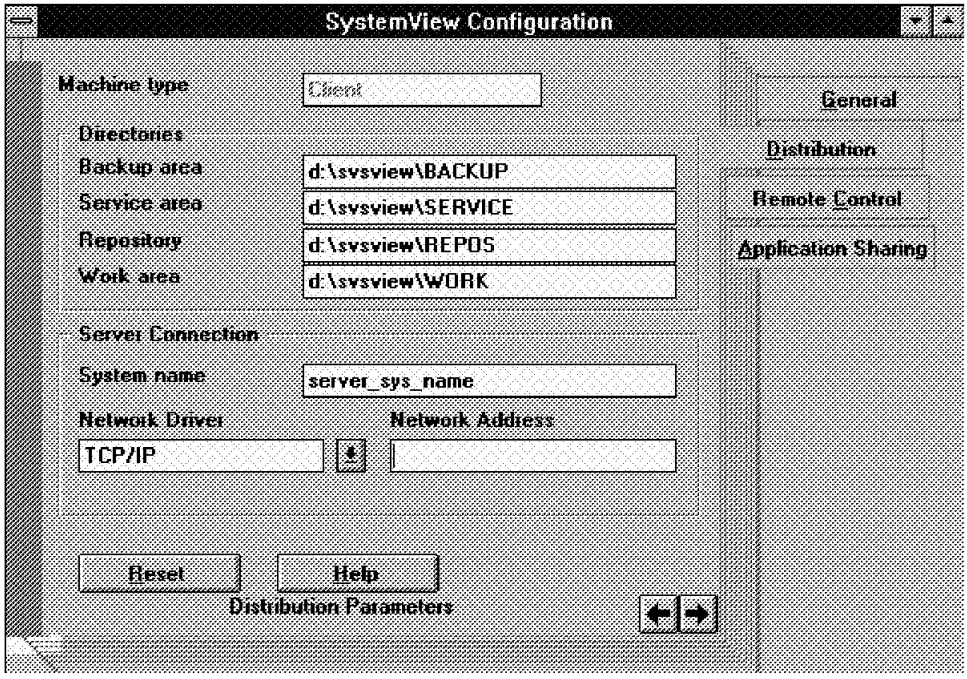


Figure 56. Configuration Notebook - Software Distribution

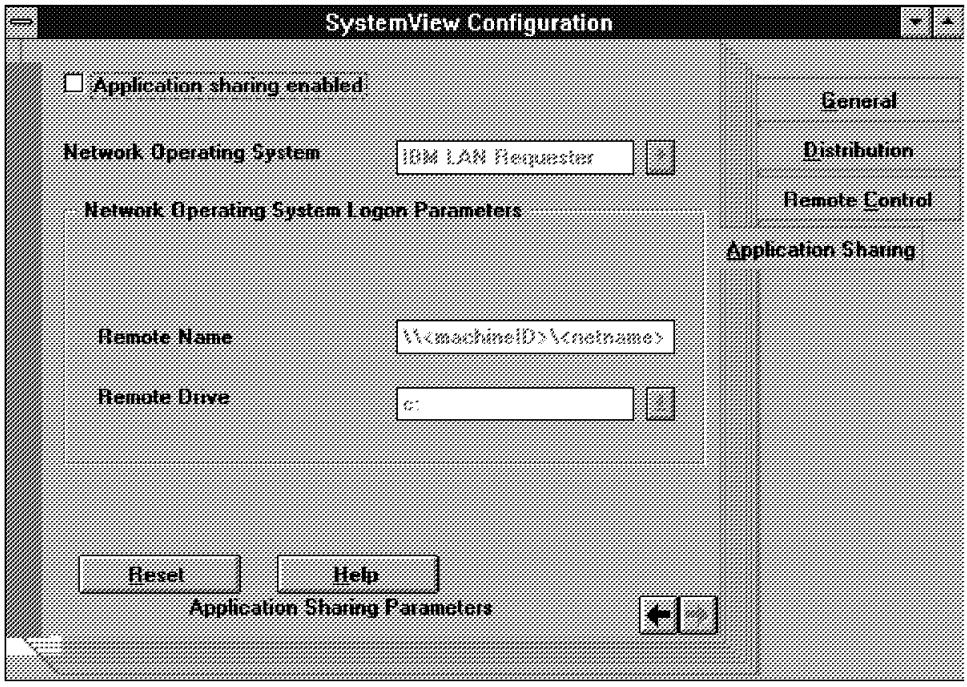


Figure 57. Configuration Notebook - Application Sharing

2.5.2 Sentry Installation

When you install new Tivoli products into the TMR, you perform the installation with the help of the TME Desktop. In the Desktop pull-down menu there is an option called Install. If you choose this option you can:

- Install a product
- Install a patch

To install Sentry you have to choose the **Install product** option. If the current path pointer for your installation media is not correct, you will get the message shown in Figure 58.

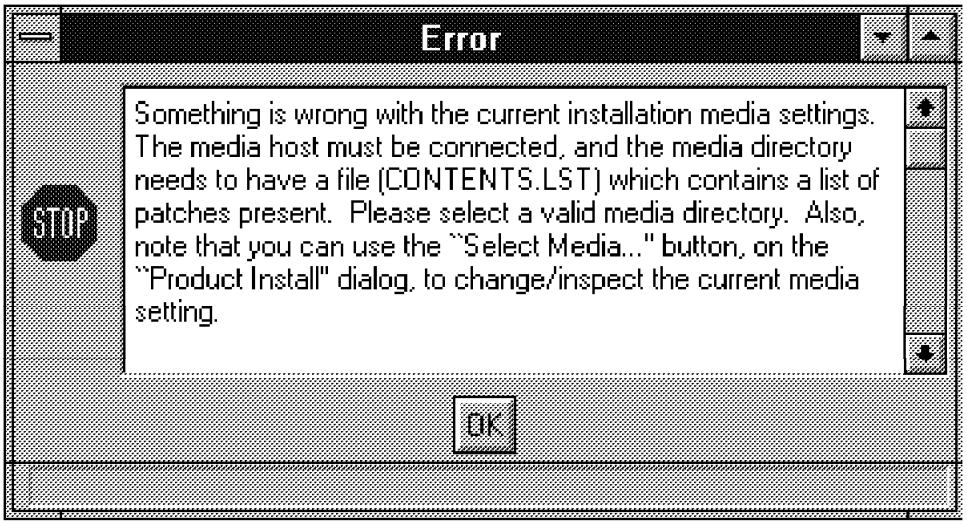


Figure 58. Error Message for Installation

To change the current media settings select **OK** and you will get redirected to the File Browser window, where you can select the correct host, logical drive and directory.

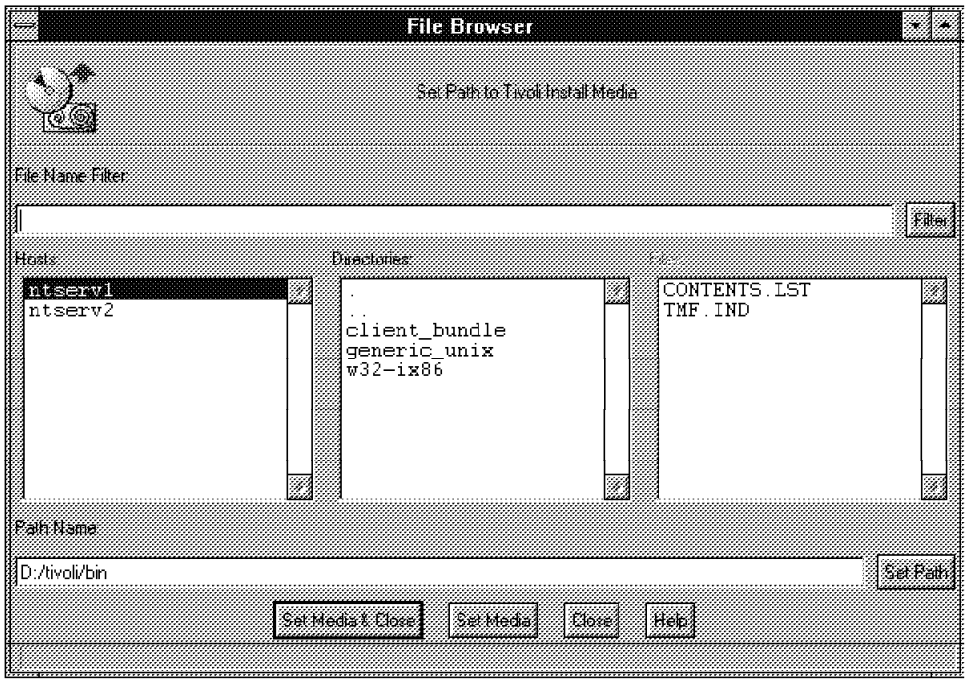


Figure 59. File Browser Window

Once you determine the correct path and update it in the field called Path Name, you just click on the **Set Media & Close** button to confirm the path. If the path is correct, the next window presents an overview of the installable products for Tivoli. In our case, as shown in Figure 62 on page 50, it will be for the Sentry product.

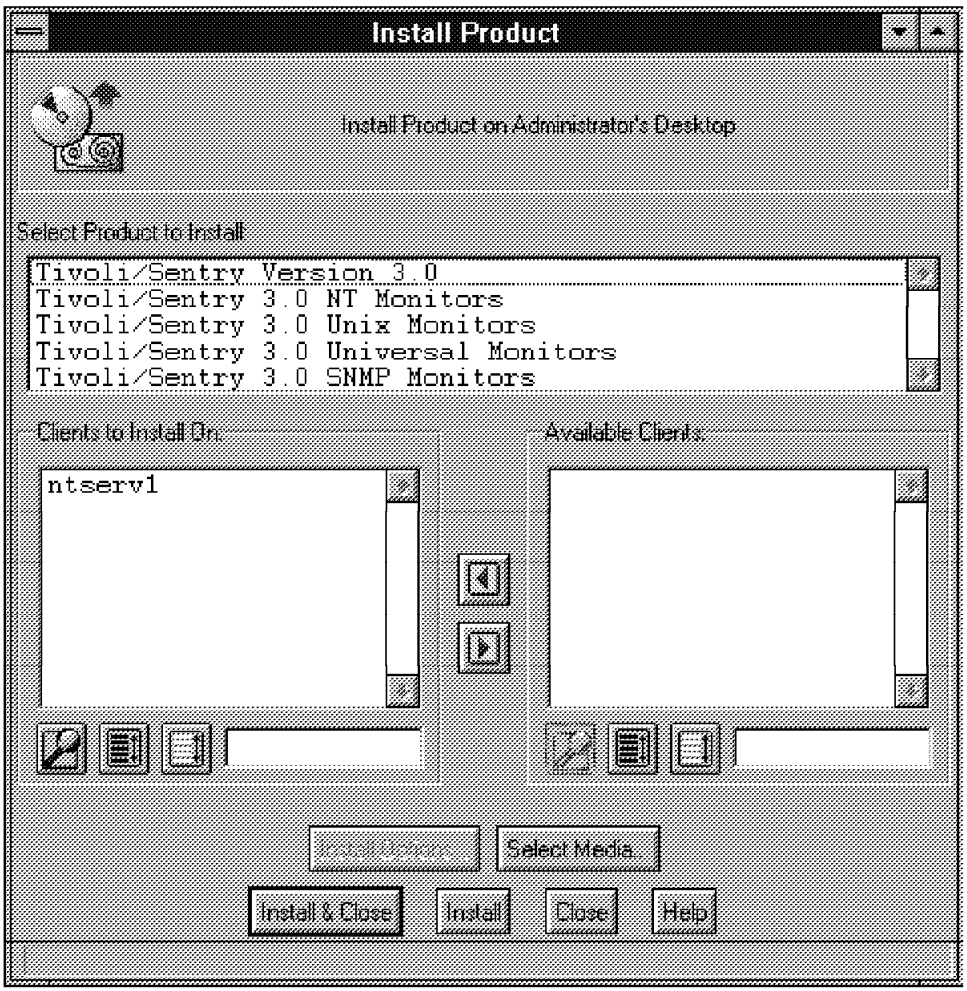


Figure 60. Sentry Installation Selection

Sentry has two components:

- **Sentry** - It has to be installed on each node which will be monitored directly. It monitors the appropriate resources, determines whether the monitor should be triggered, and runs most of the automated responses.
- **Monitors** - A monitor contains code that defines how data about a resource is collected.

The monitors contain the following:

- *NT Monitors* is one of the biggest collections. It includes 26 different collections, each filled with a lot of monitors. You can build three logical monitor collection groups for NT:
  - The Core NT Resource Group, which allows you to monitor objects that are installed on all NT servers and clients, for example logical disk, memory, paging file and a physical disk.
  - The Network Protocol Collection, which allows you to monitor the availability and the use of standard network protocols. This collection includes collections such as IP, ICMP, TCP, UDP and NetBEUI.
  - The Network Resource Collection, which allows you to monitor the NT objects necessary to use the network protocols.

An example of what some of the monitors are follows:

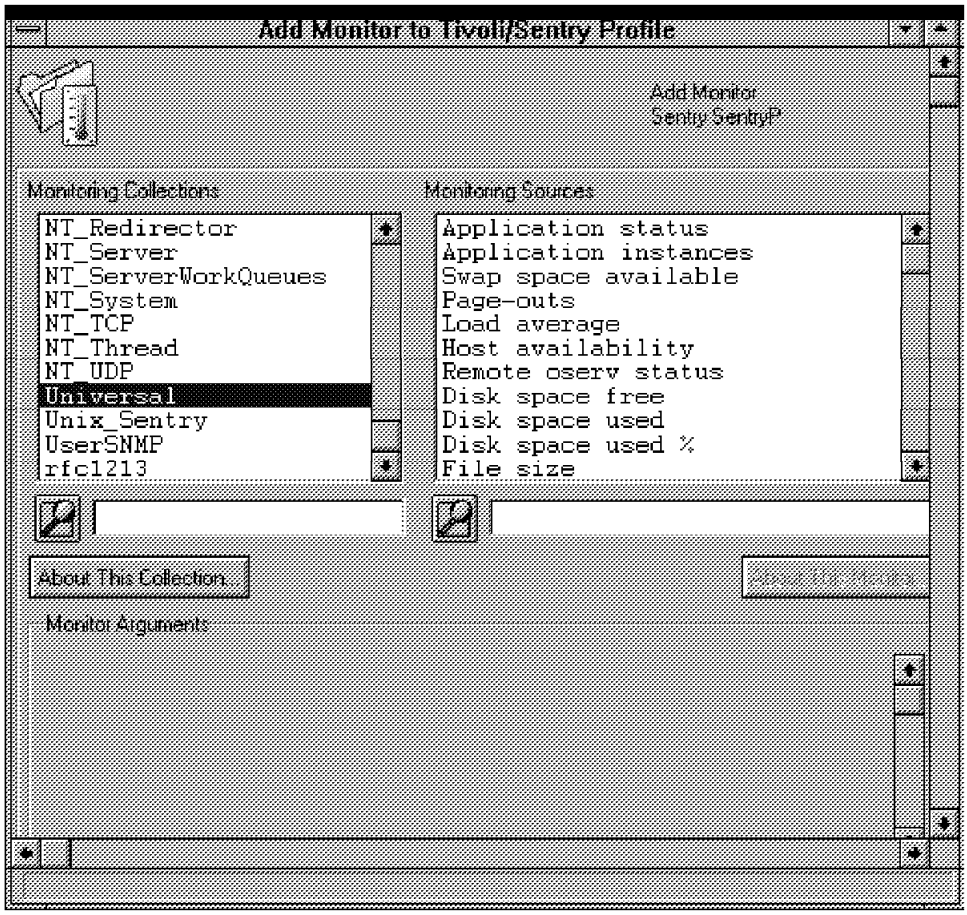


Figure 61. Sentry Installation Selection

- *SNMP Monitors* contain three monitoring collections that allow you to monitor the following types of MIBs (management information bases):
  - Compaq Insight Manager
  - MIB-II (RFC1213)
  - User-specified

  Each of these monitors collections contain a lot of monitors to work with.
- The *UNIX Collection* contains 44 monitors, which are divided into 6 logical groups:
  - Disk Resource Monitoring Sources (for example, Space free, Inodes free Tivoli DB free space, etc.)
  - Security Monitoring Sources (for example, Daemon status, File size)
  - Network Monitoring Sources (for example, Host status, Input packets)
  - System Resource Monitoring Sources (for example, available swap space, Page-outs)
  - Printer Monitoring Sources (for example, Daemon status, Jobs in print queue)
  - User-Defined Monitoring Sources

- *The UNIX-NT Monitoring Sources* contains 19 monitors that are divided into four different groups:
  - Disk Resource Monitoring Sources (for example, Disk space free or used).
  - Security Monitoring Sources (for example, application instances, file checksum)
  - System Resources Monitoring Sources (for example, host availability and swap space available)
  - User-Defined Monitoring Sources
- *Tivoli RFC 1213 Monitoring* runs on UNIX, Windows NT, Windows 95 and Novell Systems with SNMP agents. You have five different monitor collections to choose:
  - System Monitoring Sources (for example, Host Description, Host name)
  - Interface Monitoring Sources (for example, Bytes received, receive errors)
  - Internet Protocol Monitoring Sources (for example, IP Transmit Packets, IP Received Packets)
  - Transmission Control Protocol Monitoring Sources (for example, Maximum TCP connection)
  - User Datagram Protocol Monitoring Sources (for example, UDP NO-Port Errors)

For further information please refer to the special monitor guides delivered with the Sentry product package.

You have to choose each product that you want to install. You can′t select multiple products at a time to install, so you will need to come back to this installation window after each of the individual functions are installed. For Sentry, you must select the **Tivoli/Sentry Version 3.0** product. Then you have to choose the client where you want to place Sentry. In the right list box you see a list of available clients. On the left side is the list of clients that the install will actually occur on. You can select new clients from the right box and add them to the installation list.

After this you have to select the **Install & Close** button. The installation starts. In an overview window you will see the progress. First you will get asked to confirm the installation of the product function. It will indicate what the implications are of installing the product in terms of what will get copied onto the client. To continue, choose the **Continue Install** button.

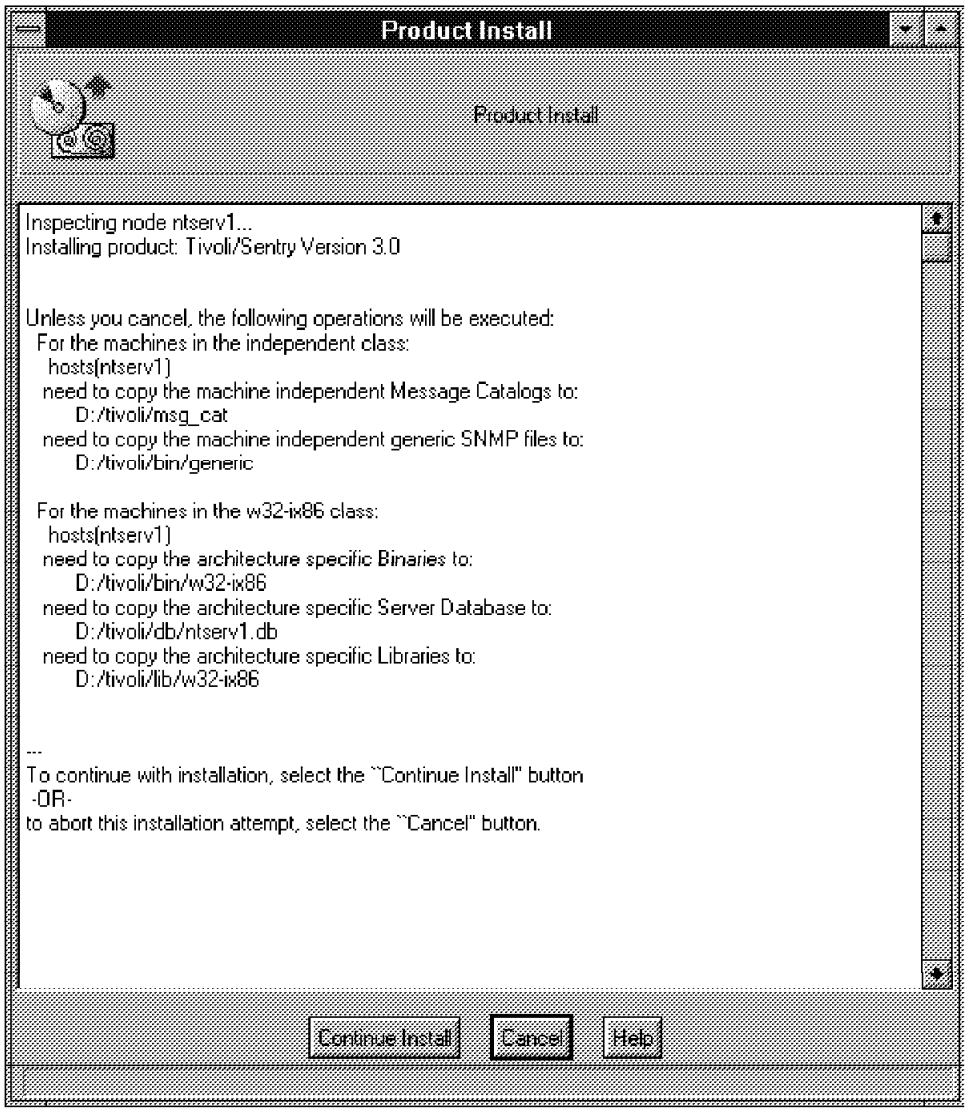


Figure 62. Installation Window I

After the installation starts the window shows the progress.

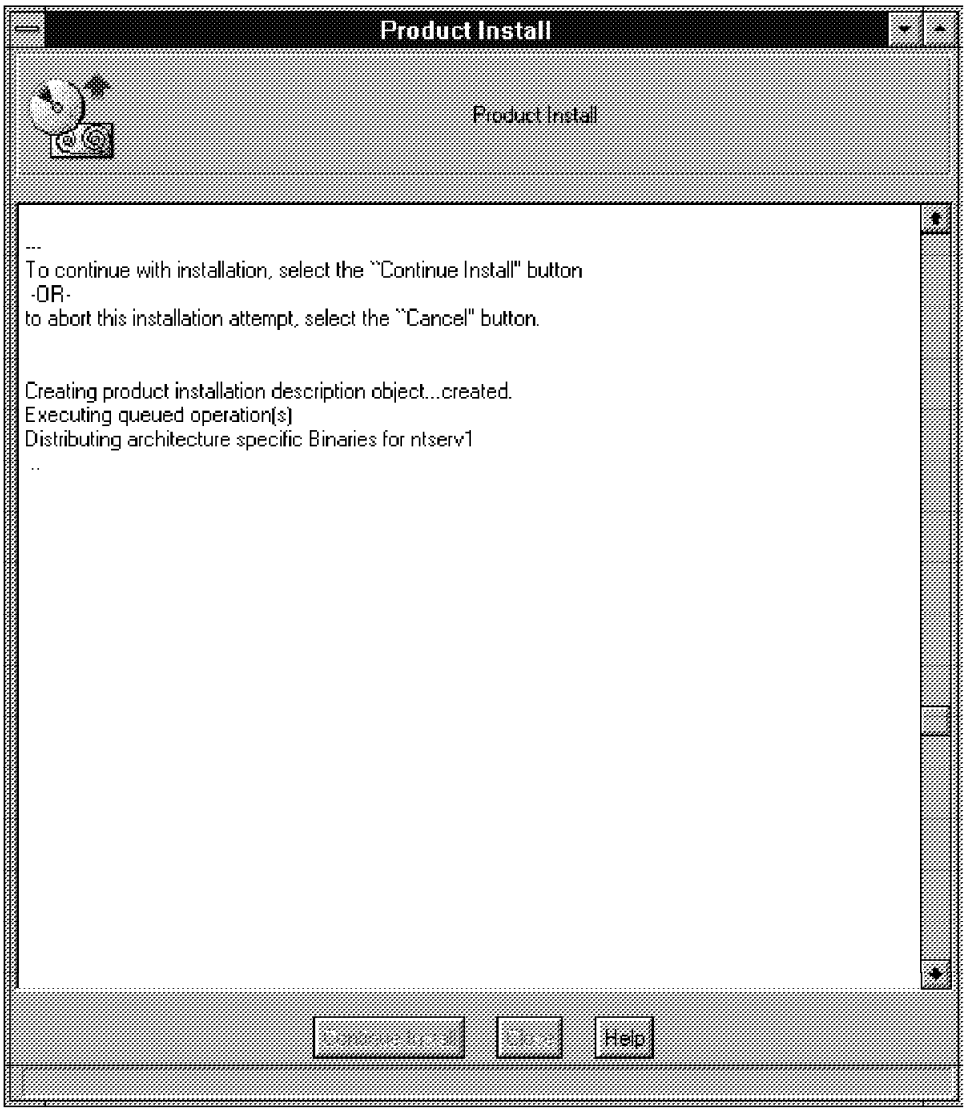


Figure 63. Installation Window II

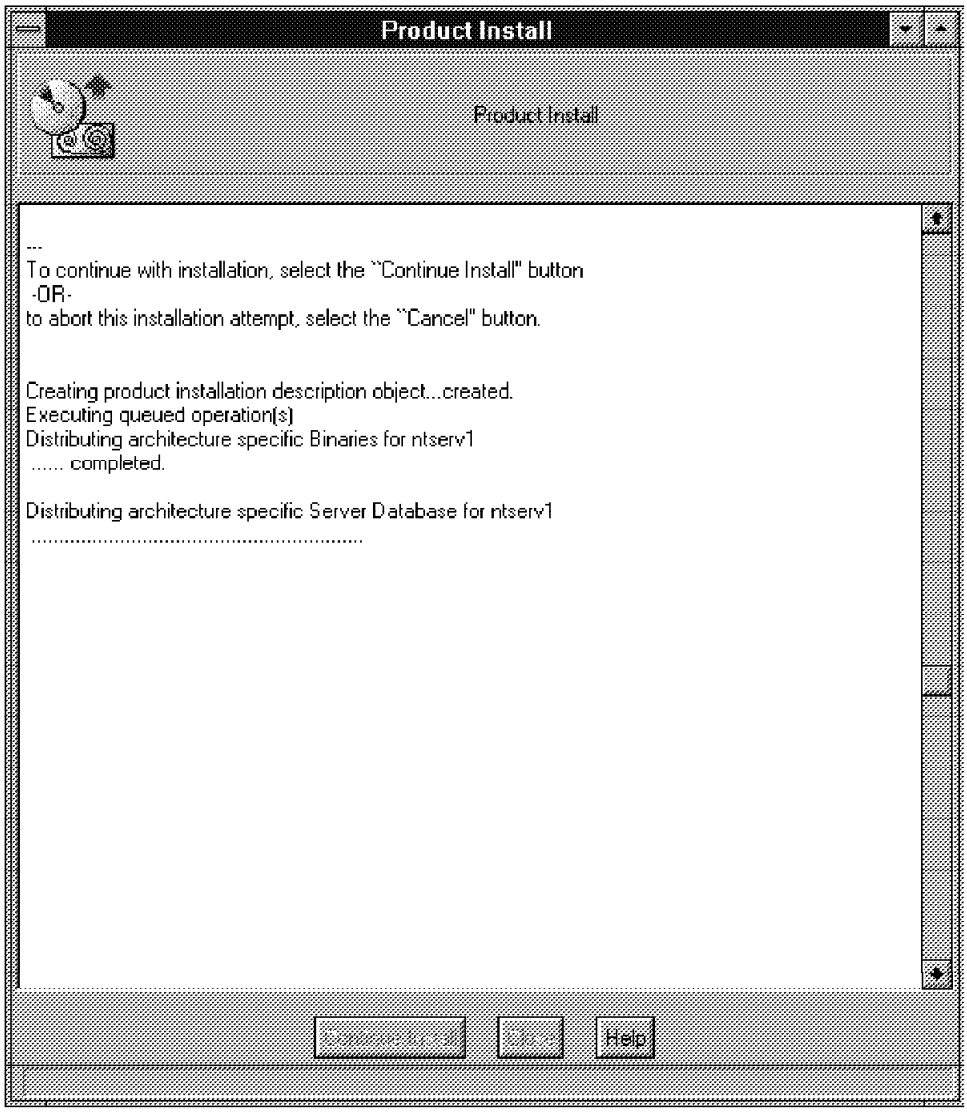


Figure 64. Installation Window III

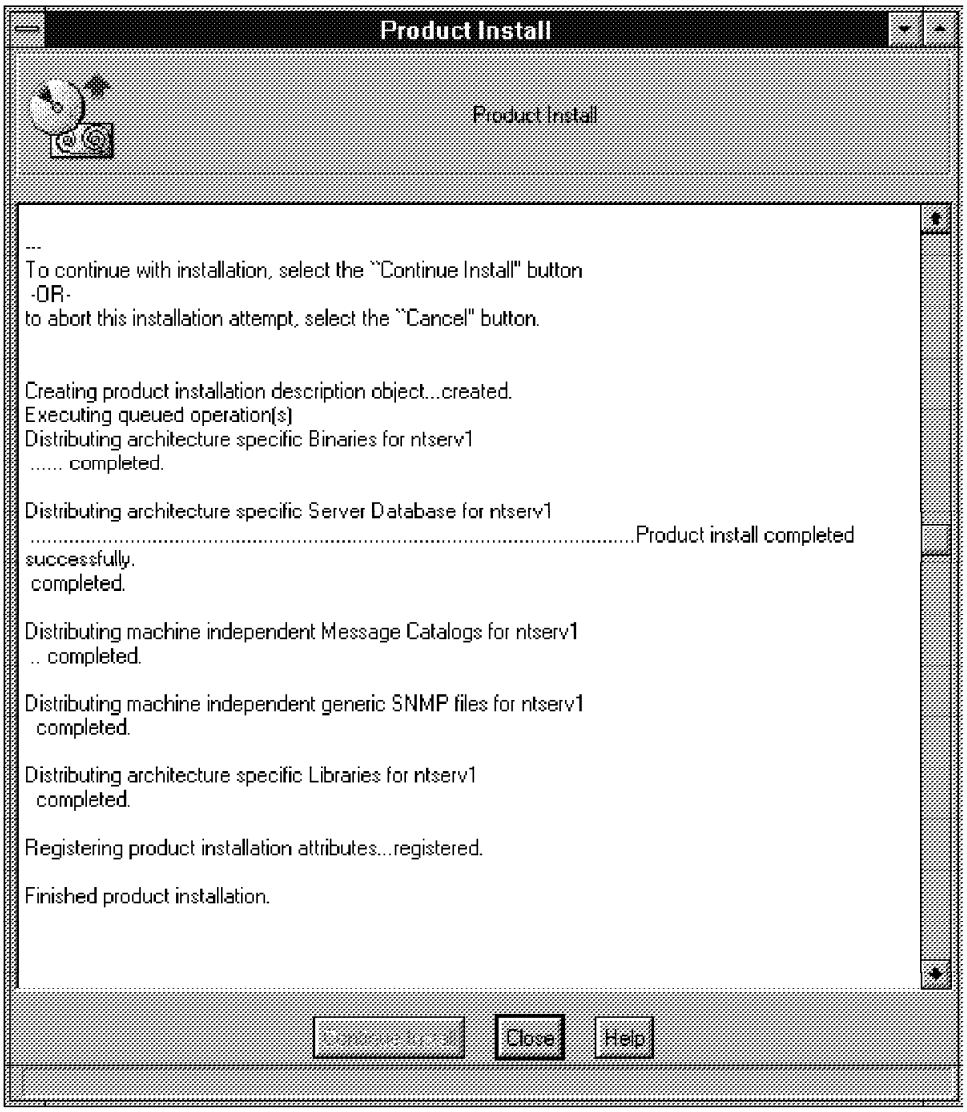


Figure 65. Installation Window IV

After the installation has finished you can install the different monitors for Sentry. You do that the same way as the product installation itself. The installation windows for the monitors are shown in Figure 66 on page 54 and Figure 67 on page 55.

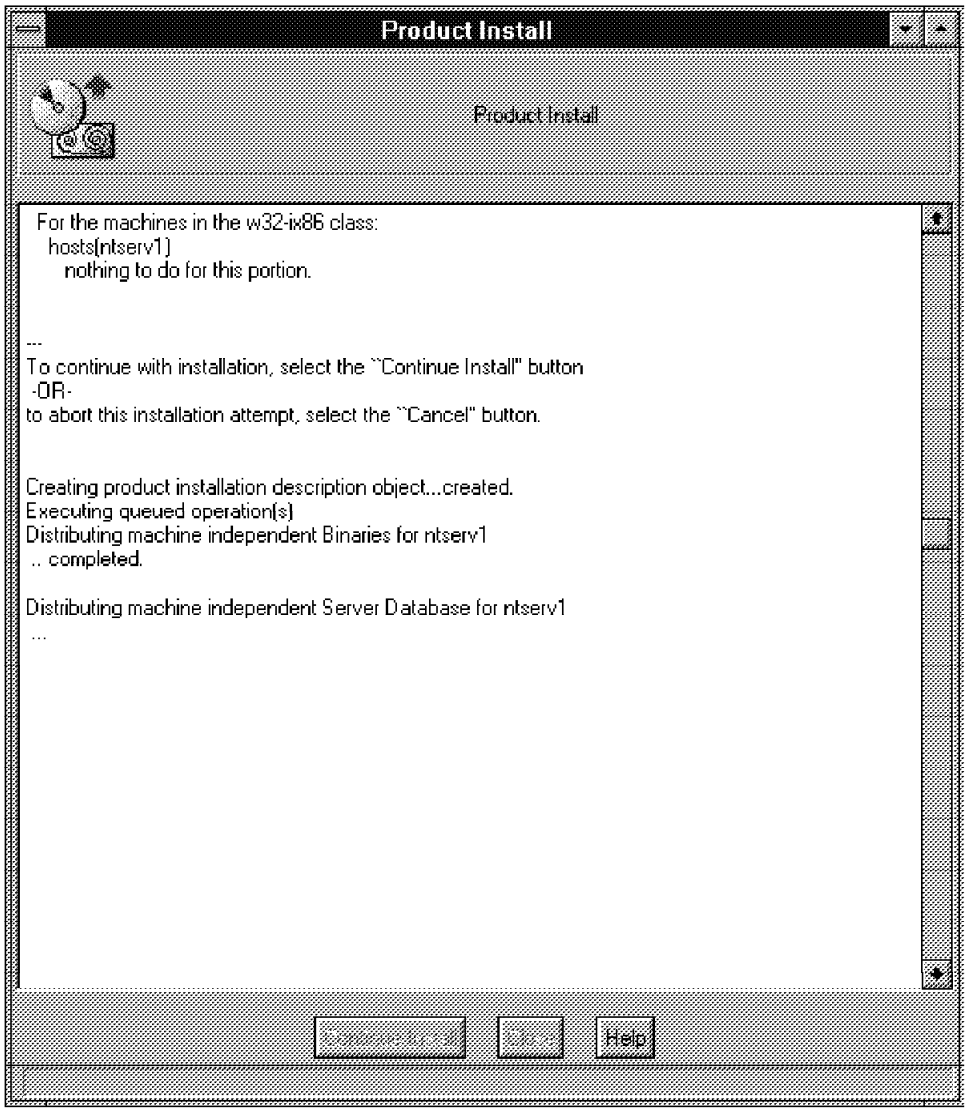


Figure 66. Monitor Installation I

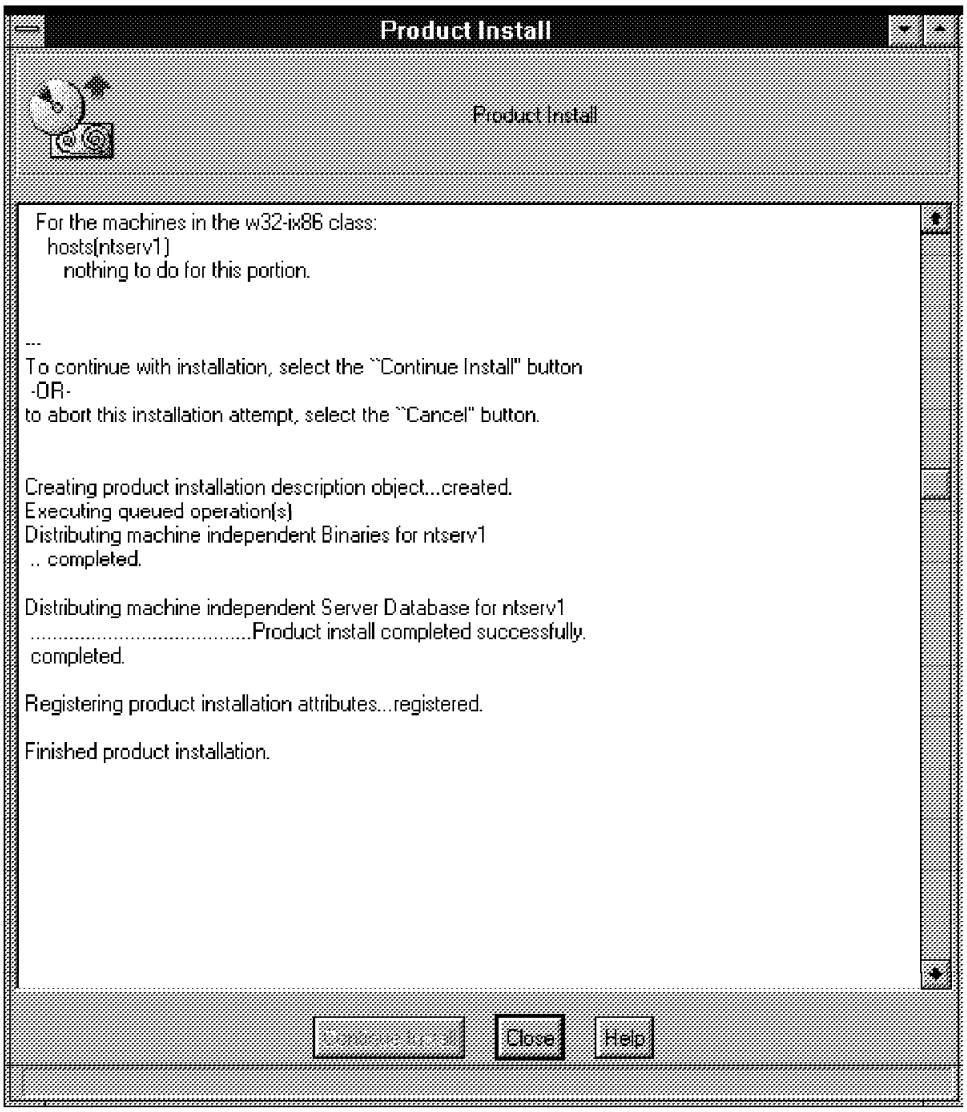


Figure 67. Monitor Installation II

Chapter 3. Connecting Tivoli Management Regions - TMRs

This chapter shows the possible ways to connect TMRs, both on a single platform and across different platforms.

A requirement for most organizations is to manage many resources across a large number of locations, to arrange their systems management facilities across many networks or to subdivide management between different types of user groups. An example might be to divide general users and high profile specialist groups. The TME achieves all of the above by providing the capability to split enterprise networks into structured Tivoli Management Regions.

A Tivoli Management Region (TMR), essentially, is a TME server and the clients which it serves. Connecting TMRs enables management of resources defined in remote TMRs from connected TMRs.

3.1 Connecting TMRs

In this section we discuss:

- Properties of TMRs
- Possible ways of connecting TMRs
- Why we connect TMRs
- Connecting platform-independent TMRs
- Updating resources amongst connected TMRs

When TMRs are connected, there is an initial exchange of information between the connected TMRs. The information exchange includes an update of names and object identifiers. These resource updates can occur immediately or can be scheduled, and should be performed at relevant intervals so as to reflect the volume of changes that occur within the complete TME network. An example of a change would be the addition or removal of a number of Managed Nodes.

The update of resources across TMRs is always a pull operation from the remote TMR to the local pulling TMR. The pull occurs only on those resource types managed within the pulling TMR. There can be times when the local TMR does not have the exact resource types as the remote TMR resources being pulled.

3.1.1 TMR Properties and Capabilities

There are a number of properties and capabilities associated with TMRs:

- Name Registry - Each TMR has an associated name registry, acting as a region-wide naming service which maps resource names to resource identifiers as well as relevant information.
- Resource Types - Within the name registry for each TMR, there are several possible resource types. Examples of resource types can be seen in the left-hand column of Figure 68 on page 58.
- Instances - Within the resource types contained in the name registry for each TMR, there can be instances of each resource type. Examples of the instances of the resource type Policy Region can be seen in the right-hand

column of Figure 68 on page 58. These are made available by checking the resource type box on the left side.

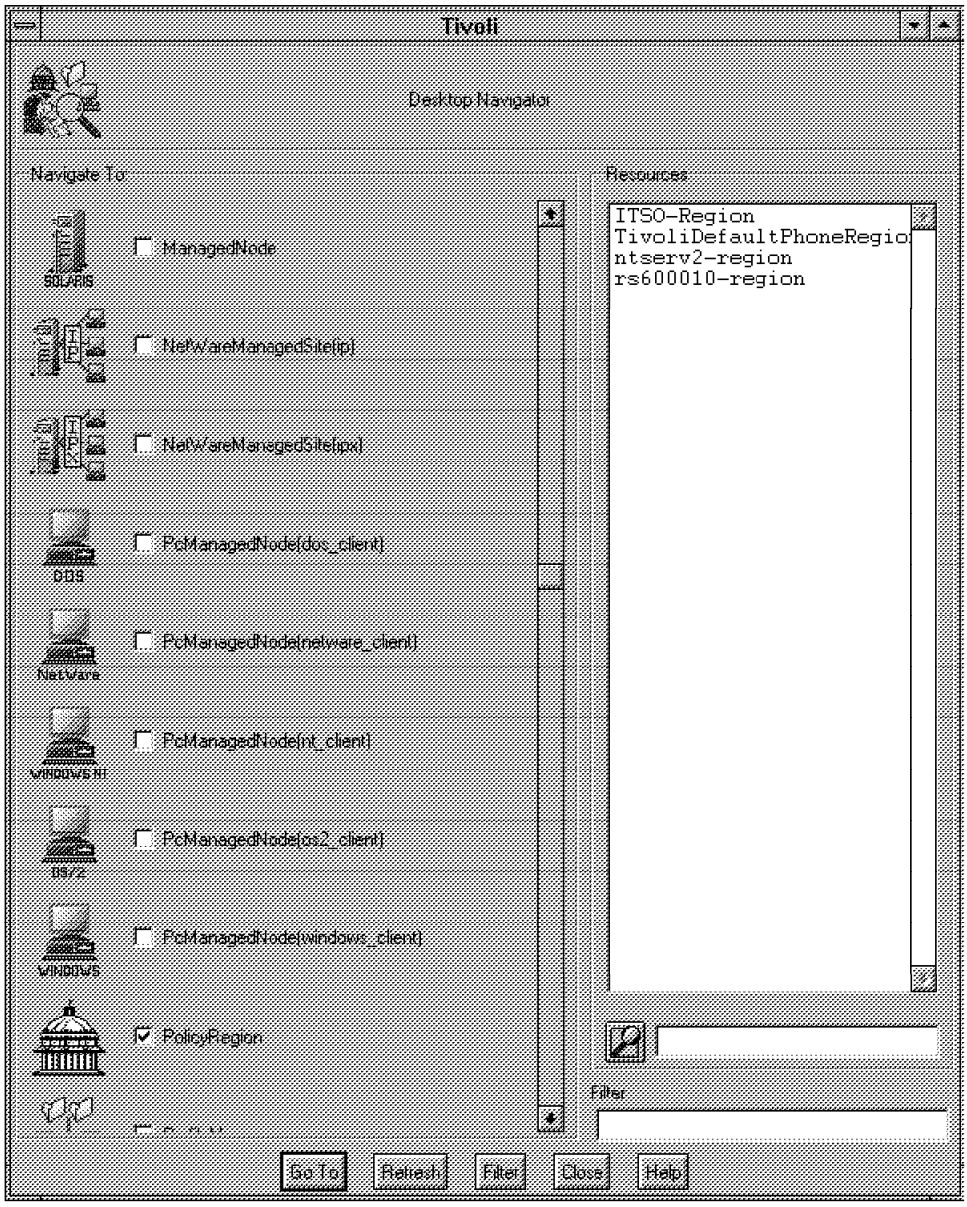


Figure 68. TMR Inter-Connections - Resources and Instances

- Connecting TMRs - It is possible to connect TMRs to allow the sharing of information and resources.
- Information Exchange - Connected TMRs can exchange and update information only with TMRs to which they are directly connected. Therefore the connection of TMRs does not support hierarchical inter-connection rules.

  A complete list of what resources are and what resources are not exchanged between TMRs, when an update is performed follows. These resources are relevant only to the TME platform.

Table 1. Platform Resources Exchanged at Resource Update	
Resource	**Description**
Administrator	Name registry list of administrators and logins
AdministratorCollection	List of administrators as seen from the administrators collection view
Job	Name registry list of task library jobs
ManagedNode	Name registry list of managed nodes
PcManagedNode	Name registry list of PC Managed Nodes
PolicyRegion	Name registry list of Policy Regions
ProfileManager	Name registry list of profile managers
Repeater	Name registry list of repeaters
TMF_Notice	Name registry list of task libraries
TaskRepository	Definitions of tasks
TopLevelPolicyRegion	List of Top Level Policy Regions as seen from the Top Level Regions collection view

Table 2. Platform Resources Not Exchanged at Resource Update	
Resource	**Description**
ActiveDesktopList	A list of currently active desktops
Classes	A list of classes with friendly names in the local TMR
Distinguished	A list of distinguished (one-of-a-kind) objects in the local TMR
PatchInfo	A list of patch info objects in the local TMR
Presentation	A list of presentation objects in the local TMR
ProductInfo	A list of product info objects in the local TMR
ProfileEndpoint	An abstract resource type that contains no instances, but is used by wgetallinst to look up resources that are profile endpoints
Scheduler	The Scheduler object for a TMR

3.1.2 TMR General Issues

When utilizing multiple TMRs, which is likely within most Tivoli installations, certain guidelines should be followed.

These include the location of the TME database files within each TMR. These should be located on a file system local to that TMR′s TME server. This is true whatever the underlying environment (AIX, NT, etc.). The database files should never be made accessible through NFS mounts, since the TME database contains information relevant to the TMR it represents.

Indeed, for better performance, it is recommended that all files should remain local, including binary and library files as well as the TME database files. It should therefore be considered in advance, the amount of local file system storage the TME server has.

3.2 TMR Connections

A TMR connection requires that each TMR has a unique name among all the TMRs to which it is connected, be it a direct or indirect connection. Connecting TMRs with nonunique names, either directly or indirectly, will result in a failed connection attempt. Further structural considerations are discussed in 3.3, “Structuring Connected TMRs - Possible Scenarios” on page 71.

When connecting TMRs, information regarding the TME server name, region number, encryption level and the remote TMR password are required.

- TME Server Name - The name of the remote TME server with which the connection is intended.
- Region Number - The number of the remote TMR with which the connection is intended.
- Encryption Level - The level of password encryption to be used for the connection. It is recommended that the encryption levels remain consistent between connecting TMRs, although this is not mandatory. There are three different encryption levels, None, Simple and DES.
- Remote TMR Password - The remote TMR password.

Connections between TMRs are initiated either from the TME Desktop or from the command line interface.

3.2.1 Connection Types

Here we discuss the different types of connections that can be performed between TMRs as well as the configuration options available.

3.2.1.1 Remote Connections

A remote connection between two TMRs is the least secure of the two possible connection types. These connection types need only be initiated at one of the TMR servers performing the connection. Access to the desired TMR is granted one of two ways:

1. By remote login access or shell access.
2. By the Trusted Host Facility.

The following panel is presented when a remote TMR inter-connection is desired.

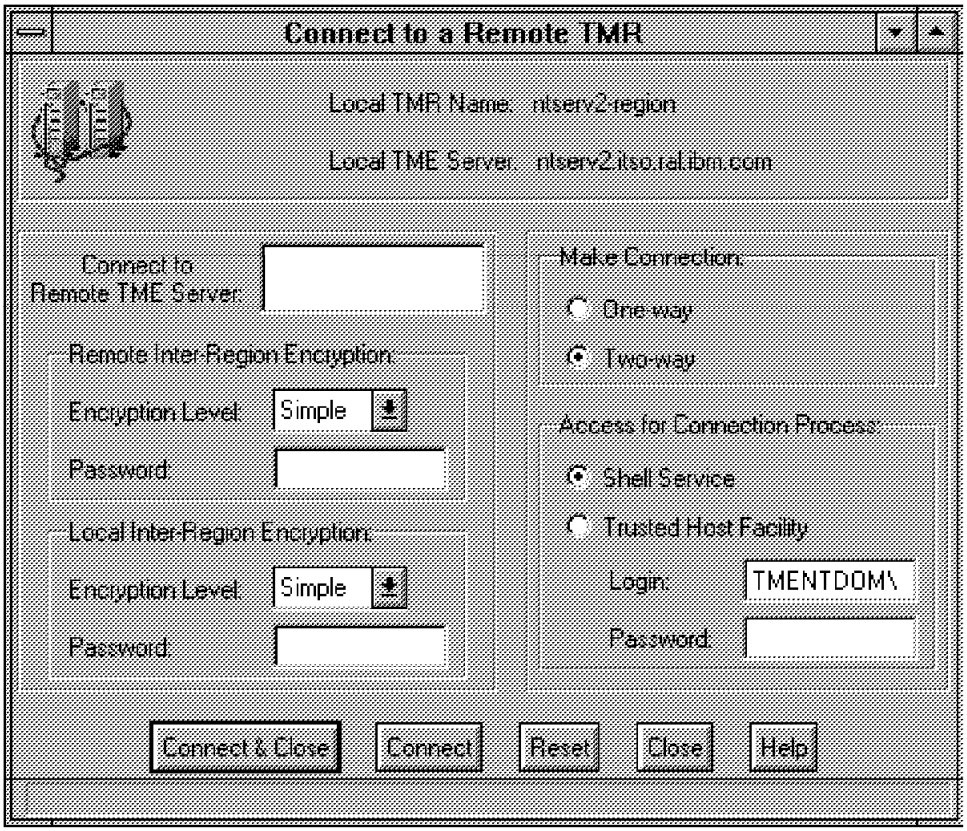


Figure 69. TMR Inter-Connections - Remote TMR Connections

Using the Shell Service to achieve a remote connection requires a login name and password for the remote server, where the remote TMR resides. As can be seen, this involves sending a root password across the network.

Using the Trusted Host Facility requires that the remote server has enabled access for the connecting server, such that the connecting server can open up a remote shell on the remote server.

3.2.1.2 Secure Connections

Creating a secure connection, requires actions to be performed at both of the TME servers wishing to connect their TMRs. This type of connection only becomes valid when the connection has been defined locally on both machines. These secure connections are achieved by using region numbers.

TMR numbers are obtained by running the command line utility odadmin at each server within the TMR connection as shown in Figure 71 on page 62. We did not write a shell to automatically do this but it is conceivable to write a shell to extract the information from each odadmin command and feed it into a shell that would issue the connect commands. This would obviously require the ability to be able to do commands like TCP/IP REXEC, or have the output placed in a file that was accessible from all systems.

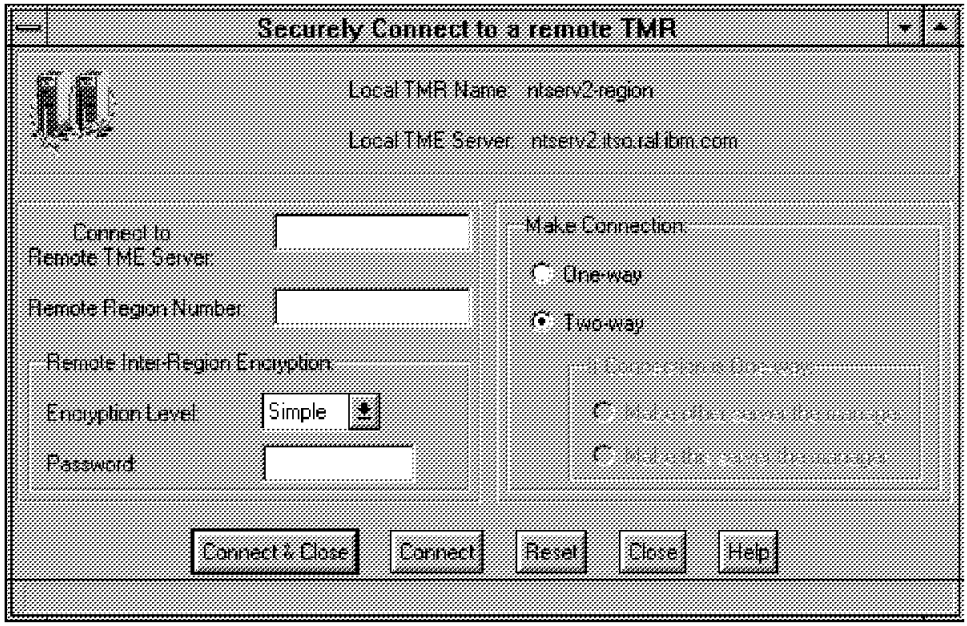


Figure 70. TMR Inter-Connections - Secure TMR Connections

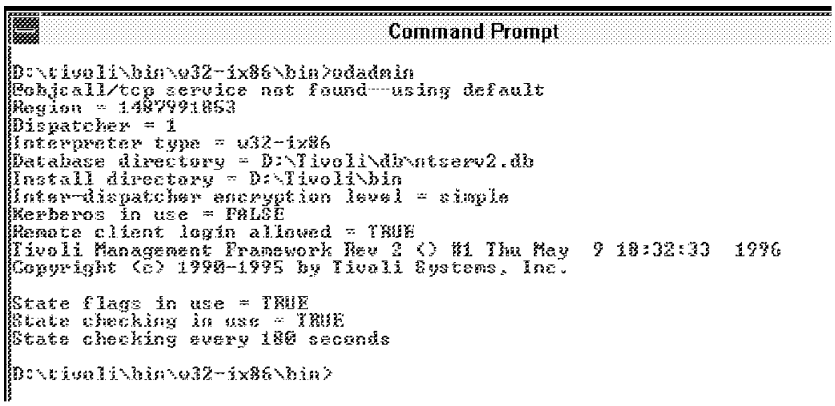


Figure 71. TMR Inter-Connections - Obtaining TMR ID Numbers

Entering an invalid number for the remote region will result in a null connection between the two TMRs. In essence this will result in a connection between the two TME servers, but not between the two intended TMRs. A null connection can be identified by looking at the following two Connection Status panels.

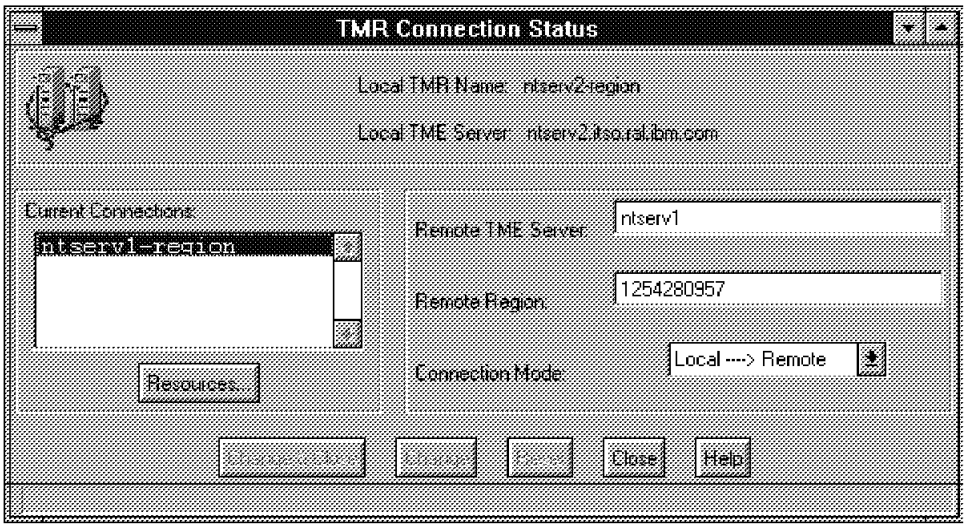


Figure 72. TMR Inter-Connections - Correct TMR Connection

Figure 72 is a result of clicking on **List Connections** from the Desktop/TMR Connections pull-down menu. The panel displays the connection to ntserv1-region, which is the actual name of the remote TMR. In the following panel, we can see displayed under Current Connections ntserv1, which refers to the remote TME server with which we intended to connect. The connection between servers succeeded, but the connection between regions on these servers failed, resulting in a null connection.

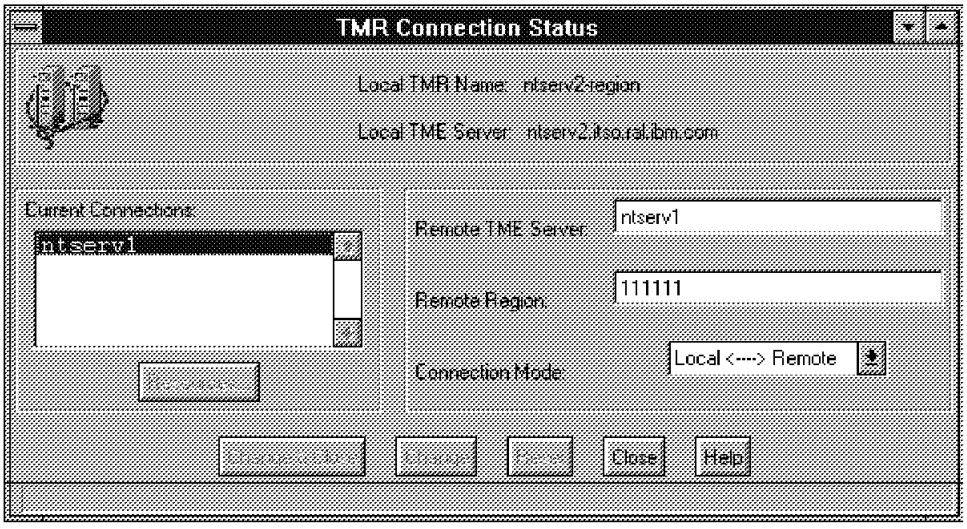


Figure 73. TMR Inter-Connections - Incorrect TMR Connection

Although there are secure and nonsecure connection routes available, once two TMRs are inter-connected, regardless of how that connection was established, there is no difference in functionality.

3.2.2 One Way Connections

It is possible to connect TMRs. such that only one TME server manages the resources which the inter-connection provides. One-way TMR connections can be achieved across either a normal remote connection or a secure connection.

In a one-way connection only one TMR has access to the information about all the resources which the TMR inter-connection provides. The TMR with access to all the information is known as the managing TMR, while the other TMR partner is known as the managed TMR.

Creating a one-way remote connection is achieved by choosing **Desktop** and **Connections** from the TME desktop on the server establishing the connection. Since it is only possible to create this one-way connection from the local server, this server should have been selected earlier.

Creating a one-way secure connection is achieved by choosing **Desktop** and **Secure Connections** from the TME desktop on either server involved in establishing the connection. Since it is a secure connection, it is irrelevant which server chooses to initiate the connection.

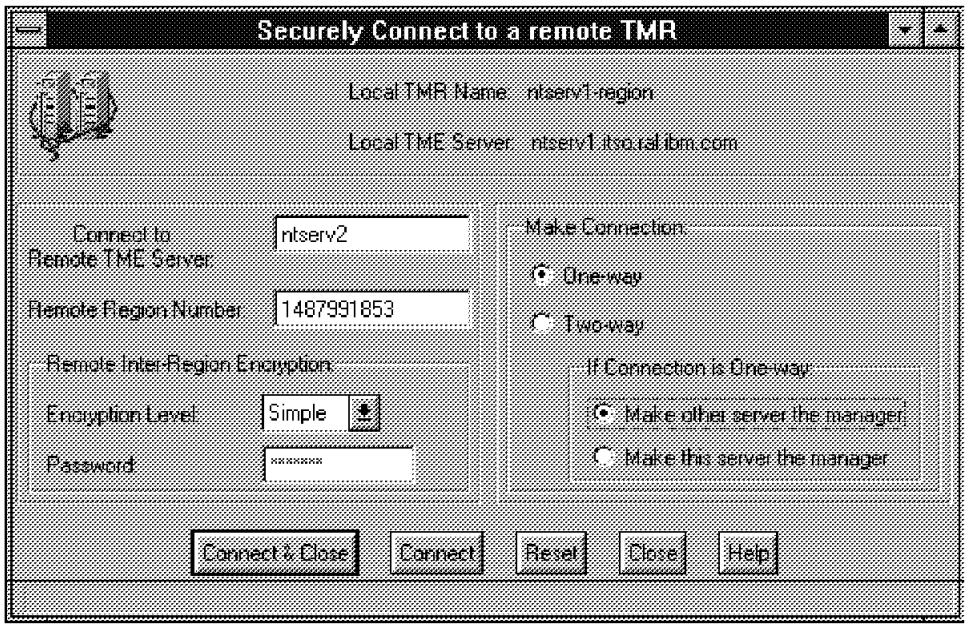


Figure 74. TMR Inter-Connections - Secure One-Way Managed TMR

Figure 75. TMR Inter-Connections - Secure One-Way Managed TMR Confirm

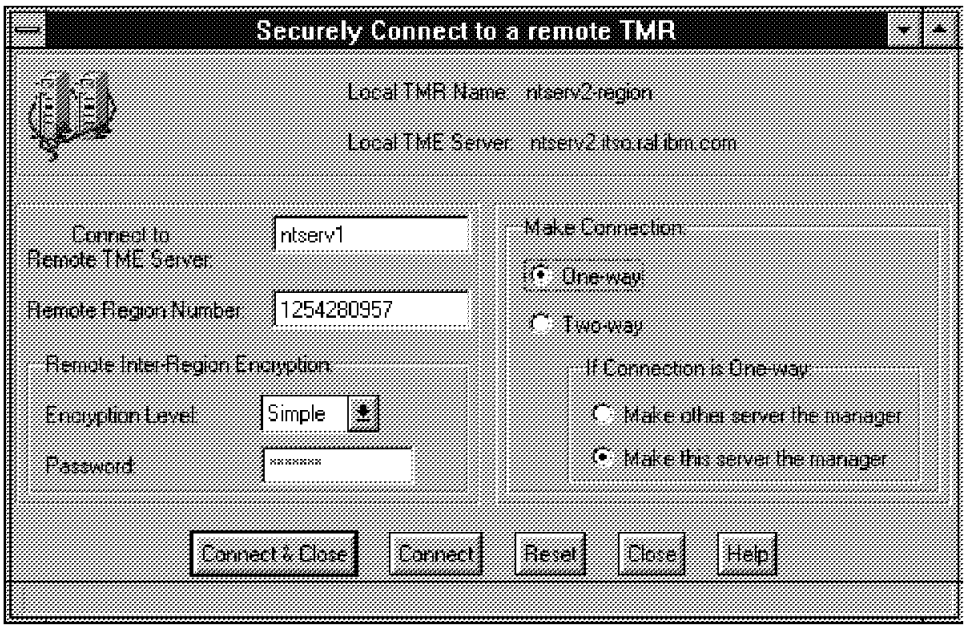


Figure 76. TMR Inter-Connections - Secure One-Way Manager TMR

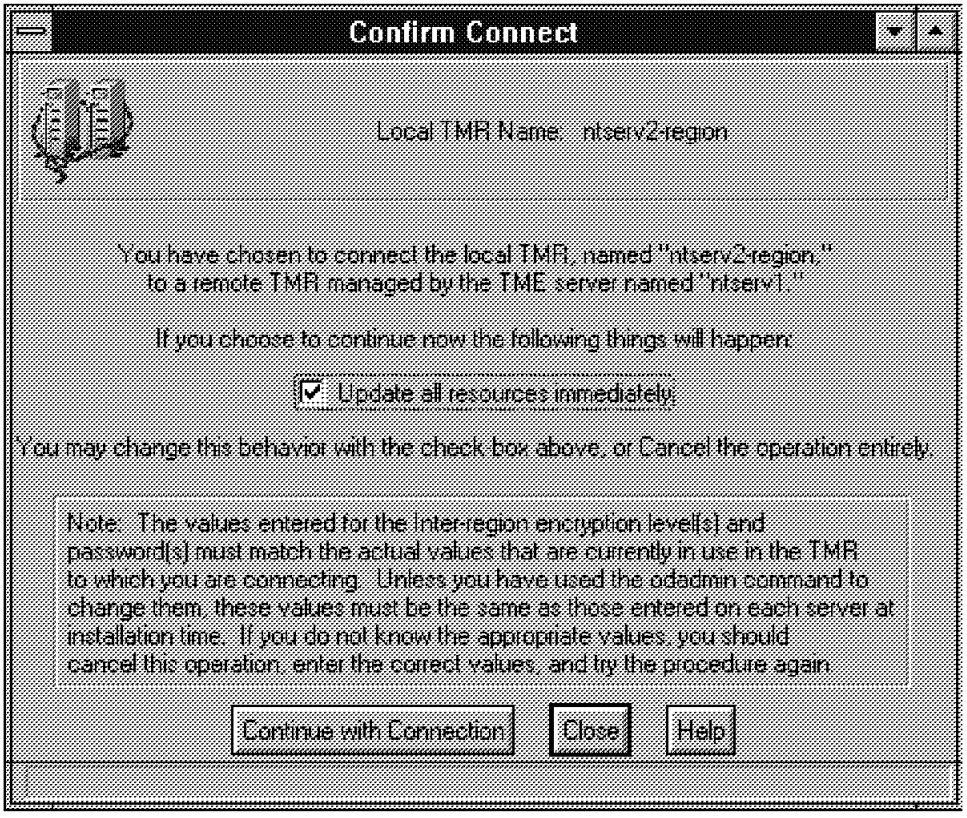


Figure 77. TMR Inter-Connections - Confirm Resource Update

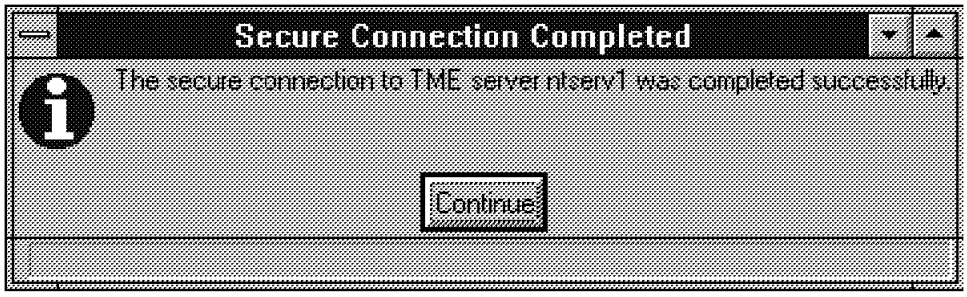


Figure 78. TMR Inter-Connections - Secure One-Way Manager TMR Confirm

In order to successfully connect a one-way connection, valid region numbers need to be provided. You can get them with the odadmin command. When selecting this type of connection one server is defined as the managing TMR and the other as the managed TMR.

When the type of connection has been decided upon and configured correctly, the connection is achieved by choosing **Connect** or **Connect & Close**. A panel next asks whether or not to update resources. Finally a panel confirming this will be presented upon connection creation.

3.2.3 Two-Way Connections

Two-way TMR connections present each TME server with access to the resources contained in each TMR. Therefore, each TME server can update and exchange information and resources between the connected TMRs. The order in which two-way connections are established does not matter. When creating a two-way remote connection the operation can be performed at either of the two TME servers. It is done the same way as a one-way connection except the two-way connection box should be checked.

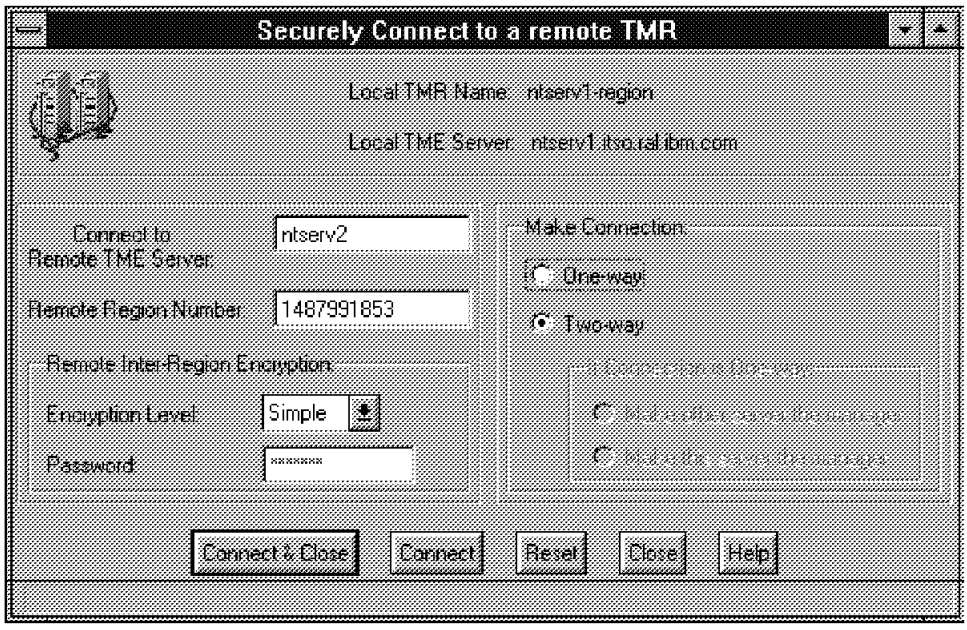


Figure 79. TMR Inter-Connections - Remote Two-Way TMRs

For a secure two-way connection the order in which the servers are configured does not matter since the operation must be carried out on both sides with valid TMR numbers to establish the connection.

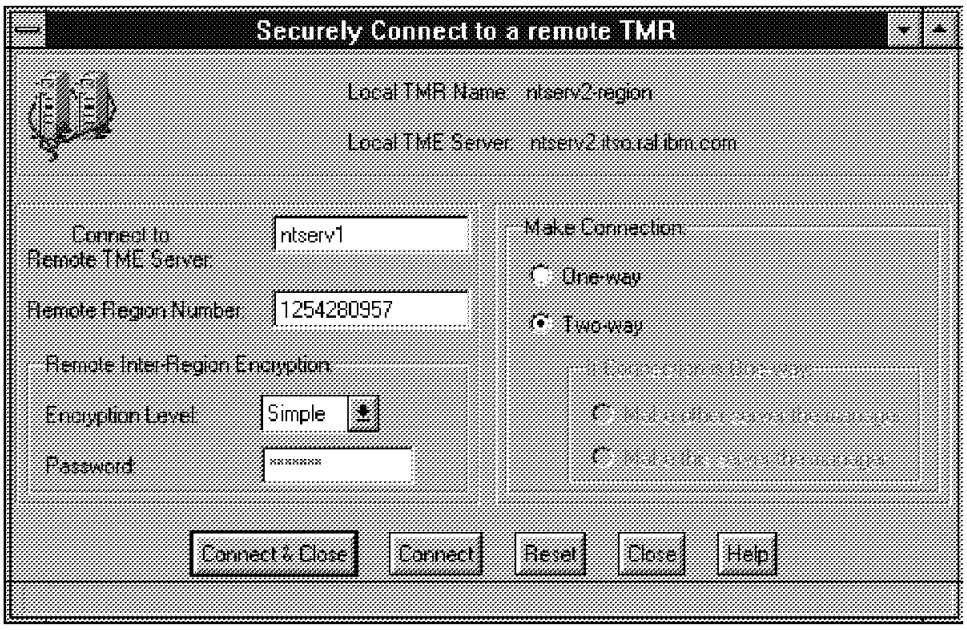


Figure 80. TMR Inter-Connections - Secure Two-Way TMRs

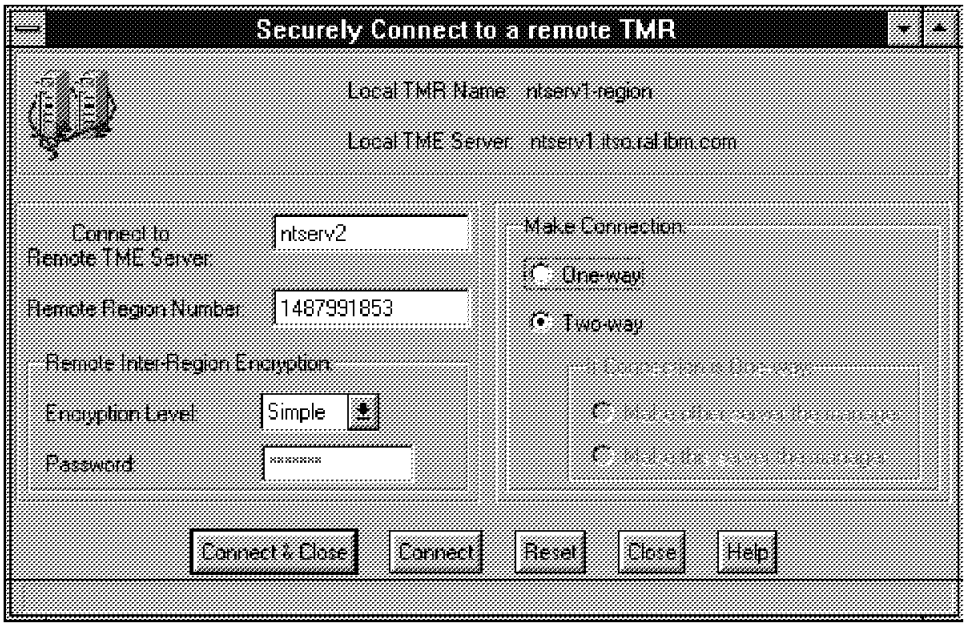


Figure 81. TMR Inter-Connections - Secure Two-Way TMRs

3.2.4 Changing Connection Types

After establishing connections between TMRs, whether they be one-way or two-way, they cannot be changed dynamically. Therefore it is not possible to change a one-way connections managing node to a managed node and vice versa. It is also not possible to change from a one-way connection to a two-way connection dynamically. To achieve any of these scenarios, the existing TMR connection must be broken (disconnected) and then re-established using the desired connection mode. When performing this, a database check should be run after breaking the connection and after establishing the new connection.

The command to perform the database check is wchkdb.

3.2.5 Cross Platform Windows NT - AIX TMR Connections

Connecting TMRs from Windows NT to AIX is identical to connecting Windows NT to Windows NT TMRs, whether the connections be one-way, two-way, remote or secure. The only provision being that the previously discussed rules for connecting TMRs are followed.

3.2.6 Updating Resources in Connected TMRs

When connecting TMRs, an update of the resources contained within each TMR should be carried out at regular intervals. These updates will be required less and less as the TME environment stabilizes through time, although initially may be quite frequent. It is also possible to schedule the update of resources to occur when the network is at its least utilization.

To update resources from the TME Desktop, select **Desktop**, **TMR Connections** and **Update Resources**, and you are presented with the following panel:

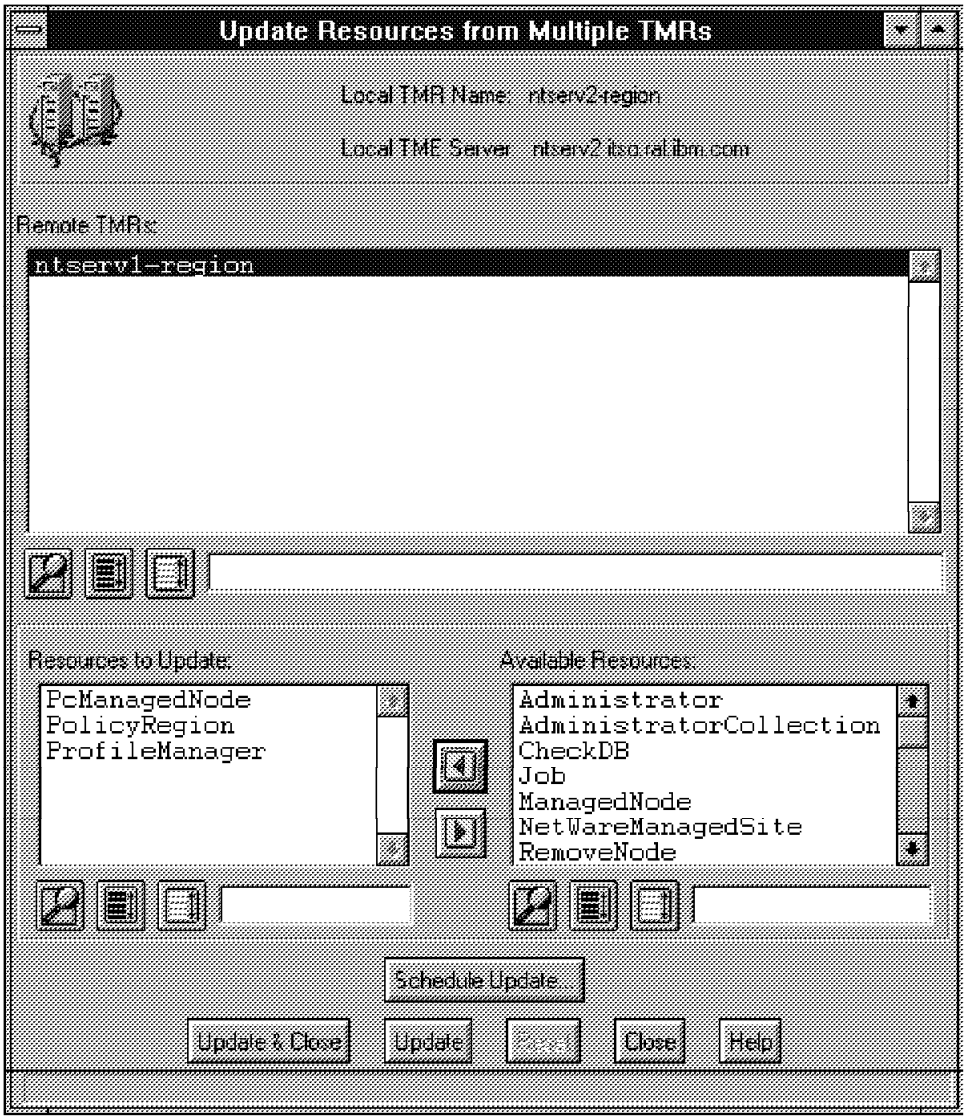


Figure 82. TMR Inter-Connections - Updating Resources

Using this panel it is possible to update either all of the available resources or selected resources, in either single or multiple available TMR connections.

3.2.7 Viewing TMR Connections

It is possible to view the valid TMR connections currently available within the TME. From the TME Desktop, choose **Desktop**, **TMR Connections** and **List Connections**.

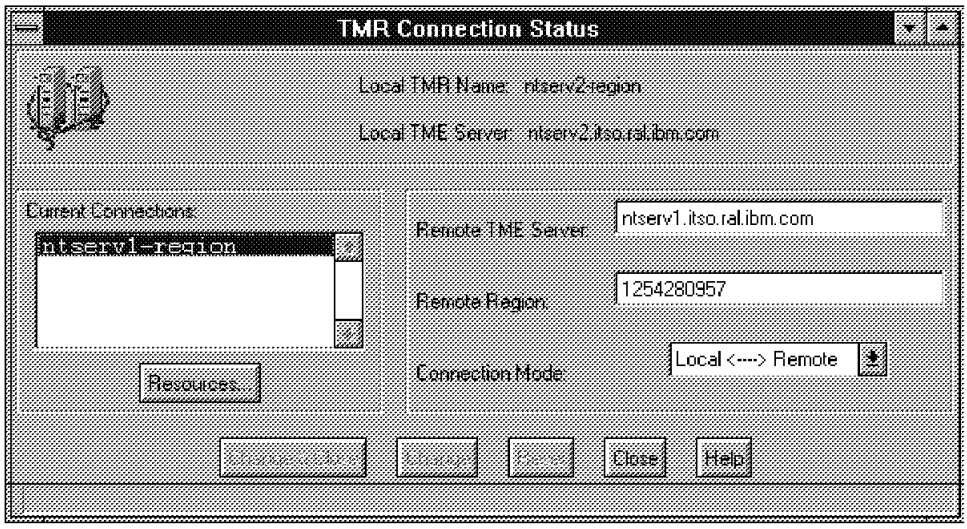


Figure 83. TMR Inter-Connections - Viewing TMR Connections

By selecting a current connection, you can obtain the remote server with which you are connected, the remote region number and the connection mode with which the TMRs are connected. It is also possible to view and update the resources within the selected current connection by selecting the **Resources** button and following the same procedures as described in the previous section.

3.2.8 Disconnecting TMRs

TMRs can be disconnected from the desktop panel, by selecting **Desktop**, **TMR Connections** and **Disconnect**. The process of disconnecting TMRs can be an extremely dangerous and time consuming process. It is therefore recommended that TMR disconnection be done with caution and at times suitable to the organization or even at a scheduled time. After disconnecting TMRs, the TME databases should be checked for consistency using the wchkdb command.

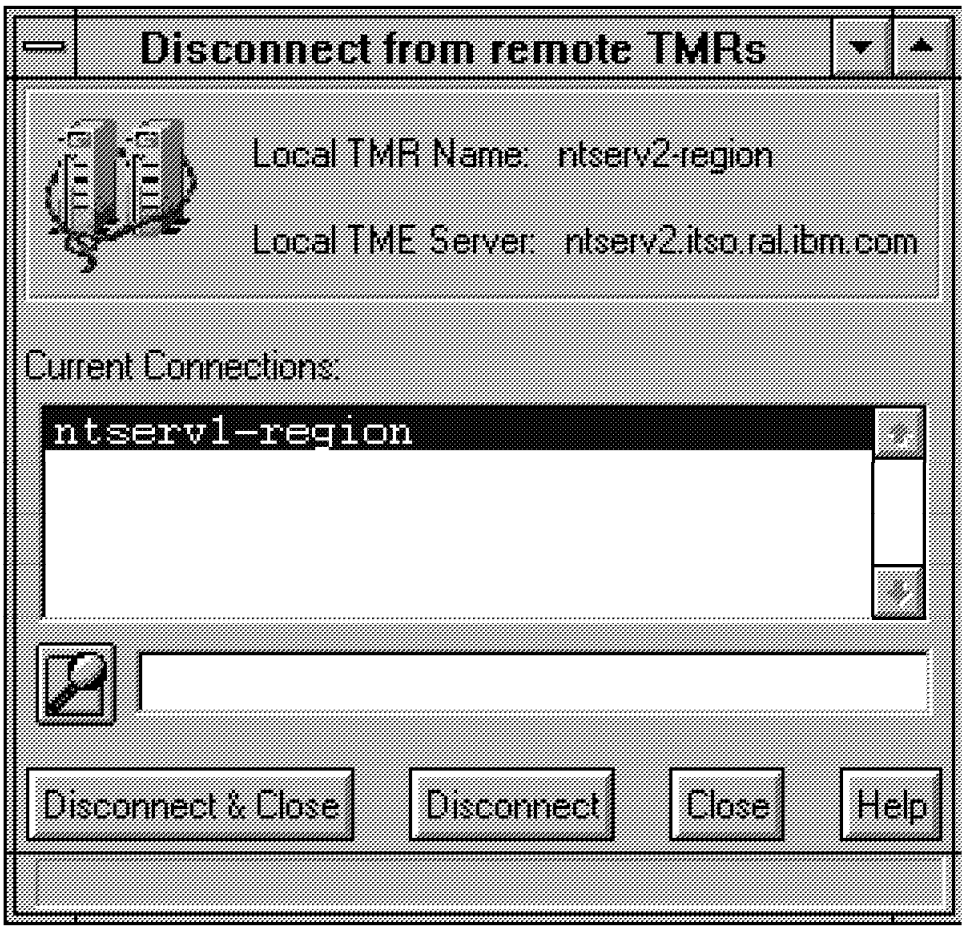


Figure 84. TMR Inter-Connections - Disconnecting TMRs

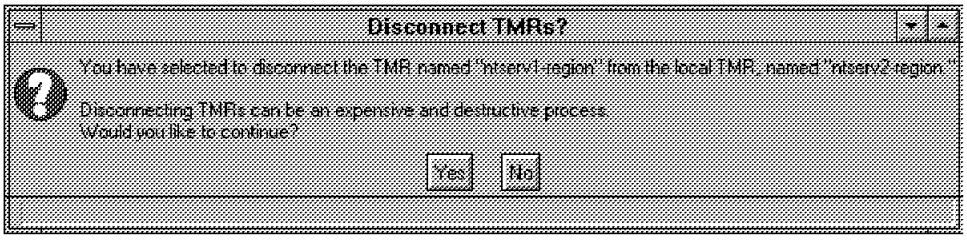


Figure 85. TMR Inter-Connections - Disconnecting TMRs

3.3 Structuring Connected TMRs - Possible Scenarios

There are a number of possible ways to structure TMR inter-connections. They differ greatly from organization to organization. In this section discuss some likely scenarios.

3.3.1 Single TMR Scenario

This would involve one TME server with no connections to other TME servers, hence only consisting of a single TMR. This would probably only be used within small organizations, as each TME server has an approximate upper limit of 200 nodes per TMR. It is possible to have multiple TMRs in an enterprise with less than 200 nodes.

This scenario would have only one TME server. It would contain up to 200 nodes within its TMR, and would have no connections to other TMRs.

Figure 86. TMR Scenario - Single TMR Scenario

3.3.2 Star Configuration Scenario

In Figure 87 we have a scenario where one central TMR E has connections to the other four TMRs A, B, C, and D. TMR E, therefore, acts as a central repository of information as it in effect is the manager of all four inter-connected regions. TMR E can alter and update any of the other four regions, as it has control of each one. This can be shown from the desktop using the pull-down from the desktop for **TMR Connections**, then **List Connections**. TMRs A, B, C, and D cannot update any of the other TMRs as they have no connection authority which allows them this capability.

A possible usage for this configuration would be in an organization with all the IT service personnel located at a single site, where they can control what is managed by each region and what each user gets on their TME desktop.

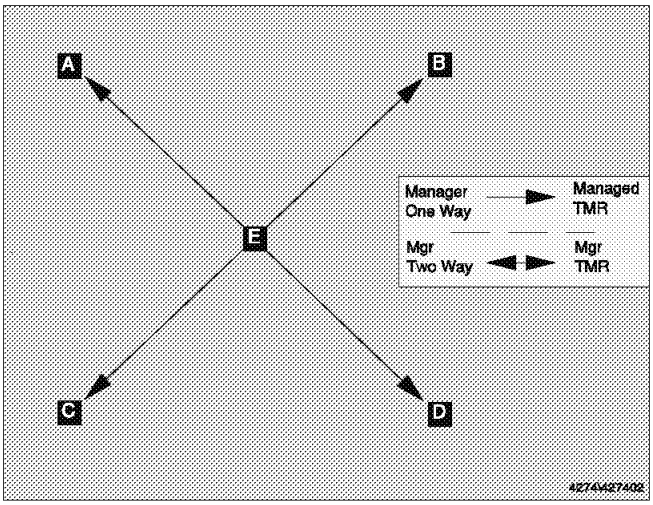


Figure 87. TMR Scenario - Star TMR Scenario

3.3.3 Hierarchical Configuration Scenario

Here we have a hierarchical configuration, similar to the directory structure found under most operating systems. TMR A has two one-way connections to TMRs B and C, and these have one-way connections to TMRs D and E and F and G. Since TMR A has no connection with TMRs D, E, F or G, it cannot view or update resources held in these TMRs directly. They can be accessed by manipulating those resources held within their directly connected TMRs B and C.

To allow for D, E, F or G to be accessible from A, they would have to have a connection from A, either a one-way connection, where A is the managing node, or a two-way connection, where either end′s resources are visible to both.

A good use of this configuration could be for software distribution purposes, where a distribution job is sent to TMRs B and C, and fanned-out from there on to TMRs D, E, F and G.

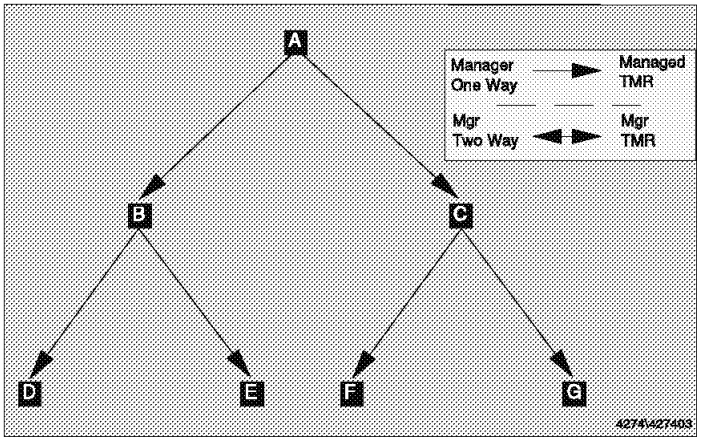


Figure 88. TMR Scenario - Hierarchical TMR Scenario

3.3.4 Star (Two-Way Connections) TMR Scenario

In the following scenario, although there appears to be connections between all the TMRs, each of the perimeter TMRs can only see two of the three other perimeter TMRs. TMR E however, again acts as a central repository although in this case is not the sole managing TMR, as it has two-way connections with other TMRs.

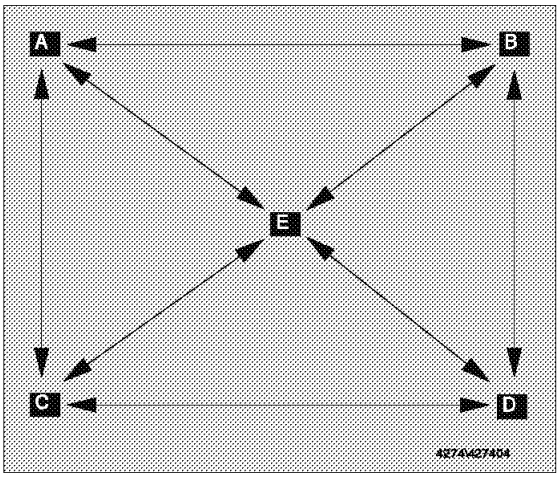


Figure 89. TMR Scenario - Star (Two-Way Connections) Scenario

3.3.5 Triangular TMR Scenario

This triangular scenario, with two two-way connections from to TMR A to TMRs B and C, does not provide a path between B and C to update and exchange information. It does allow for updates from TMR B to TMR A and from TMR C to TMR A.

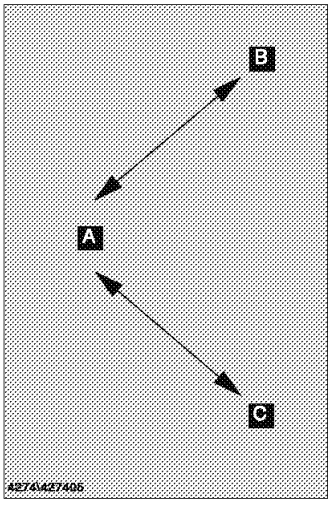


Figure 90. TMR Scenario - Triangular TMR Scenario

3.4 Our TME 3.0 NT TMR Structure Scenario

Our Tivoli management environment consisted of three TMRs. Two of the TMRs contain NT TME servers, while the third contains an AIX TME server.

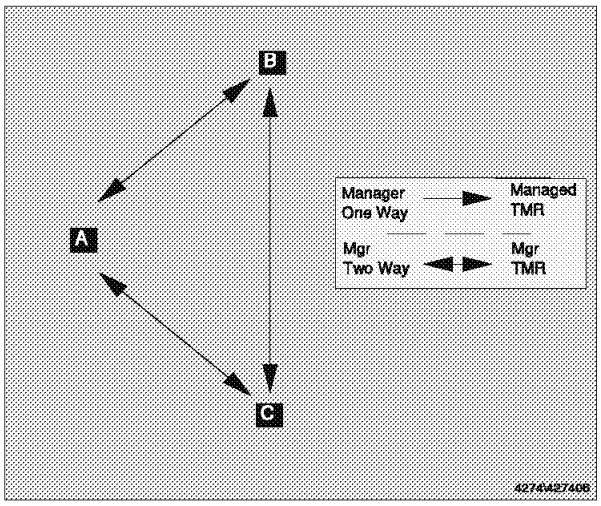


Figure 91. TMR Scenario - Our NT TME/TMR Scenario

From the above diagram TMR A contains the TME server ntserv2 with the following PcManagedNode resources:

- Windows NT Workstation
- Windows 95 Client
- Windows 3.1 Client
- Windows 3.11 Windows for Workgroups Client
- OS/2 Warp Connect Client

TMR B contains the TME server ntserv1 with only OS/2 PcManagedNode resources.

TMR C contains an AIX TME server rs600012 and contains only AIX ManagedNode resources.

Each of the TMRs are inter-connected by two-way connections, thus permitting each TMR to view and manage the other.

3.5 Command Line Usage for Connecting TMRs

NAME: wconnect

The wconnect command establishes connection between TME servers in two TMRs. This command is used for both one and two-way connections, whether those connections are established securely or by the nonsecure remote connection method.

PURPOSE: To inter-connect two TMRs

SYNOPSIS:

- connect [-u] [-m mode.] [-r encrypt_level] -s server region
- wconnect [-nu] [-c encrypt_level] [-l login] [-m Two-way | Managing:] [-r encrypt_level] server

EXAMPLES:

1. `wconnect -u -c simple -l root -m Managing -r simple rs600012` - Creates a one-way remote connection with the issuing TM server ntserv2 defined as the managing server within the TMR connection. When issuing this command you are prompted for the root password as well as the passwords for both the local and the remote TMRs forming the inter-connection.
2. `wconnect -u -s rs600012 1899640400` - Will establish one side of a two-way secure connection. To complete the connection a similar command must be executed on the other server performing the connection. A valid command would be wconnect -u -s ntserv2 1254280957.

3.5.1 Updating TMRs

NAME: wupdate

The wupdate command will update resources held in the local name registry, with information from one or more connected TMRs.

PURPOSE: To update resources held in the local name registry.

SYNOPSIS:

- wupdate -r *resource* [-r *resource...*] *TMRs...*

EXAMPLES:

1. `wupdate -f -r All ntserv1-region` - Updates the local name registry (ntserv2) with all the resource types from the ntserv1-region TMR.
2. `wupdate -r PcManagedNode -r PolicyRegion All` - Will update the local name registry (ntserv2) with the resource types of PcManagedNode and PolicyRegion from all connected TMRs.

3.5.2 Checking TMR Resources

NAME: wlookup

The wlookup command will look up the object information for resources held within the Tivoli name registry.

PURPOSE: To look up resource object information.

SYNOPSIS:

- wlookup [-l] -R
- wlookup [-r resource_type] [-n resource_name] -a [-L | -o] | name

EXAMPLES:

1. `wlookup -R` - Will list all of the resource types registered for that TMR.

```
ActiveDesktopList
Administrator
AdministratorCollection
CheckDB
Classes
distinguished
Job
ManagedNode
NetWareManagedSite
NtRepeat
PatchInfo
PcManagedNode
PolicyRegion
Presentation
ProductInfo
ProfileEndpoint
ProfileManager
RemoveNode
Repeater
Scheduler
TaskLibrary
TaskRepository
TMF_Notice
TopLevelPolicyRegion
```

Figure 92. Registered Resource Types

2. `wlookup -r PcManagedNode -a` - Will list all of the resource instances of resource type PcManagedNode:

```
tmecli1 1254280957.1.426#TMF_PcManagedNode::Pc_Managed_Node#
tmecli2 1487991853.1.464#TMF_PcManagedNode::Pc_Managed_Node#
tmecli3 1487991853.1.428#TMF_PcManagedNode::Pc_Managed_Node#
tmecli6 1487991853.1.429#TMF_PcManagedNode::Pc_Managed_Node#
tmecli7 1487991853.1.431#TMF_PcManagedNode::Pc_Managed_Node#
```

Figure 93. Resource Instances of PcManagedNode

3. `wlookup -r PcManagedNode ntserv1` - Will fail since ntserv1 is not an instance of a PcManagedNode.
4. `wlookup -r ManagedNode ntserv1` - Will however succeed, since it is an instance of a ManagedNode resource:

   `1254280957.1.322#TMF_ManagedNode::Managed_Node#`

3.5.3 Listing TMR Connections

NAME: wlsconn

The wlsconn command will display a list of current TMR connections or information about a specific connection. It is also possible to complete the information exchange between the local TMR and its connected region.

PURPOSE: To list the valid current TMR inter-connections, and to complete the information exchange.

SYNOPSIS:

- wlsconn [*TMR_name*]
- wlsconn -u *region*

EXAMPLES:

1. wlsconn - Displays all the current connections, including connection mode, connected TMR, connected server and region number.

```
MODE    NAME              SERVER                    REGION
<---->  ntserv1-region    ntserv1.itso.ral.ibm.com  1254280957
---->   rs600012-region   rs600012                  1899640400
```

Figure 94. Current Connections

2. wlsconn ntserv1-region - Displays details about the connected TMR called ntserv1-region.

```
Name:   ntserv1-region
Server: ntserv1.itso.ral.ibm.com
Region: 1254280957
  Mode: two_way
  Port: 94

Resource Name          Last Exchange
-------------          -------------
TMF_Notice             05/07/96 12:53:25
Administrator          06/07/96 01:44:21
PolicyRegion           06/10/96 05:15:37
TaskLibrary            06/07/96 10:17:44
Job                    01/01/70 12:00:00
ProfileManager         06/07/96 09:50:59
ManagedNode            06/04/96 05:27:32
PcManagedNode          06/06/96 03:48:18
Repeater               06/04/96 05:28:57
CheckDB                05/07/96 12:45:57
RemoveNode             05/07/96 12:45:59
NetWareManagedSite     01/01/70 12:00:00
```

Figure 95. Details About the Connected TMR

3. wlsconn -u 1254280957 - Updates the information exchange between the two connected TMRs by using the remote TMR number. The exchange mechanism used is a pull from the remote TMR, to the local TMR (in this case ntserv2). There is no screen output.

3.5.4 Checking Database Integrity

NAME: wchkdb

The wchkdb command is used to perform the update of resources between connected TMRs. It is also used to complete the information exchange between the local TMR and its connected region.

PURPOSE: To update information between connected TMRs.

SYNOPSIS:

- wchkdb [**-o** *outfile*] [**-u**] [**-x**] {**-f** *infile* | **-i** | *object*[*object*.]

EXAMPLES:

1. wchkdb - Checks the database for any inconsistencies. It will not attempt to fix any inconsistencies.

```
wchkdb: Preparing object lists:
wchkdb: Checking object database:
     ........................................
wchkdb: Done checking object database.
```

Figure 96. Database Checking

2. wchkdb -u -x checks the database and, if required, fixes the TME database.

```
wchkdb: Preparing object lists:
wchkdb: Checking object database:
     ........................................
wchkdb: Processing "tmecli6"
(1487991853.1.429#TMF_PcManagedNode::Pc_Managed_Node#):
The object type of "ntserv1-region"
(1254280957.1.195#TMF_PolicyRegion::GUI#)
is not supported by its Policy Region.
     ........................................
wchkdb: Done checking object database.
```

Figure 97. Check and Fix the Database

3.5.5 Disconnecting TMRs

NAME: wdisconn

The wdisconn command is used to provide the facility to disconnect established TMR inter-connections. It can be used on any type of TMR connection.

PURPOSE: To disconnect TMRs.

SYNOPSIS

- wdisconn [-s] TMR_name
- wdisconn. [-s] -r region

EXAMPLES:

1. wdisconn -s rs600012 - Disconnects the connection between the local TMR at ntserv2 and the remote TMR at rs600012.
2. wdisconn -r 1899640400 - Disconnects the connection between the local TMR at ntserv2 and the remote TMR at rs600012 using the TMR number rather than the TMR name.

Chapter 4. Microsoft NT Server Environment

The configuration of our NT servers within the TME environment is discussed in this chapter. The reasons for configuring NT in this way, as well as how NT combines with other network operating systems is shown. We also discuss the Microsoft Operating System family positioning with regards to NT Server V3.51, NT Workstation V3.51 and Windows 95. This includes the differences between each of them.

We also discuss any extra features installed on the NT Server platform, which were used to give added functionality.

4.1 NT Operating System

Windows NT Server is a high-end network operating system provided by Microsoft. It is a full 32-bit operating system, with preemptive multitasking, and multithreading capability. A multitasking operating system is one which allows applications to run concurrently by dividing up the CPUs time between them. Preemptive multitasking, is a scheme that allows the operating system to override a CPU-intensive application. Multithreading is a process which allows a multitasking operating system to split an application into multiple sub-processes, also called threads, and multitask them.

NT is very scalable, supporting symmetric multiprocessing and the ability to run on Intel as well as DEC Alpha and PowerPC architectures. NT is a highly reliable operating system with a micro-kernel based upon the VMS operating system. It provides fault tolerance to the level that no single rogue process will bring the system down.

NT Server comprises, among other things, centralized management of file and print sharing, networking facilities, security and fault tolerance. NT Workstation, is similar to the server product, but without the functionality of centralized management.

The following represents a quick overview of the internal structure of the NT operating system. A more detailed discussion can be found in NT reference books.

The internal system architecture of NT can be split into two sections: the different subsystems that can be accessed, and the system services that these subsystems access. As can be seen from Figure 98 on page 82, we have a number of operating system subsystems which all feed into the main NT subsystem. Before accessing different system services, applications call their relevant subsystem. For example, a Windows 16-bit application would access the DOS/Win16 subsystem and any system calls are then mapped to the Win32 subsystem. All of this happens in what is called user mode (or unprivileged mode). That is to say that all user mode execution does not access system resources. The system resources, such as the hard disk, must be requested through what is known as the NT executive. The executive is the bulk of NT′s operating system kernel and is fully protected from the subsystems and any user applications. The executive provides all the services that the subsystems cannot provide. When a subsystem requires a service it makes a call to the appropriate service manager in the executive and runs in protected mode (prevents outside

intervention from applications) to perform the desired action. Upon completion, the executive returns control to the controlling party.

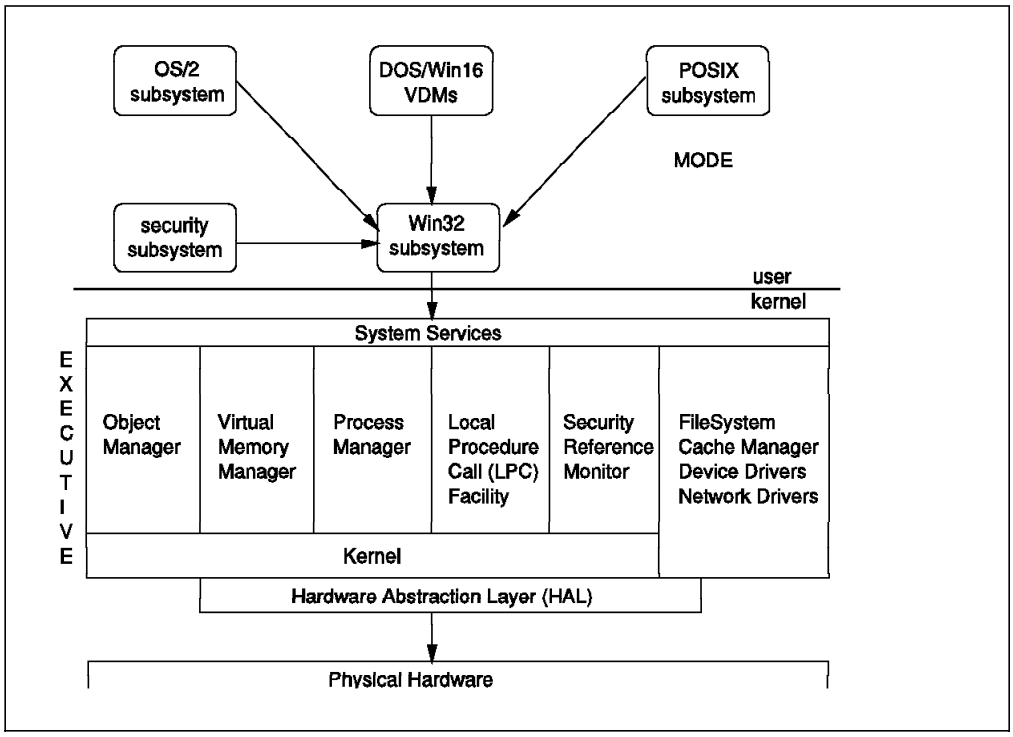


Figure 98. NT Base Architecture

Also, within NT, we have a layer called the hardware abstraction layer (HAL). The HAL basically is the base of NT's operating system, forming a bridge between NT and the physical hardware it is running on. In essence, the only parts of NT that communicate with the hardware are the kernel and I/O drivers. All the other modules within NT do not know anything about what machine they are running on. The HAL allows NT to be transportable among hardware platforms, requiring only minimal alteration since the majority of system modules should be able to be ported with no alteration.

4.1.1 NT Registry

Windows NT stores its configuration within a database. This database takes on a tree-like structure, and stores all aspects of the NT system. The registry holds all the configuration details of the system including hardware and software parameters. The registry of an NT machine can be accessed by issuing the command `regedt32.exe`. This command initiates the registry editor as shown in Figure 99 on page 83. The editor can also be accessed by double-clicking on the **Administrative Tools** and then **Windows NT Diagnostics**. Then click on the **Tools** pull-down menu and select **Registry Editor**.

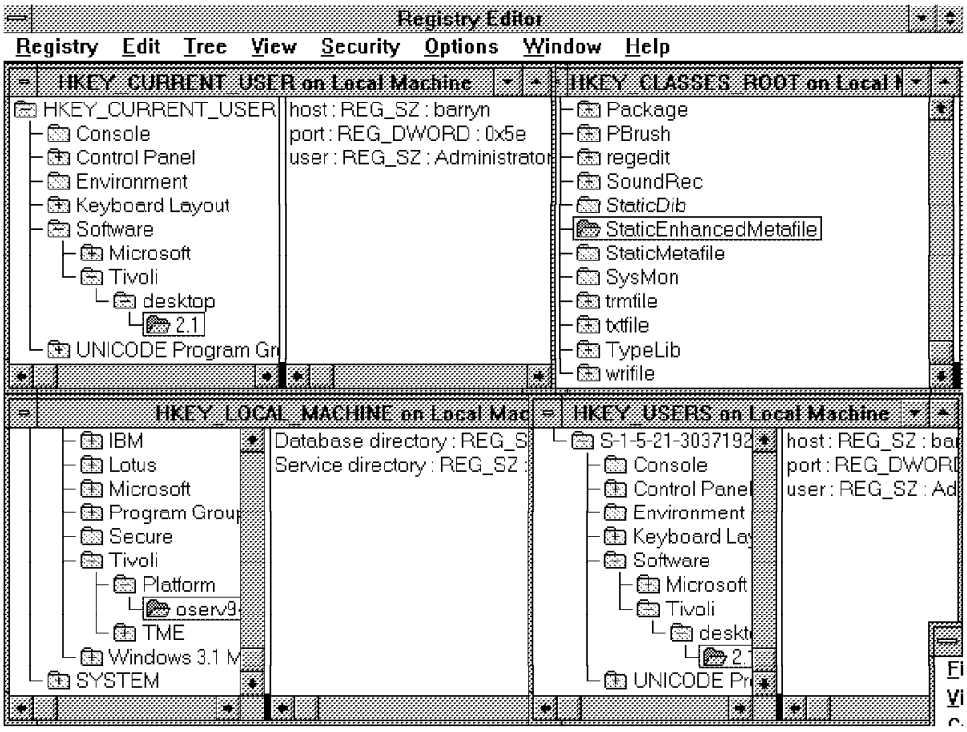


Figure 99. NT Registry Editor - A Hierarchical Information Repository

Figure 99 shows that there are four default windows within the registry editor, each representing a key on the local system. Keys are located within the registry window representing folders. There is extensive help for using the registry by clicking on the **Help** pull-down menu item and **Contents**. This will result in the following window:

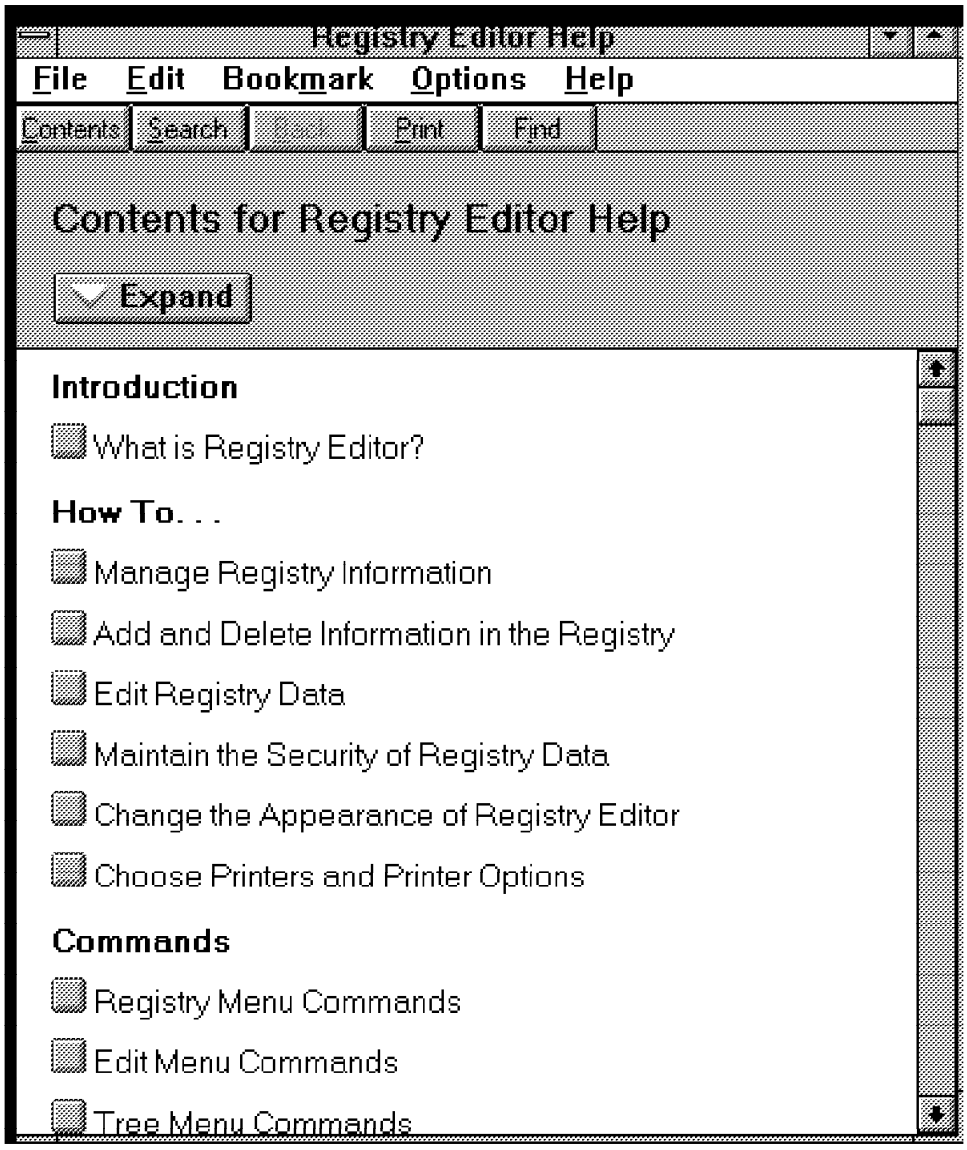


Figure 100. NT Registry Editor Help

In addition, you can get help on all of the fields if you have the NT Resource Kit installed as shown in the following window:

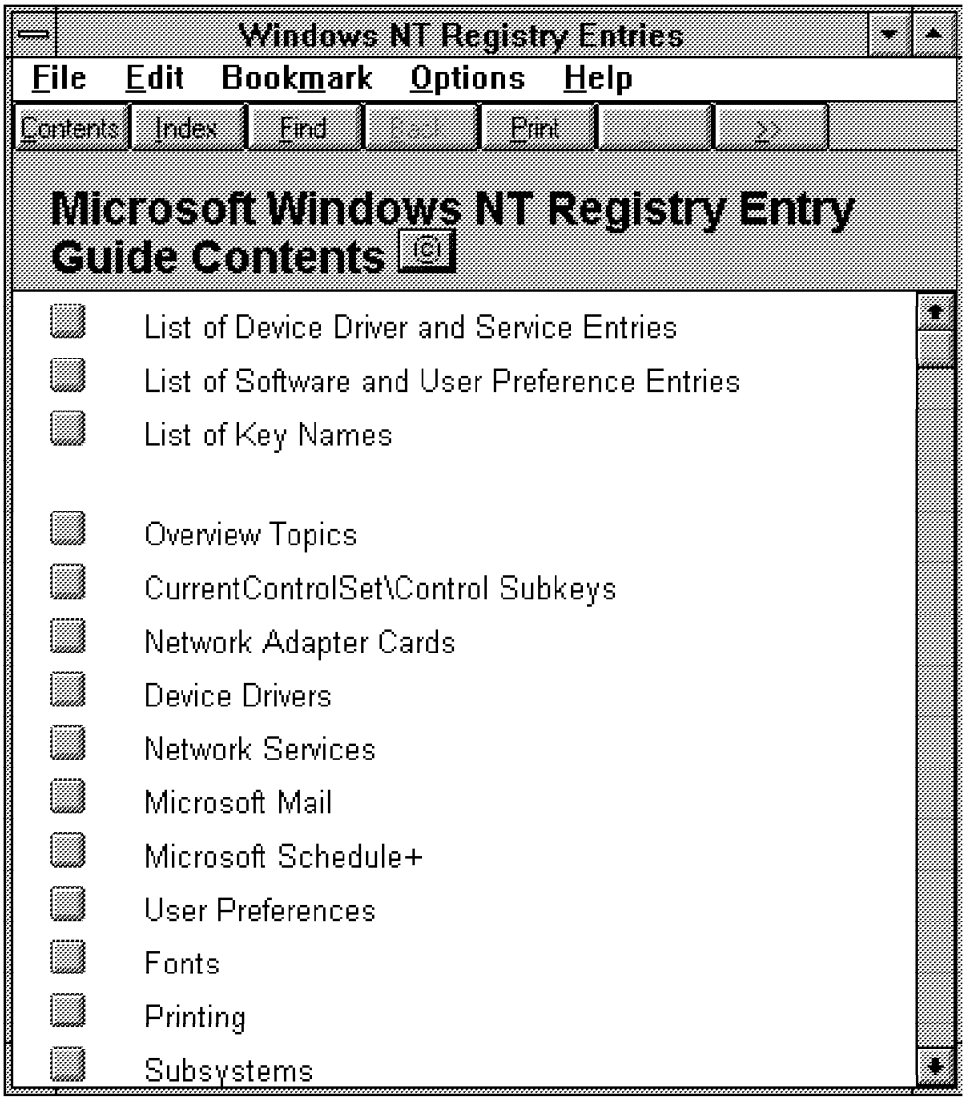


Figure 101. NT Resource Kit Registry Field Help

- HKEY_CURRENT_USER - Root configuration information for the currently logged on user, containing all the parameters for that user's setup, such as desktop settings and environment setup.
- HKEY_USERS - Contains all the user profiles for the system, therefore HKEY_CURRENT_USERS is a subkey of HKEY_USERS.
- HKEY_LOCAL_MACHINE - Contains the configuration specific to the system regardless of the user. This includes hardware configuration and software configuration.
- HKEY_CLASSES_ROOT - A subkey of HKEY_LOCAL_MACHINE. Here information such as file association and OLE is stored.

Within the registry editor, it is possible to assign values to new keys or to alter the value entries assigned to a currently selected key. Registry value entries appear as strings consisting of three components.

At the left-most side of the value entry pane, the name of the value appears, followed by the value entry class and finally the value entry of the selected key.

Registry key classes can be split into four categories, which have an associated editor:

- REG_BINARY binary entry.
- REG_SZ data string entry.
- REG_DWORD DWORD entry.
- REG_MULTI_SZ multiple string entry. REG_EXPAND_SZ indicates that the entry is an expandable string.

Note

Before making any changes to the registry, you should make sure that you have a backup copy of it. The regback command from the NT Resource Kit is helpful at backing up the registry.

Double-clicking on an entry in the registry puts the item into edit mode. The following example shows an excerpt from the registry, showing the key for SCSI devices. We have two values for that. One named Identifier and one named Type. The class value for both of them is REG_SZ, and the corresponding values follow:

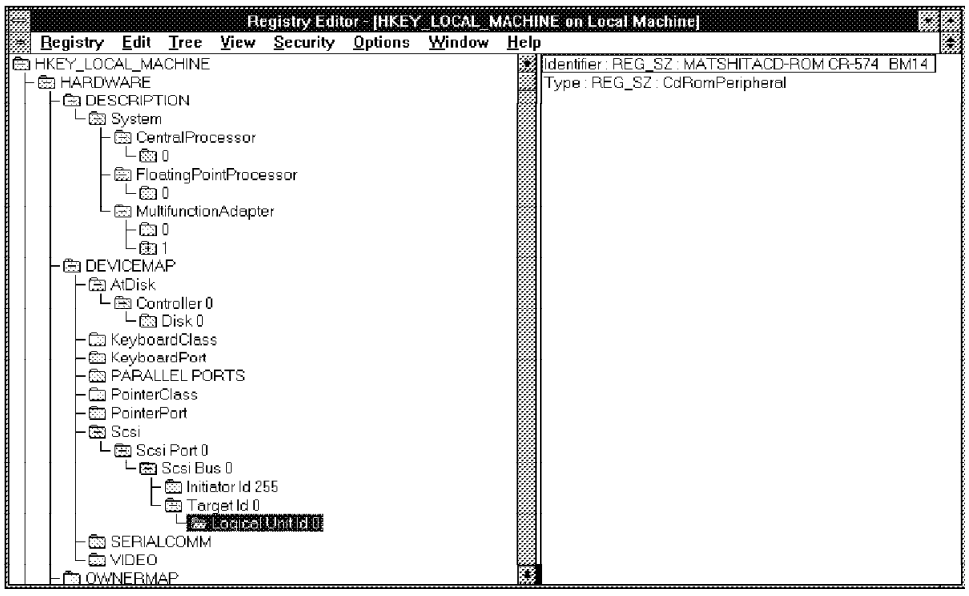


Figure 102. SCSI Registry Key Example

It is not normally necessary to edit the registry, as this can cause more problems than it solves. At times, it may be necessary to browse the registry and remove keys that are no longer required. For example, previously installed software that is no longer required may have a few entries left within the registry.

Also defined within the registry are hives. Hives are keys that are defined within the registry that appear as physical files on the local machines hard disk drive. These files can only be edited by the registry editor, but they can be transported to remote machines.

Most of (but not all) the registry is stored in the hives. The hives are binary files, so the only way you can edit them (caution here) is through the registry editor.

You need to have administrator authority to edit the registry/hives. The hives are stored in the WINNT35\SYSTEM32\CONFIG directory. The hives consist of keys, subkeys and various values for them. The hive files/keys are:

- HKEY_LOCAL_MACHINE-SAM
- HKEY_LOCAL_MACHINE-SECURITY
- HKEY_LOCAL_MACHINE-SOFTWARE
- HKEY_LOCAL_MACHINE-SYSTEM
- HKEY_USERS-DEFAULT
- HKEY_USERS-Security ID
- HKEY_CURRENT_USER
- HKEY_CLASSES_ROOT

The following shows where the hives are located:

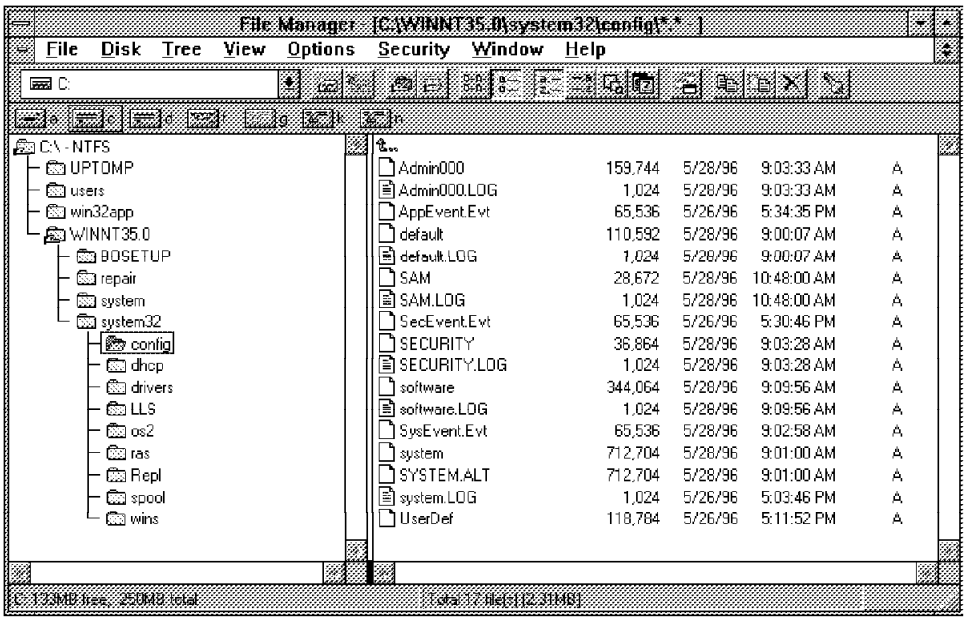


Figure 103. NT Registry - NT Registry Hives Location

In fact it is possible to look at remote machines registry information from the name registry, by selecting **Select Computer** from the Registry pull-down menu within the registry editor.

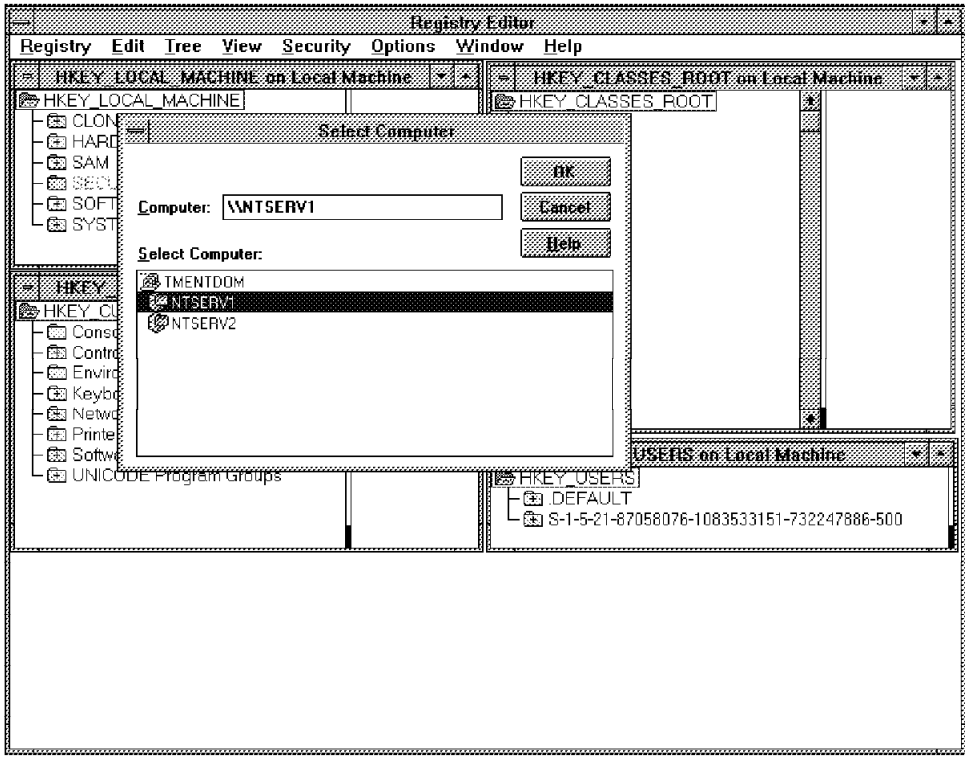


Figure 104. Selecting Remote NT Registries

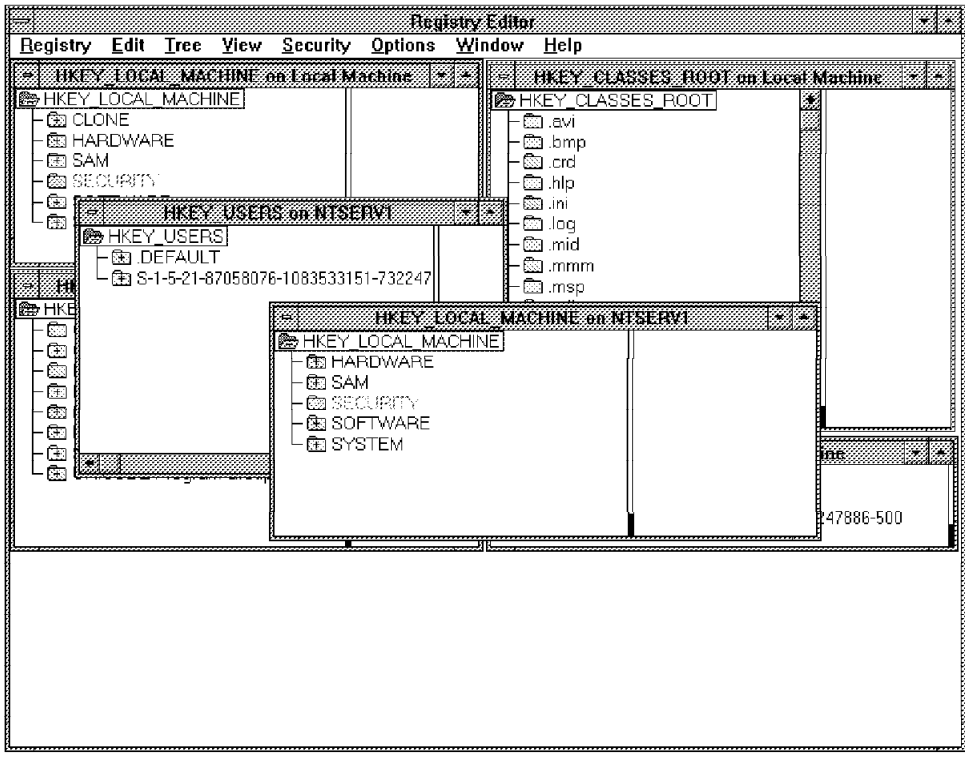


Figure 105. Viewing Remote NT Registries

4.1.1.1 TME NT Registry Entries

The following entries are made into the registry when Tivoli is installed. The first example is in HKEY_LOCAL_MACHINE and is a reference to the Tivoli de-install procedure.

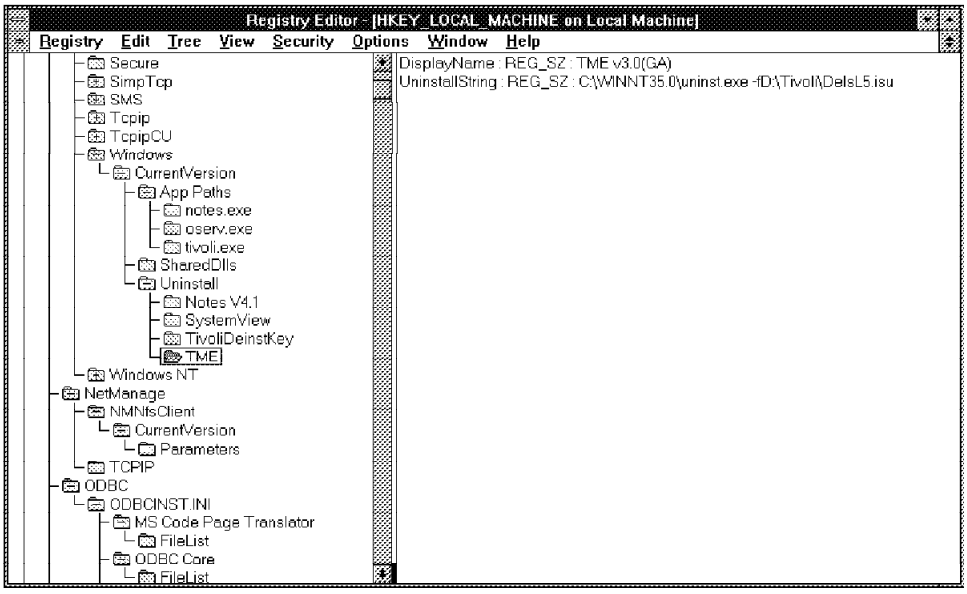


Figure 106. The Tivoli De-Install Registry Entry

The next entry is the one for the Tivoli Program group, to appear on the desktop, again within HKEY_LOCAL_MACHINE.

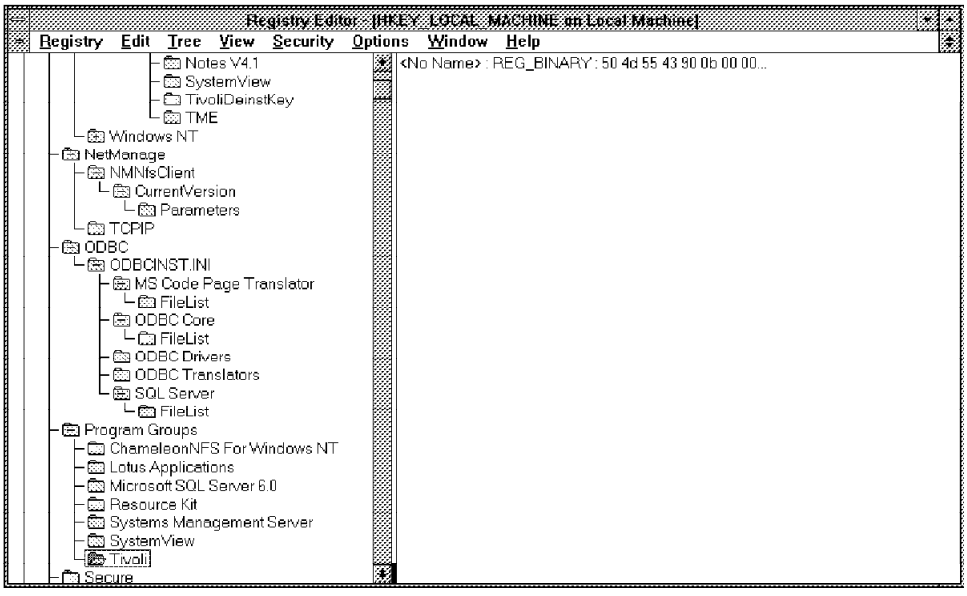


Figure 107. The Tivoli Program Group Registry Entry

The next entry contains all the installation details, such as name, company and installation key, again within HKEY_LOCAL_MACHINE.

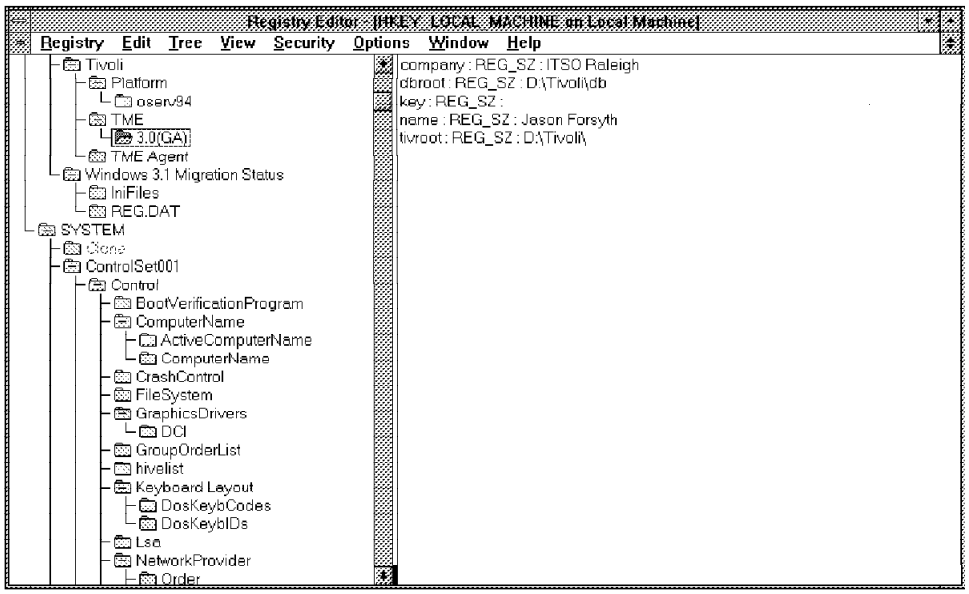


Figure 108. The Tivoli Platform Installation Details Entry

Finally, there are two entries, within both HKEY_USERS and HKEY_CURRENT_USER, containing details of the host to connect to and the user to connect as.

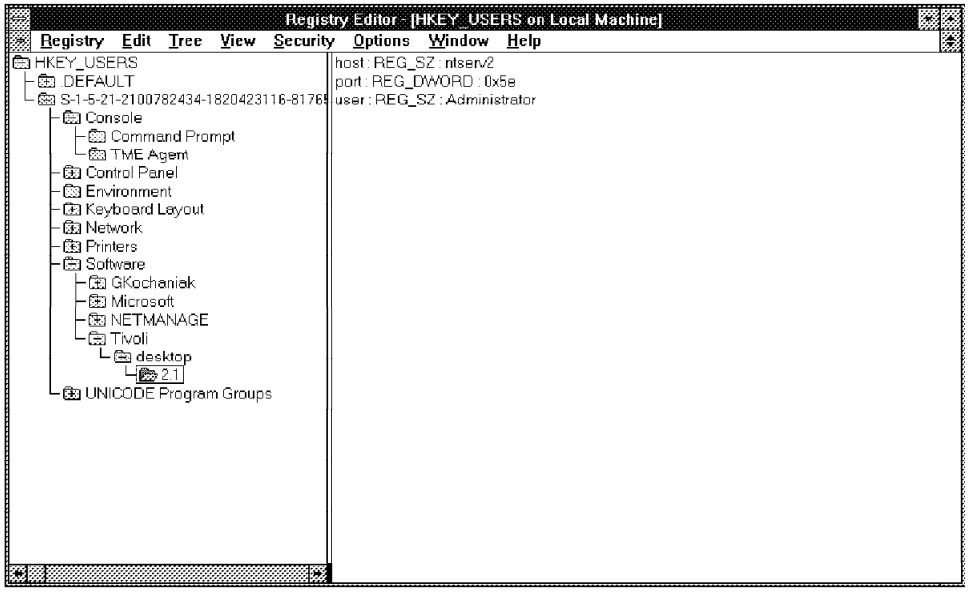


Figure 109. The Tivoli Logon Details Entry

If a TME re-install is required, these entries remain in the NT registry, and are picked up by the new install. Deleting all the Tivoli entries within the registry is the only way to get a clean re-installation.

4.1.2 NT Networking

The networking functionality within NT is such that it supports the NDIS specification. That is to say it bypasses the early network operating systems approach that provides monolithic device drivers, which prohibits a network interface card from using multiple transport layer protocols. The foundation for the NT networking architecture is NDIS (Network Device Interface Specification). Although NT does not provide a protocol manager, it uses bindings between the different protocols required, which are stored within the NT Registry. The following diagram depicts the NT networking architecture, and compares it with the OSI model.

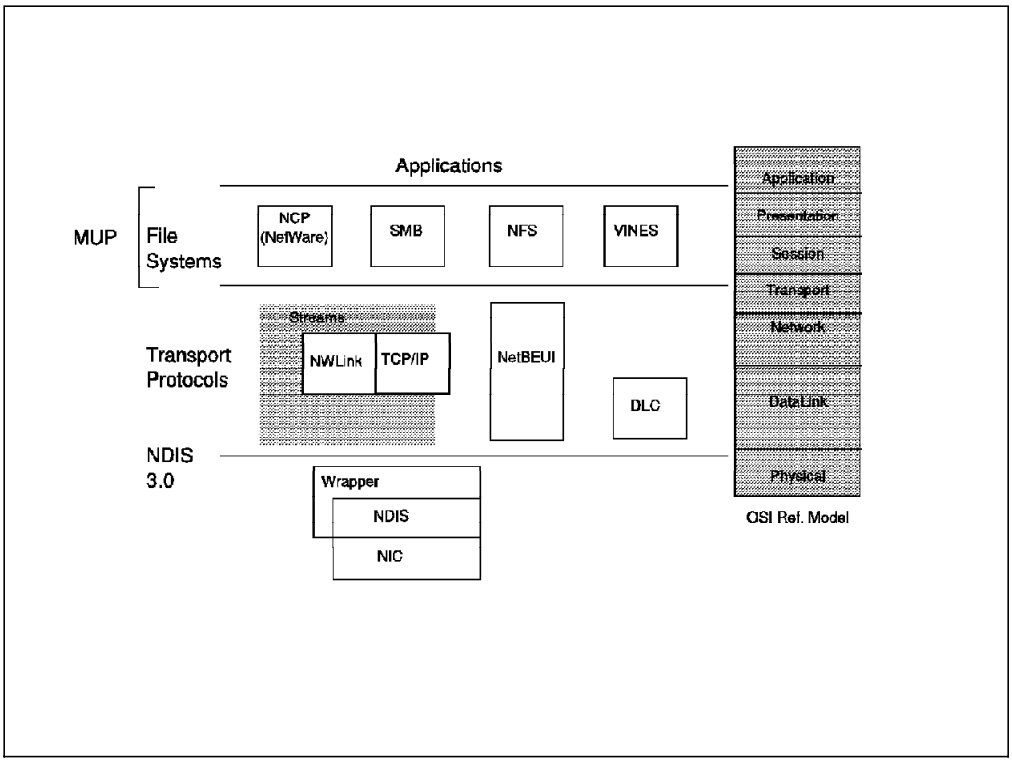


Figure 110. NT Networking Architecture

This is all very well, but how does NT access other network operating systems that are not from Microsoft? Access to OS/2 LAN Servers alias and printer facilities, AIX NFS mounts, and NetWare servers need to be examined.

Since NT has NDIS capability we can communicate with different transport layer protocols concurrently. We require more than just transporting data; we need to use the data transported. NT does this with a server and workstation, or redirector service. Between them these services provide the ability for work to be done at the higher levels of the OSI reference model.

The redirector service will take requests from applications, and translates them into SMBs (server message blocks) which are then sent to the network server. These are accepted by the server, which deals with the request and sends the requested data back to the requesting station.

Accessing resources from other systems can be done by referring to the required resources UNC (universal naming convention) name. UNC naming uses the following conventions:

\\server\share\sub_directory\filename

For example, accessing an OS/2 LAN Server alias can be done by:

net use x: \\<server_name>\<share> /USER:\\<domain_name>\<user>

Note

Here, the definition of share is any defined area within a server, made available to clients to access file or print facilities.

This gives access to an alias on the desired LAN Server share, with the user privileges defined by the given user name. It is also possible to connect to NFS mounts on remote UNIX boxes using a similar command. NT does not provide an NFS server or client but this can be achieved by using a third-party package, such as Chameleon NFS from NetManage.

```
net use y: \\<full hostname inc domain>\mount_point\sub_directory\file
net use y: \\barry6k.ibmus2.ibm.com\mnt\residency\winnt35.0
```

Figure 111. Net Use NFS Drives

The following example shows connections to local and remote shares, OS/2 LAN Server shares, Microsoft Windows for Workgroups shares, external CD-ROM shares and a connection to an NFS mount point on AIX. As can be seen, all the remote drives are treated as local drive letters.

```
net use

Status Local Remote                      Network


-------------------------------------------------------------------------
OK     F:    \\NTSERV1\pntcdrom          Microsoft Windows Network
OK     G:    \\BORG\CSET                 Microsoft Windows Network
OK     H:    \\WARPED\VIRUS              Microsoft Windows Network
OK     I:    \\CORNHOLIO\SEEDRIVE        Microsoft Windows Network
OK     N:    \\wtras2\lanmgmt            Microsoft Windows Network
       X:    \\rs6000012\var\spool\Tivoli  NetManage Chameleon32NFS
The command completed successfully.
```

Figure 112. Net Use Shared Drives

The following image of the NT File Manager shows all these connections as well as the various file system types that can be accessed, including FAT, HPFS386, NTFS, CDFS and an AIX mount point.

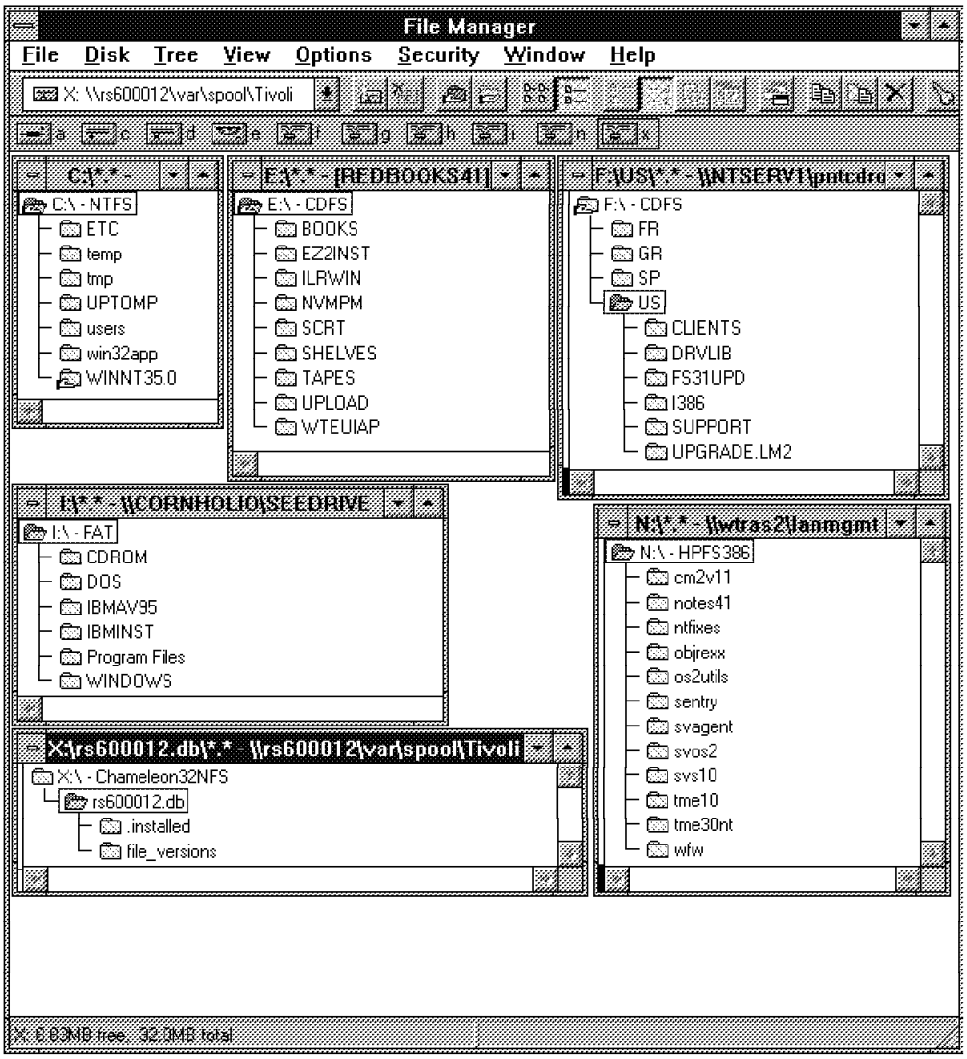


Figure 113. Various Network Connections and File Systems

NT will also support multiple transport protocols and multiple requestors concurrently. This enables access to NT, NetWare, OS/2 LAN Server and UNIX file systems all at the same time. This is achieved with the Multiple UNC Provider facility (MUP). This can be viewed as a multiple UNC name locator resulting in a \\server name. NT will attempt to find any share points available to be used.

NT uses a structure similar to that of the OSI model. It's a layered structure to give access to file and print sharing facilities. This layered structure, which encompasses boundaries such as NDIS and MUP, allows a single NT machine to access multiple transport protocols as well execute multiple requestors simultaneously.

4.1.3 NT Services

NT services are internal software routines that provide particular functions for a workstation application. These services can be NT operating system-specific routines such as the redirector or workstation service. These provide the ability to send messages to a server on the network requesting some action or a

third-party service to provide the NT station with NFS capability. Examples of this can be seen in Figure 118 on page 97.

All services that NT can start have remote procedure call (RPC) capability. This means that the services can be accessed from other remote systems over the network if required.

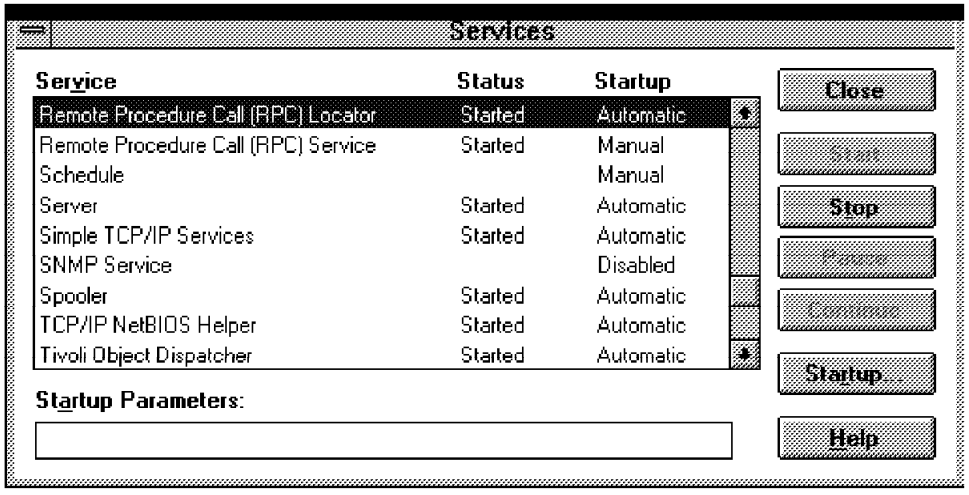


Figure 114. RPC Service

By using the NT command net start, a list of the active services can be obtained from the desired computer.

```
net start


These Windows NT services are started:


   Alerter
   Computer Browser
   EventLog
   FTP Server
   License Logging Service
   Messenger
   Net Logon
   NetManage Chameleon32 LPD
   NetManage Chameleon32NFS Client
   NetManage Chameleon32NFS Portmapper
   NetManage Chameleon32NFS Server
   Remote Shell Service
   Server
   Simple TCP/IP Services
   Spooler
   TCP/IP NetBIOS Helper
   Tivoli Object Dispatcher
   Tivoli Remote Execution Service
   Workstation

The command completed successfully.
```

Figure 115. Windows NT Services

The possible services on NT are as follows:

```
   ALERTER
   CLIENT SERVICE FOR NETWARE
   CLIPBOOK SERVER
   COMPUTER BROWSER
   DHCP CLIENT
   DIRECTORY REPLICATOR
   EVENTLOG
   FTP SERVER
   LPDSVC
   MESSENGER
   NET LOGON
   NETWORK DDE
   NETWORK DDE DSDM
   NETWORK MONITORING AGENT
   NT LM SECURITY SUPPORT PROVIDER
   OLE
   REMOTE ACCESS CONNECTION MANAGER
   REMOTE ACCESS ISNSAP SERVICE
   REMOTE ACCESS SERVER
   REMOTE PROCEDURE CALL (RPC) LOCATOR
   REMOTE PROCEDURE CALL (RPC) SERVICE
   SCHEDULE
   SERVER
   SIMPLE TCP/IP SERVICES
   SNMP
   SPOOLER
   TCPIP NETBIOS HELPER
   UPS
   WORKSTATION

These services are available only on Windows NT Server:
   FILE SERVER FOR MACINTOSH
   GATEWAY SERVICE FOR NETWARE
   MICROSOFT DHCP SERVER
   PRINT SERVER FOR MACINTOSH
   REMOTEBOOT
   WINDOWS INTERNET NAME SERVICE
```

Figure 116. Possible Services

For our project, we have a number of third-party services started, from Tivoli and NetManage, as well as the required services. These required services, and a description of their function are as follows:

- Alerter - Notifies selected users of administrative events. This service is used by the server service, and requires the messenger service
- Computer Browser - Maintains an up-to-date list of computers available to applications on request.
- Event Log - Records any events that occur and writes them to a log.
- Messenger - Sends and receives messages.
- Net Logon - Provides logon authentication and maintains the domain controller database on primary domain controllers and also as a logon authentication on workstations participating within a domain.
- Server - Provides RPC support as well as file and print sharing.

- Spooler - Provides print spooler capability.
- Workstation - Provides network connections and communication.

These services can be started, and stopped from the command line using the following commands:

```
net start <service>
```

```
net stop <service>
```

These services can also be scheduled to start manually or automatically when the NT machine starts. By using the srvmgr.exe command a GUI can be used to access which services are currently running, and from there can be manipulated accordingly.

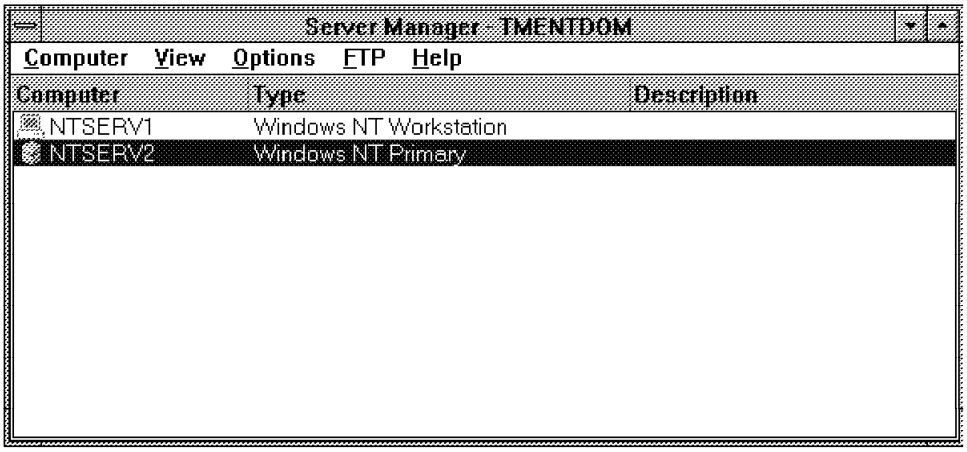


Figure 117. NT Server Manager Program

By selecting **Computer** and then **Services** from the Server Manager window, the services available for manipulation are displayed. These can be started and stopped manually or configured for startup.

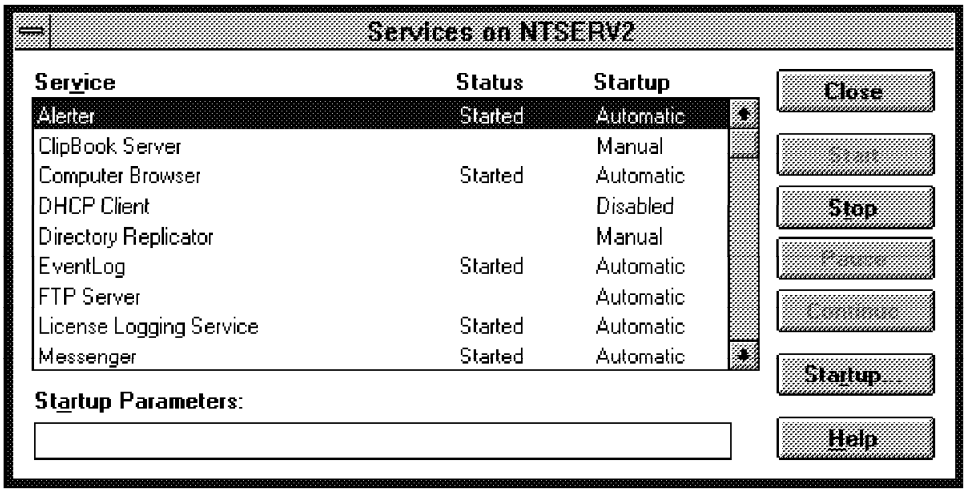


Figure 118. Services for Manipulation

4.2 TME NT Environment

We now discuss how our NT environment is set up and how it accommodates the Tivoli Management Environment. The complete environment consists of two NT 3.51 Servers, various Microsoft family operating system clients, IBM operating system clients with the NT servers doubling up as TME servers, and all the clients as TME agents. Within the environment, there is also an RS6000 AIX machine which is also defined as a TME server.

4.2.1 TME NT Domain Configuration

Our NT domain configuration, consists of two NT servers following a single domain model. One domain is all that is required here as we have a very small network located at one site. Within this domain, we have configured one server as a domain controller and a second server as an additional server to the domain. The following diagram illustrates the single domain model used.

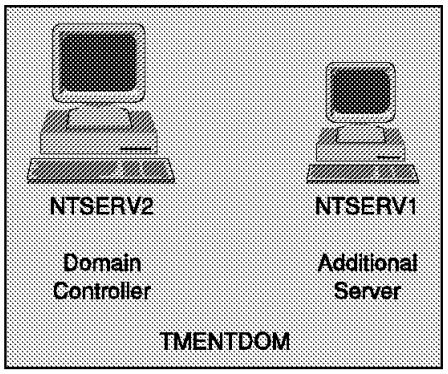


Figure 119. Our NT Domain Configuration

Although this domain looks very limited, we can still access other domains from different vendors. From NT we can make available shared files and printers from NetWare file servers and from OS/2 LAN Server file servers, as well as from NT and AIX. In fact any server that supports UNC (universal naming conventions) can be accessed.

Figure 120 on page 99 shows the two following networks:

- Microsoft Windows Network
- NetManage Chameleon 32NFS.

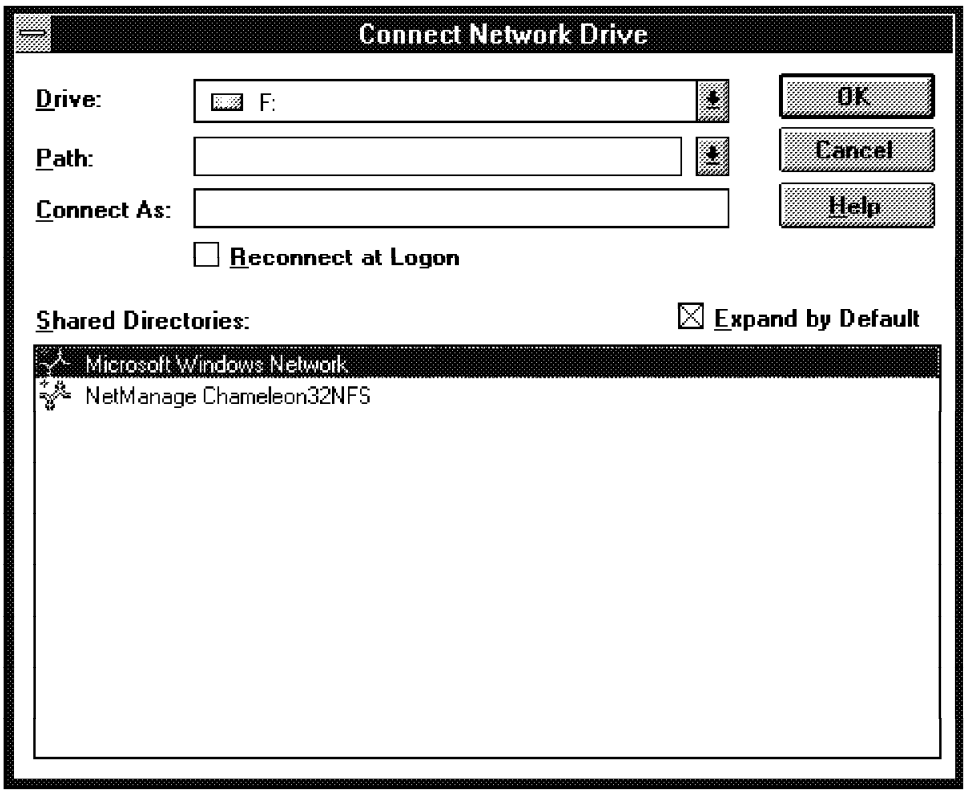


Figure 120. Networks Recognized

The following diagram shows these networks expanded, giving us access to NT domains, OS/2 domains, Windows for Workgroups as well as access to an AIX machine using NFS, which will be treated as a remote drive.

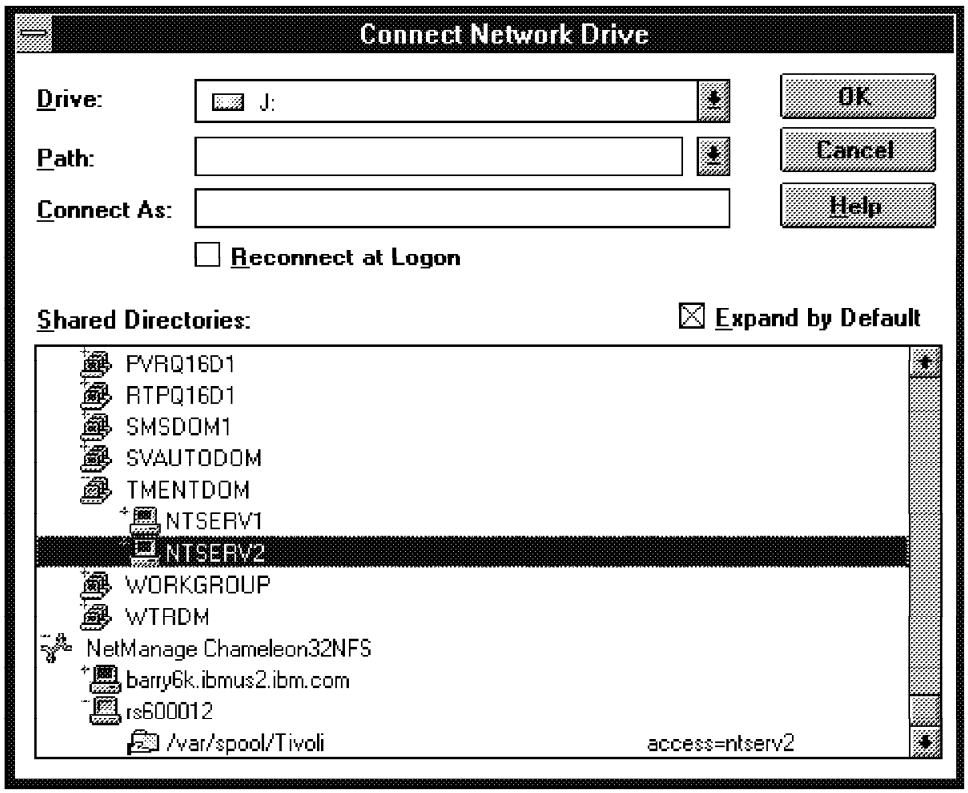


Figure 121. Visible Networks Domains

Access to the various share points on different domains will depend upon correct access rights being granted, or a suitable password being entered.

The following example takes us through connecting NT to OS/2 LAN Server and NFS mounted drives, then displaying the final connections.

The connections can be done either from the command line or by using the File Manager program. We shall first look at how to accomplish the connections to NFS mounted shares.

Once the NFS software from Chameleon has been installed, all that has to be done is to configure NFS for the local system. This is done from the NT control panel by selecting **NFS Config**.

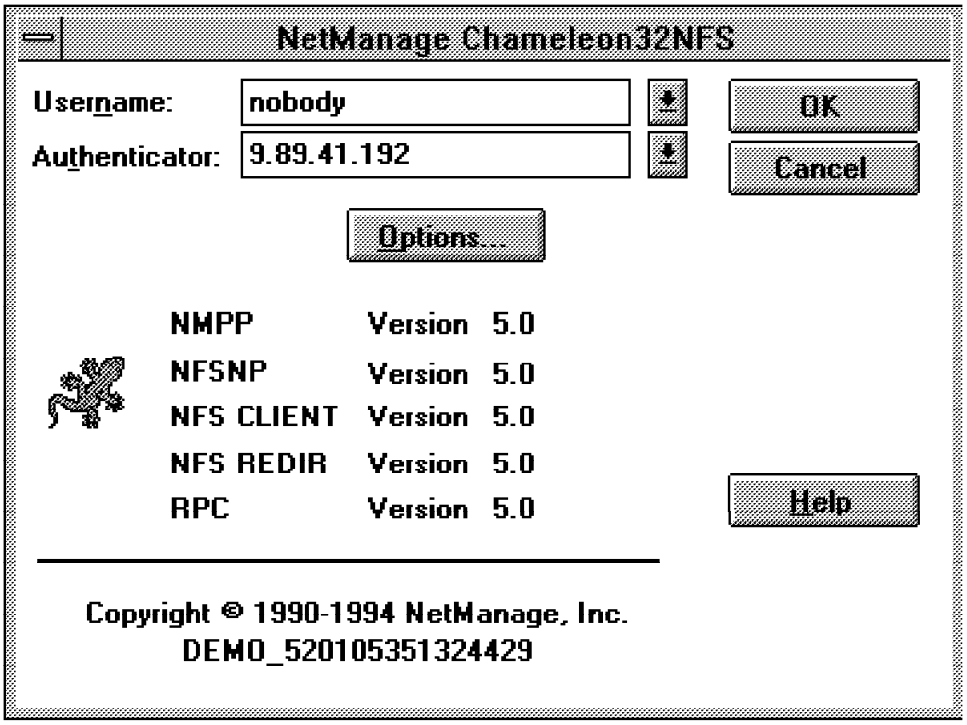


Figure 122. NFS Configuration

By selecting the correct authentication, that is, the machine with the mount established, to access this mount from NT, you can issue the commands:

```
net use x: \\9.89.41.192\mnt\reskits\winnt351.res

or

net use x: \\barry6k.us2.ibm.com\mnt\reskits\winnt351.res
```

Figure 123. NFS Mounts

This gives access to the NFS mount on a remote AIX machine, using a locally defined network drive labelled x. Another way is from the NT File Manager by selecting **File**, **Disk** and then **Connect Network Drive** and selecting the desired drive and shared directory.

To connect to the OS/2 LAN Server shares, a similar command can be issued from NT:

```
net use y: \\wtras2\lanmgmt /user:wtrdm\forsythj
```

This then prompts you for a password, unless the user ID exists under the OS/2 domain or unless you already have a valid connection to this domain.

4.2.1.1 Setting Up TME Environment Variables

The NT servers require certain environment variables to be set up, so that the servers can access the TME files, including binaries and database files. There are a number of ways these variables can be defined within NT.

Upon installation of the TME platform and desktop, there is a file placed in c:\winnt35\system32\drivers\etc\Tivoli called setup_env.cmd.

```
@echo off
rem (C) COPYRIGHT TIVOLI Systems, Inc. 1994-1996
rem Unpublished Work
rem All Rights Reserved
rem Licensed Material - Property of TIVOLI Systems, Inc.
rem
rem Component Name: 3.0 Installation shell Environment
rem
rem $Date: 1996/05/03 14:59:06 $ end
rem
rem $Revision: 1.19.2.13 $ end
rem
rem Primary Author: Francis Sullivan
rem
set BINDIR=D:\Tivoli\bin\w32-ix86
set DBDIR=D:\Tivoli\db\ntserv2.db
set o_dispatch=94
set INTERP=w32-ix86
set PERLLIB=%BINDIR%\tools\lib\perl
set TivPath=%BINDIR%\bin;%BINDIR%\tools;%BINDIR%\ADE;%BINDIR%\AEF
set Path=%TivPath%;%Path%
set TMP=%DBDIR%\tmp
set TEMP=%DBDIR%\tmp
set NLSPATH=D:\Tivoli\msg_cat\%%L\%%N.cat
echo Tivoli environment variables configured.
echo on
```

Figure 124. setup_env.cmd File from Platform Install

Instead of having to run this every time you wish to access the TME commands and utilities, we recommend that these variables are added to either the system environment variables, or the user environment variables for the administrator user. These can be accessed by selecting the system option from within the NT control panel.

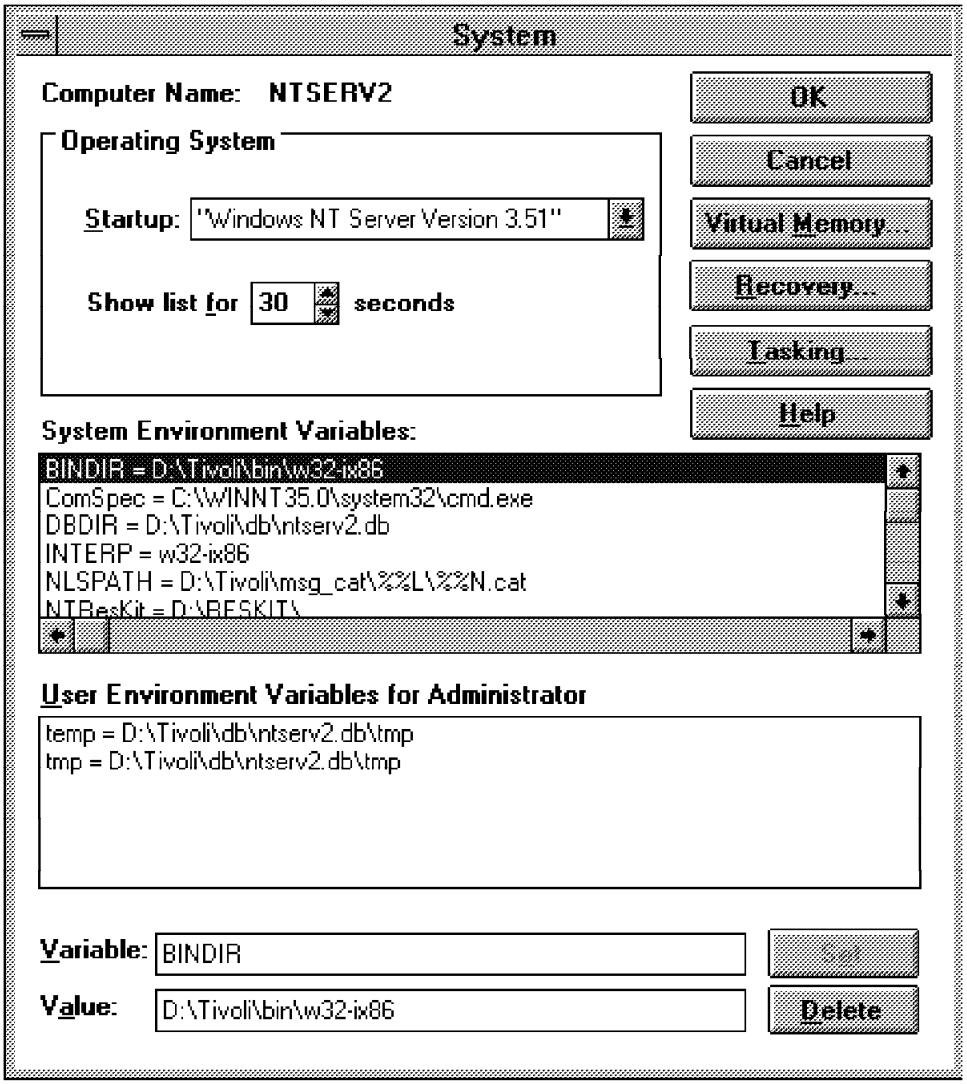


Figure 125. NFS Configuration

4.3 Microsoft Operating System Family Positioning

Here, we attempt to clear up some of the queries pertaining to which Microsoft operating system should be used, and for what purpose. Also what the major differences are between NT and Windows 95. Where should the different operating systems be deployed within an organization? We have already discussed the networking issues involved with NT and we have also noted that NT will inter-operate quite effectively with other network operating systems such as OS/2 LAN Server and NetWare.

The previous Microsoft operating systems such as Windows 3.1 and Windows for Workgroups 3.11 have been omitted here, although their operating system differences are listed later on.

4.3.1 NT Server

NT Server V3.51 is the server version of the NT operating system. It is a 32-bit preemptive multitasking operating system, with built-in security, designed to be the Microsoft production server operating system. It consists of centralized management functions for file and print sharing and security. It is a scalable operating system, capable of running on a myriad of platforms and supporting symmetrical multiprocessing.

The operating system supports different file systems including NTFS, OS/2′s HPFS and FAT, and comes with built-in networking. The NT Server operating system is designed for the most demanding business needs.

It is recommended to run NT on a minimum Intel 486 PC with 12 MB RAM.

4.3.2 NT Workstation

NT Workstation is the workstation version of the NT operating system. It has all the features of NT Server, except for the centralized management functions and has less security and fault tolerance. NT Workstation supports all the file systems that NT Server does and also comes with built-in networking facilities.

4.3.3 Windows 95

Windows 95 is the desktop operating system that Microsoft hopes will replace the older Windows 3.1 front-end to DOS. It is a newer operating system that supports 32-bit applications, has built-in networking and a number of other features. Windows 95 is intended to be the operating system choice for nonpower users such as engineering or technical users.

It is also intended for mobile users, or users where the value of NT′s reliability and security functionality outweigh cost of hardware and compatibility issues to the user.

For example, there is no point handing out NT Workstation platforms to users who use basic word processing facilities run remotely from a server, and there is also no sense in handing Windows 95 hardware and software platforms to CAD users or actuaries who possibly carry sensitive data and require extra processing power. They could utilize the extra reliability and security featured of NT.

The following table shows current Microsoft operating systems and compares the differences between them.

Table 3 (Page 1 of 2). A Comparison of Windows NT and the Rest of the Windows Family

Feature	**16-bit Windows 3.1**	**Windows for Workgroups**	**Windows 95**	**Windows NT**
Virtual memory	Yes	Yes	Yes	Yes
Multitasking	Coop	Coop	Preemp	Preemp
Preemptive Multitasking for 16-bit apps	No	No	No	No
Preemptive Multitaskingfor 32-bit apps	No	No	Yes	No
Multithreading	No	No	Yes	Yes
Symmetric multiprocessing	No	No	Yes	Yes
Portability	No	No	No	Yes

Table 3 (Page 2 of 2). A Comparison of Windows NT and the Rest of the Windows Family

Feature	16-bit Windows 3.1	Windows for Workgroups	Windows 95	Windows NT
Access security	No	No	No	Yes
Runs 16-bit real-mode Windows applications	Yes	Yes	Yes	No
Runs 16-bit standard-mode Windows applications	Yes	Yes	Yes	Most
Runs 16-bit enhanced-mode Windows applications	Yes	Yes	Yes	Yes
Runs 32-bit Windows applications	No	No	Yes	Yes
Runs OS/2 applications character mode only	No	No	No	1.x
Supports POSIXiprocessing	No	No	No	Yes
Supports DOS FAT	Yes	Yes	Yes	Yes
Supports OS/2 HPFS	No	No	No	Yes
Supports NTFS	No	No	Yes	Yes
Built-in networking	No	Yes	Yes	Yes
Built-in E-mail	No	Yes	Yes	Yes
386 or higher CPU required	No	Yes	Yes	Yes
Supports RISC chips	No	No	No	Yes
Supports multiprocessors	No	No	No	Yes
Fault tolerance	No	No	Yes	Yes

The following table shows the major differences between the Microsoft operating system family members, NT Workstation 3.51 and Windows 95.

Table 4 (Page 1 of 2). Differences of Window 95 and Windows NT Workstation

Product Feature	Windows 95	Windows NT Workstation
Application Support		
System resource capacity.	Greatly expanded	Unlimited
Runs MS-DOS applications.	Yes	most(1)
Runs IBM Presentation Manager (through 1.3) & POSIX 1003.1.	No	Yes
Application and Data Protection		
Preemptive multitasking for Win16 applications.	No	Yes
System completely protected from errant Win16 and Win32 applications.	No	Yes
NTFS file system provides complete protection of files on a stand-alone system. (Files, folders, and applications can be made invisible to specific users.)		
Has automatic recovery from a system failure.	No	Yes
System and Peripheral Requirements and Support		
Runs MS-DOS device drivers.	Yes	No
Runs Win16 device drivers.	Yes	No
Minimum recommended RAM.	8 MB	12 MB

Table 4 (Page 2 of 2). Differences of Window 95 and Windows NT Workstation		
Product Feature	**Windows 95**	**Windows NT Workstation**
Typical disk space requirement.	40 MB	90 MB
Runs on PowerPC, MIPS, and DEC Alpha AXP-based RISC systems.	No	Yes
Supports multi-processor configurations for scalable performance without changing operating system or applications.	No	Yes

4.4 Third-Party/Extended Features

Within our NT environment a number of features were used, external to basic NT. We list these now and discuss any relevant points.

4.4.1 NT 3.51 Resource Kit

The Windows NT 3.51 Resource Kit was installed on both NT Servers within our environment. The following features were found to be useful.

4.4.1.1 Net Watch

The Net Watch facility allows you to watch the different shares available, which shares have connections, and from where these connections are being made. The display options can be configured to display only In Use Shares, Open Files or Hidden Shares. Net Watch can be used on OS/2 LAN Server shares (alias) with the same effect, so long as the user net watching has the correct privileges on the required domain.

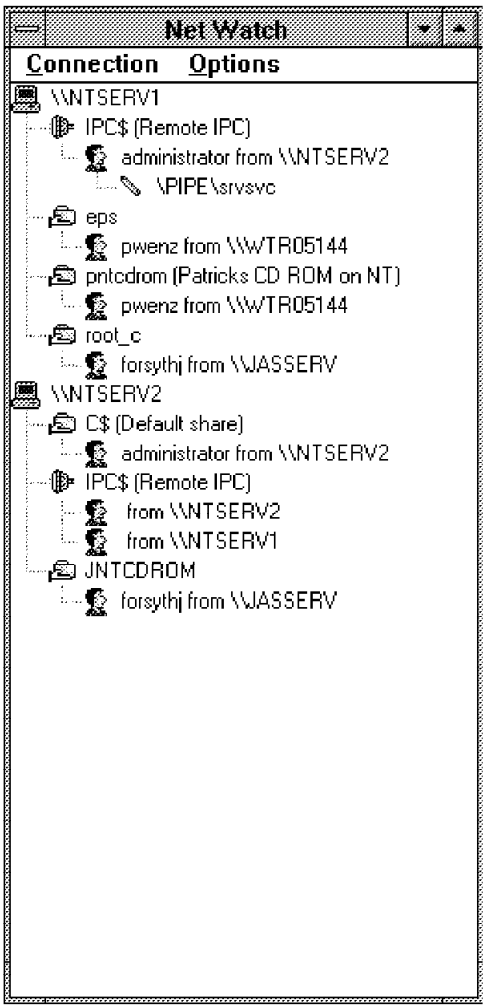


Figure 126. NT Resource Kit - Net Watch

4.4.1.2 Domain Monitor

The Domain Monitor facility enables the monitoring of NT domains, returning information about the domain controller name, state, status and synchronization details, as well as the link status to any trusted domains.

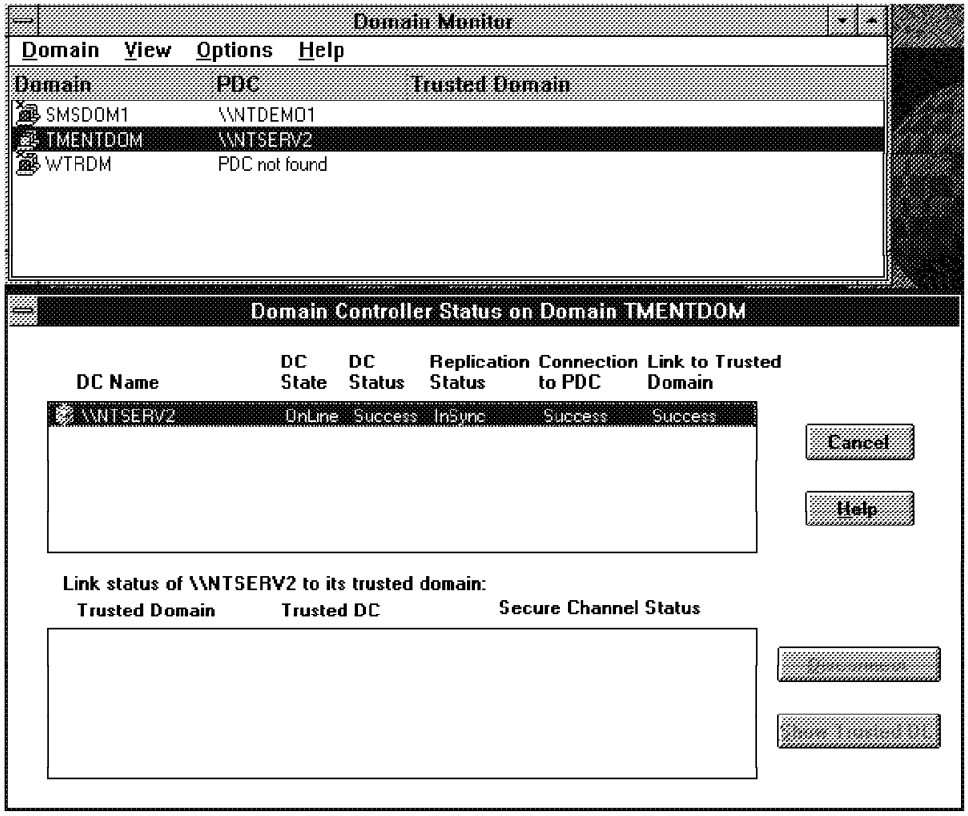


Figure 127. NT Resource Kit - Domain Monitor

4.4.1.3 Browser Monitor

The Browser Monitor facility enables the monitoring of browsers on selected domains. Each domain has its master browser and a transport mechanism (how it manages to browse and find other machines) defined. It retrieves information based on the user's name and password, so if that name and password do not exist on a domain, the request will fail.

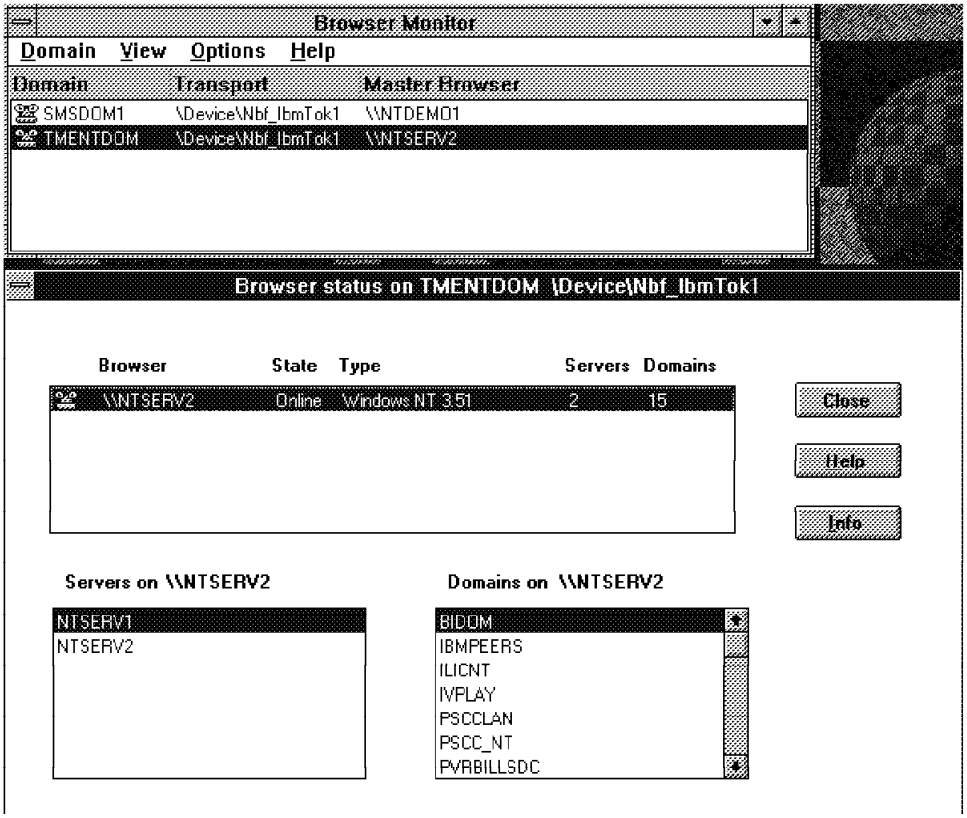


Figure 128. NT Resource Kit - Browser Monitor

4.4.1.4 Process Viewer

The Process Viewer enables viewing of all the system processes that are running on a local or remote NT machine. This utility allows processes to be prioritized or killed. It also allows a closer look into the memory details of each process.

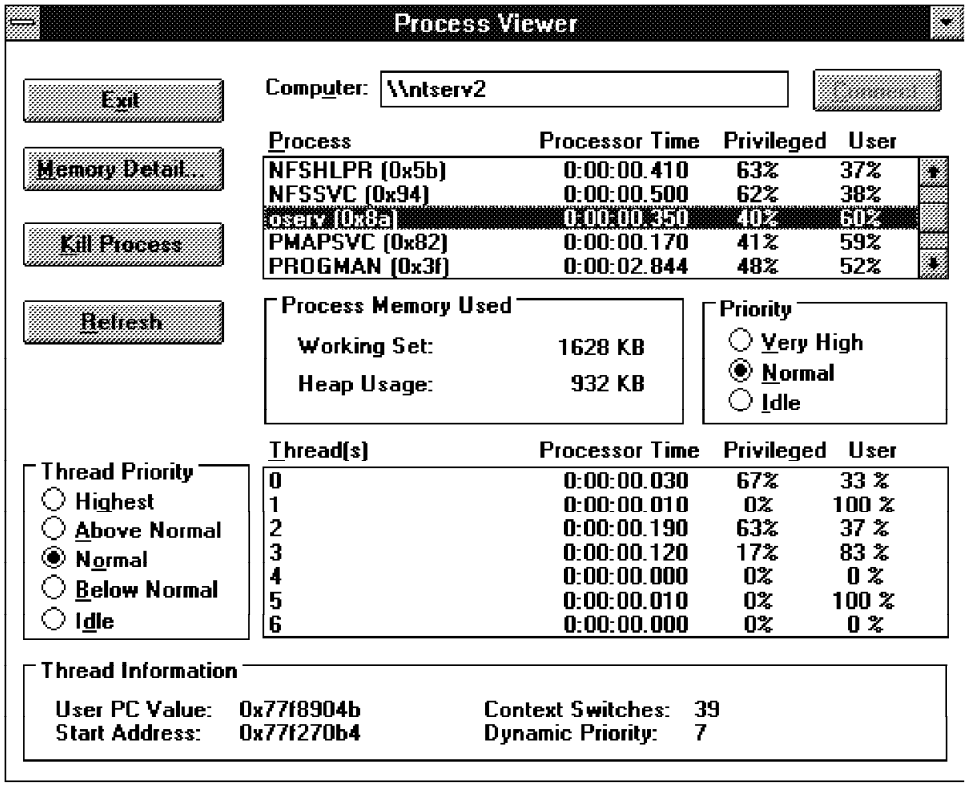


Figure 129. NT Resource Kit - Process Viewer

4.4.1.5 QuickSlice

The QuickSlice facility graphically displays the process identification number and corresponding process name, with the percentage of CPU that process is using graphically. By selecting each running process, that processes utilization can be divided down to each executing thread.

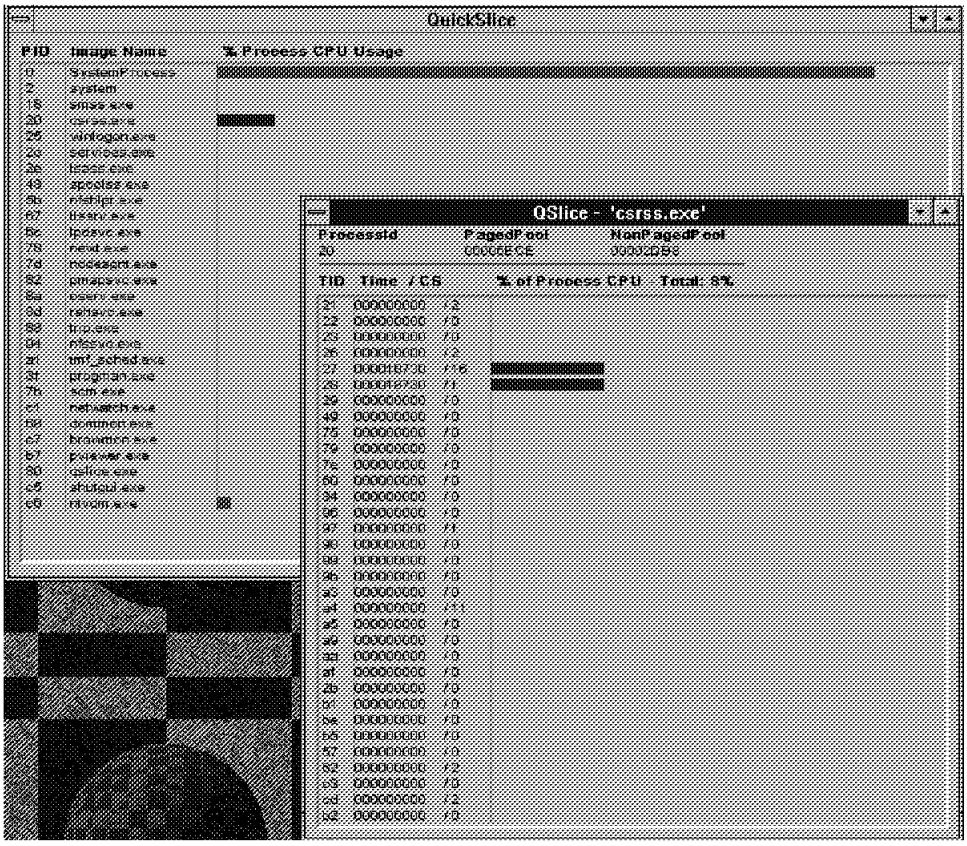


Figure 130. NT Resource Kit - QuickSlice

4.4.1.6 Shutdown Manager

The Shutdown Manager enables the remote shutdown of other NT servers and workstations from a central site. The possible options that can be set are to reboot after shutting down, or shutting down without saving open data within open applications. There is also a message text box, which can be edited to reflect the desired message to be broadcast. Also, a time delay can be set before executing the shutdown.

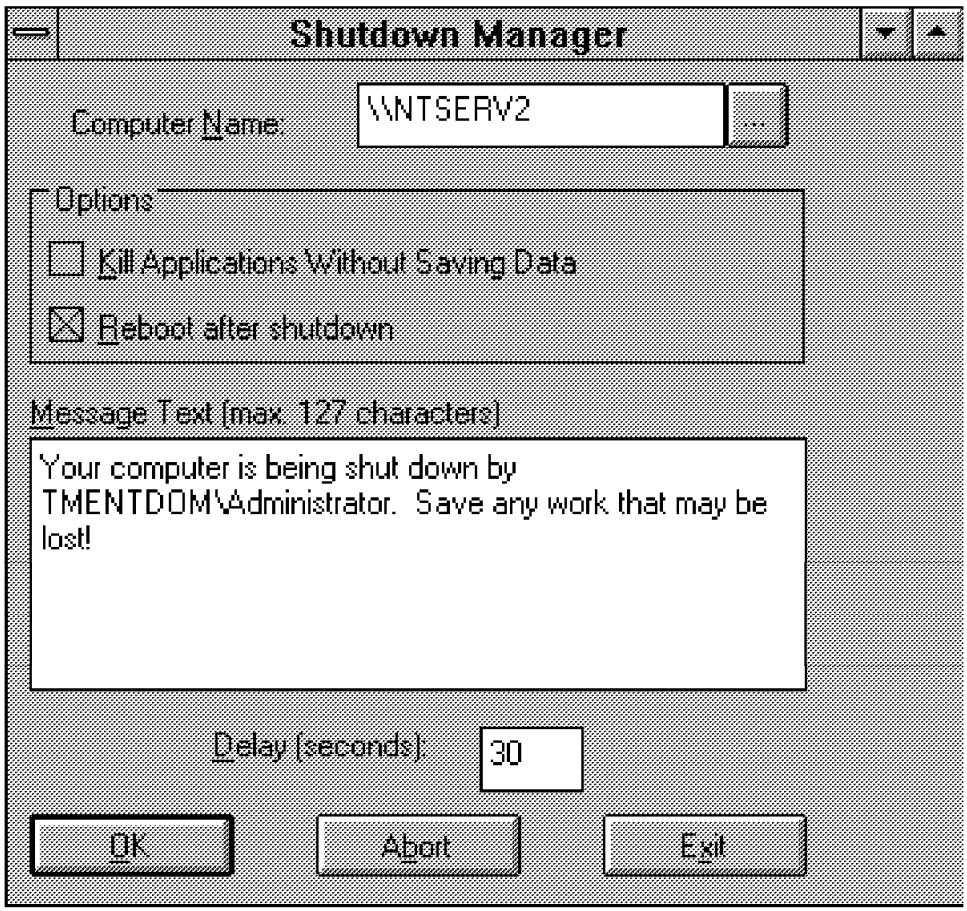


Figure 131. NT Resource Kit - Shutdown Manager

4.4.1.7 Backing Up the Registry

The Registry Backup utility enables backup of the registry to a hard or floppy disk. By running the command regback.exe from the command line with a location as a parameter, a backup can be achieved.

```
regback  c:\reg_27jun

saving SECURITY to c:\reg_27jun\SECURITY
saving SOFTWARE to c:\reg_27jun\software
saving SYSTEM to c:\reg_27jun\system
saving .DEFAULT to c:\reg_27jun\default
saving SAM to c:\reg_27jun\SAM
saving S-1-5-21-87058076-1083533151-732247886-500 to
c:\reg_27jun\Admin000
```

Figure 132. NT Registry Backup

4.5 Administration under Windows NT 3.51

This section shows how to set up the TME environment. In addition, some examples of using the command line interface instead of the GUI are provided.

Before we talk about the administration of Tivoli TME 3.0 we want to show how the administration under Windows NT Server works. Windows NT itself provides system management functions that can be used on top or in addition to the Tivoli administration functions. There are two categories of tools:

- Tools for the hardware and the machines
- Tools for the users

4.5.1 The Hardware Tools

These tools include all functions used to configure the local system or remote client systems. Because of the number of these tools, we took the most useful tools for our view of the administration.

4.5.1.1 Server Manager

Figure 133. Server Manager Icon

The Server Manager is integrated in the Windows NT Server environment.

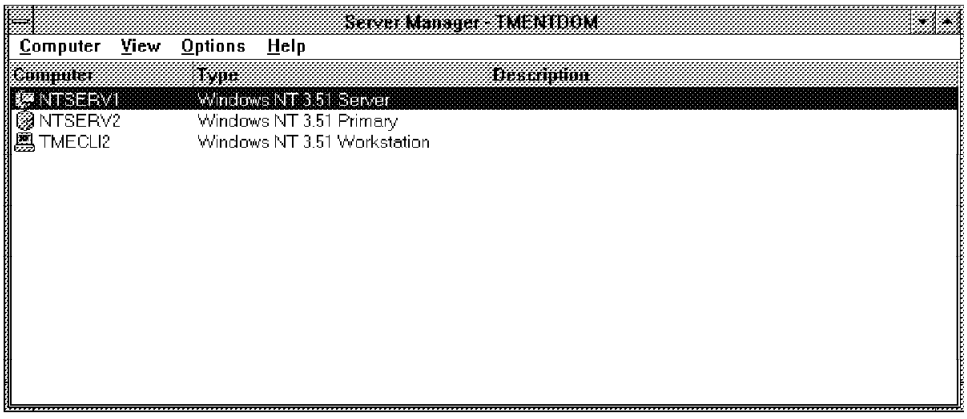


Figure 134. Server Manager Main Window

You can use the Server Manager function to manage domains and computers. The most interesting functions are:

You can:

- View a list of connected users
- View shared and open resources

- Manage directory replication
- Manage the list of administration alert recipients
- Manage services
- Manage shared directories

In more detail these are the functions of the Server Manager which is started with the srvrmgr command:

- Properties of a specific server or workstation. You can get these facts by opening the Properties window by selecting the **Computer** pull-down menu.

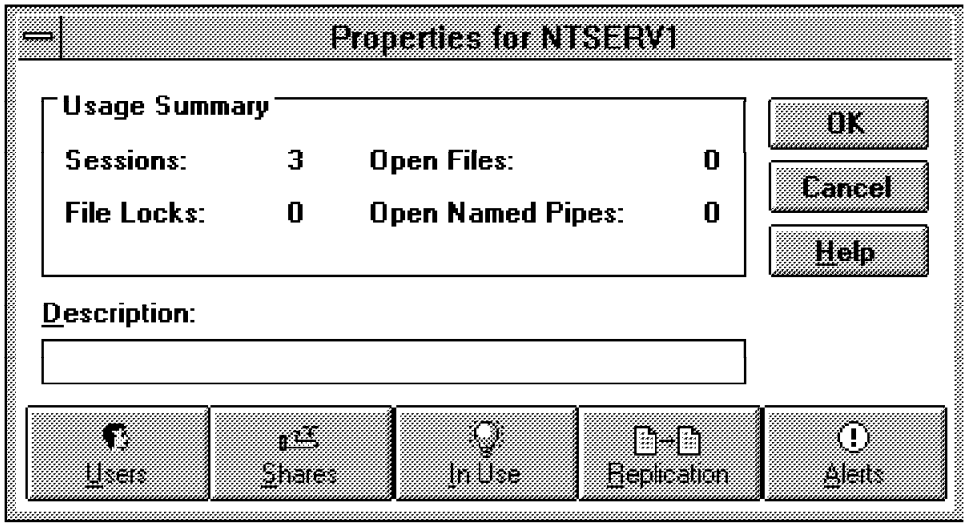


Figure 135. Properties Window

The Properties window contains a list of resources used on the current NT Server, such as sessions, open files, file locks and open named pipes. In this window you have the ability to make some more changes for the system you are working on. The choices are listed in the button bar in the window. In the following paragraphs we talk about the three functions (buttons): Users, Shares and In Use.

1. Users

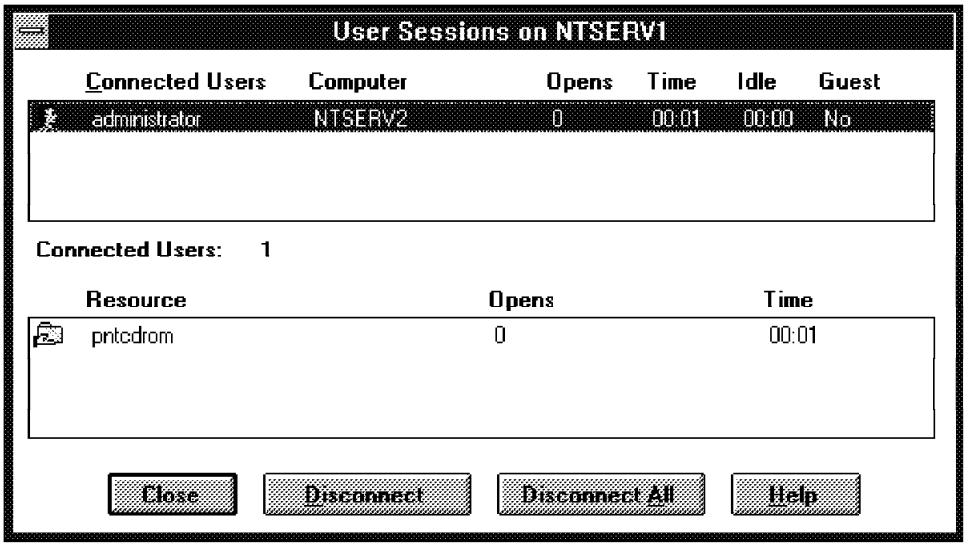


Figure 136. User Session Window

The Users Sessions window gives you an overview of the connections from the users at the NT Server (the current machine). The list shows the name of the user, the machine they are connected to, the time and a few other fields. You can disconnect a single user or all users from this panel.

2. Shares

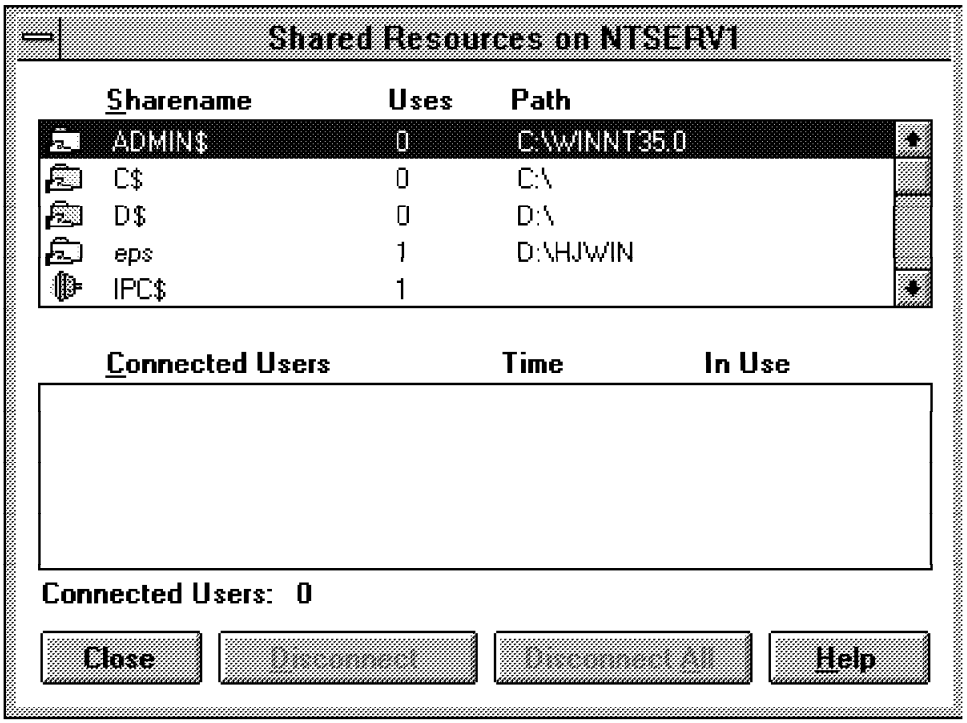


Figure 137. Shared Resource Window

The Shared Resources window presents an overview of the defined and current share names and the paths they are defined for. It also shows the number of connected users to these shared directories or

drives. Again there is the possibility to disconnect the users from a share or to disconnect everyone from the shared directories.

3. In Use

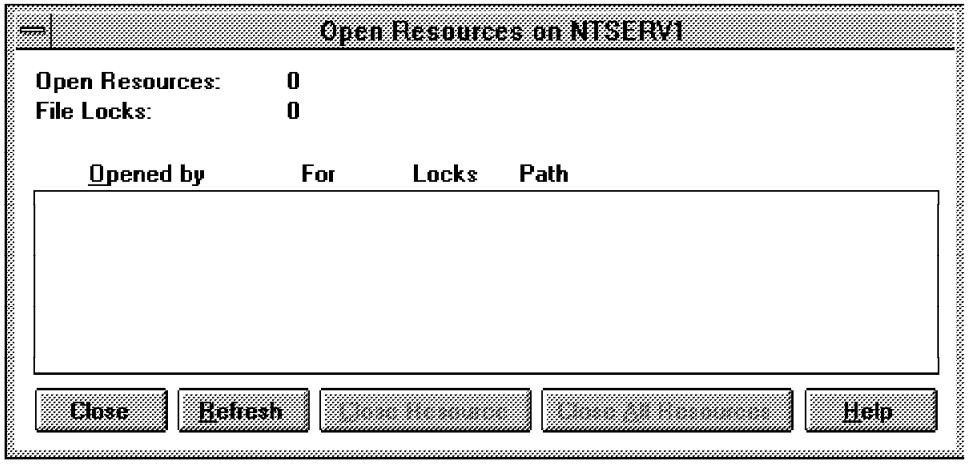


Figure 138. Open Resources Window

The In Use window shows all open resources on the current machine, who has opened them and the path of the resource. You have the opportunity to close all resources.

– *Shared Directories*

On the menu bar choose **Computer** and **Shared Directories** and Figure 139 appears.

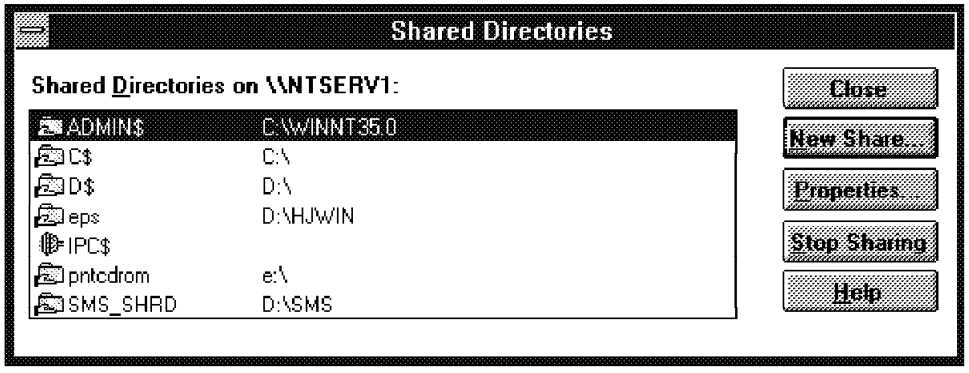


Figure 139. Shared Directories Window

It shows the current shared directories on the machine and their names. In the window you can see some push buttons on the right side. These buttons represent the actions that can be taken against the shared resources.

1. New Share

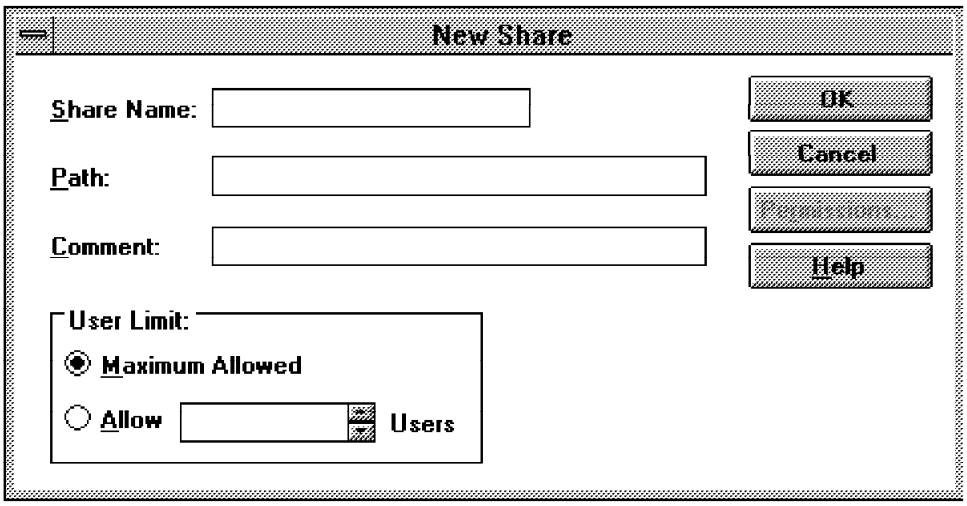


Figure 140. New Share Definition Window

The New Share function is very important for the network. This function allows the administrator (or the user) to share resources with other users. In the New Share window you have to fill in a name for a new share (alias) and the path on the machine.

After that you can change the permissions of the alias you added. To perform this action, you have to select the **Permission** button.

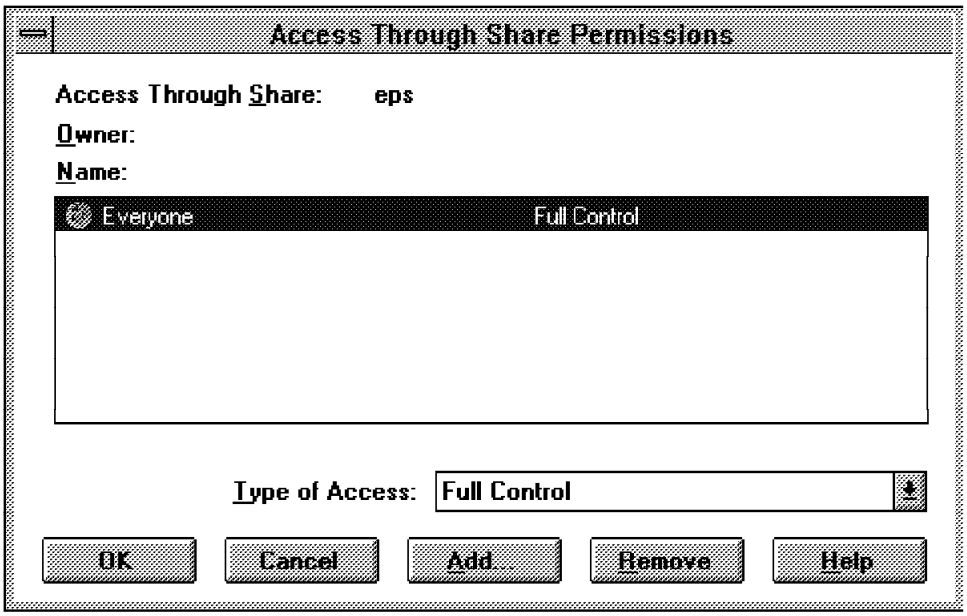


Figure 141. Access through Share Window

The next window that appears is the Access Through Share Permissions window. It shows an overview of the people who have access to the defined share resource. The default is Everyone. In the list box at the bottom of the window, you can choose the access type. There are four types:

- Full Control (default)
- Read

- Change
- No Access

For adding other users or user groups to the new share you have to select the **Add** push button. This leads you to the Add Users and Groups window.

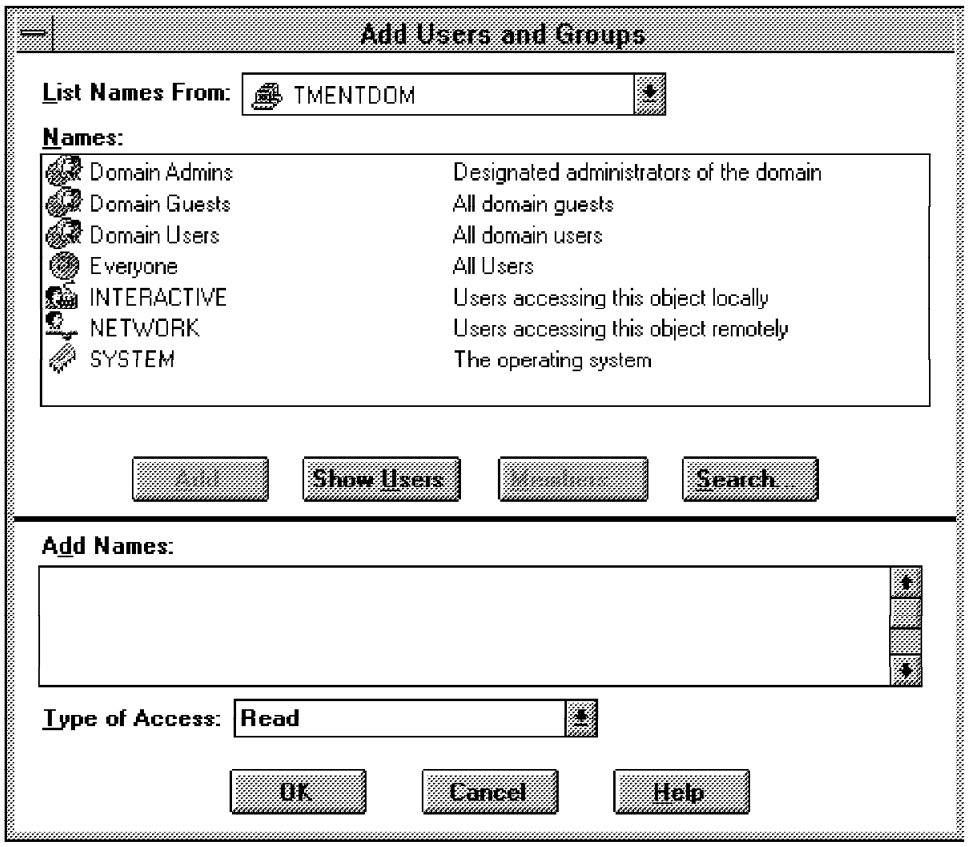


Figure 142. Add Users and Groups

This window offers you several options to define users to share the resource:

- The list box at the top of the window shows you the domain or the server from which you can select the users or user groups.
- Depending on the server or the domain, the box beneath shows the available user names or user groups for the selected domain or server.
- If you want to select a user, you first have to choose the **Show Users** button. This performs an action that shows all users that belong to the selected domain.
- Another way to select a user who belongs to a user group, is to select the **Members** button. The window that appears shows all users of the group.

Figure 143. Member Window

You can choose one (or more) members and add them to the users that can have access to the share. The newly added user appears in the box in the lower part of the window.

- After the selection of the users or user groups you can define or change their type of access. Again there are four types of access (see above). You can define an access type for the selected users or user groups.

Note

If you want to define different user accesses for different users, you have to do this for every user one at a time. For example, Paul is a member of the administrator group. You want to give him read access. Archie is a member of the administrator group, too. You want to give him full access. So first you have to finish the add process for Paul and then start a new one for Archie.

2. Properties

   The Properties button in Figure 139 on page 116 takes you to the Share Properties window.

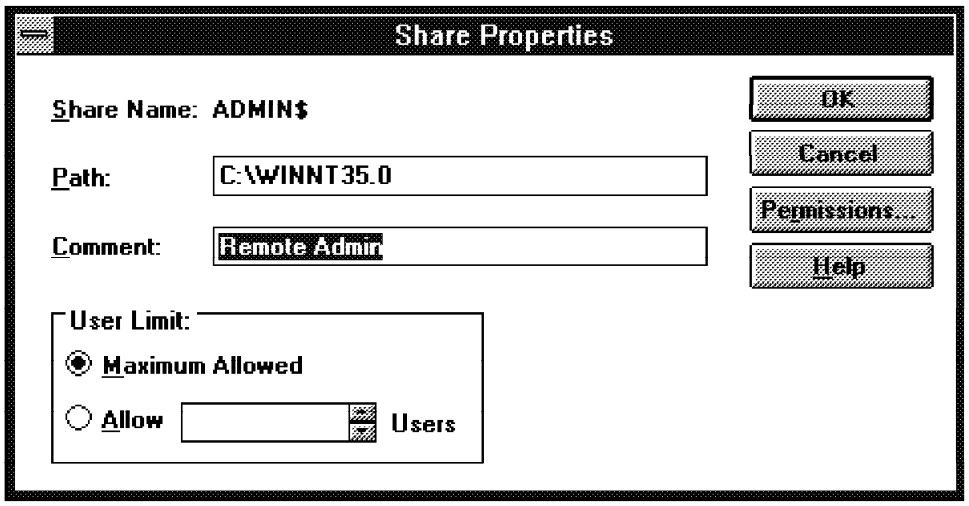


Figure 144. Share Properties Window

In this window you can change the description and the path for the share resource. You also have the chance to change the permissions as described above. That means you can change the permission for each shared resource.

4.5.2 Administative Tools

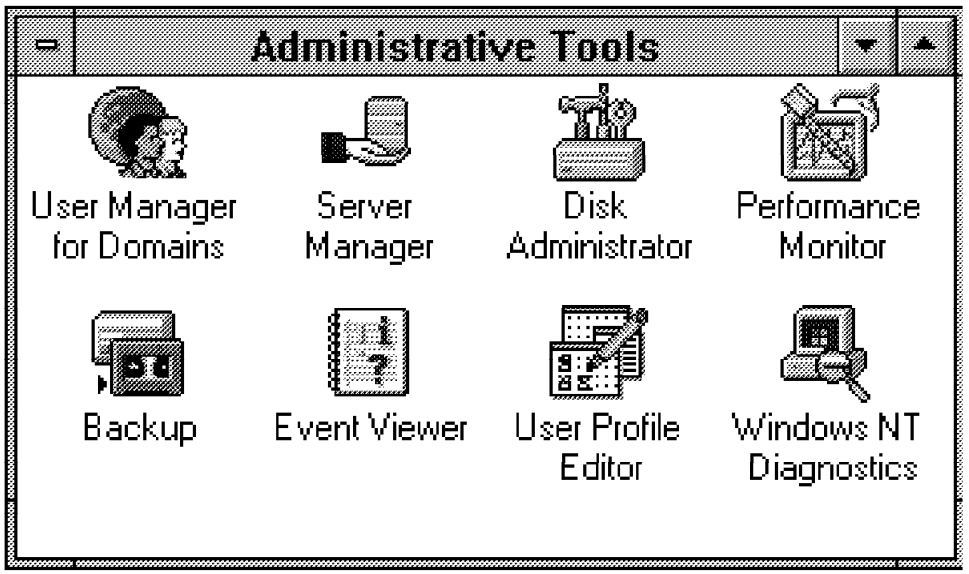


Figure 145. Administrative Tools

The user tools are very important to add or remove access to a network resource. One tool, the User Manager for Domains, covers many functions. There are several important functions integrated with this tool. You can:

- Create and manipulate local and global groups
- Create and manipulate user accounts
- Configure system-user policies

- Specify which computer the user can log on from and the times the user can be logged in
- Set up the user's profiles, login scripts and home directories

Another very important feature in the framework of Windows NT is the trusted domain and the trusted relationships. The definition of a trusted domain and access allows you to use the domain database of a foreign domain. In this case, the local domain is the trusting domain. A trust relationship is a link between two Windows NT Server domains. One domain honors the users of another domain, trusting the logon authentications performed by that other domain for its own users. You can find the definition window for the trusted domain in the User Manager from Windows NT. It is integrated into the Policies menu. Figure 146 shows what it looks like.

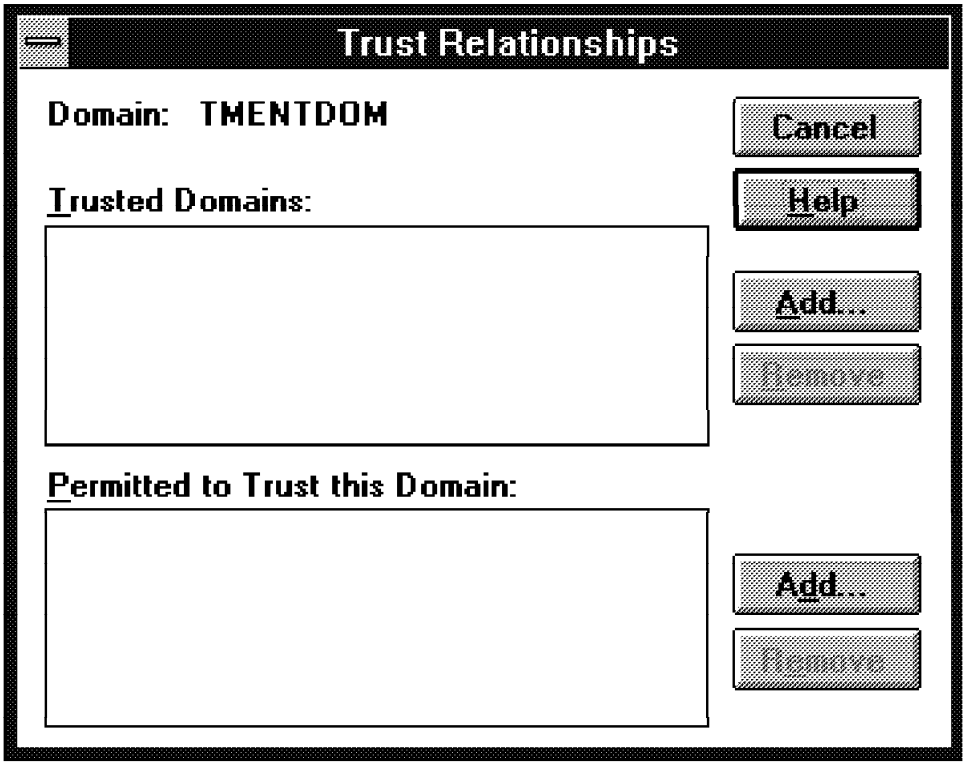


Figure 146. Trust Relationship Window

We now go into more detail on the functions that are performed from the User Manager icon.

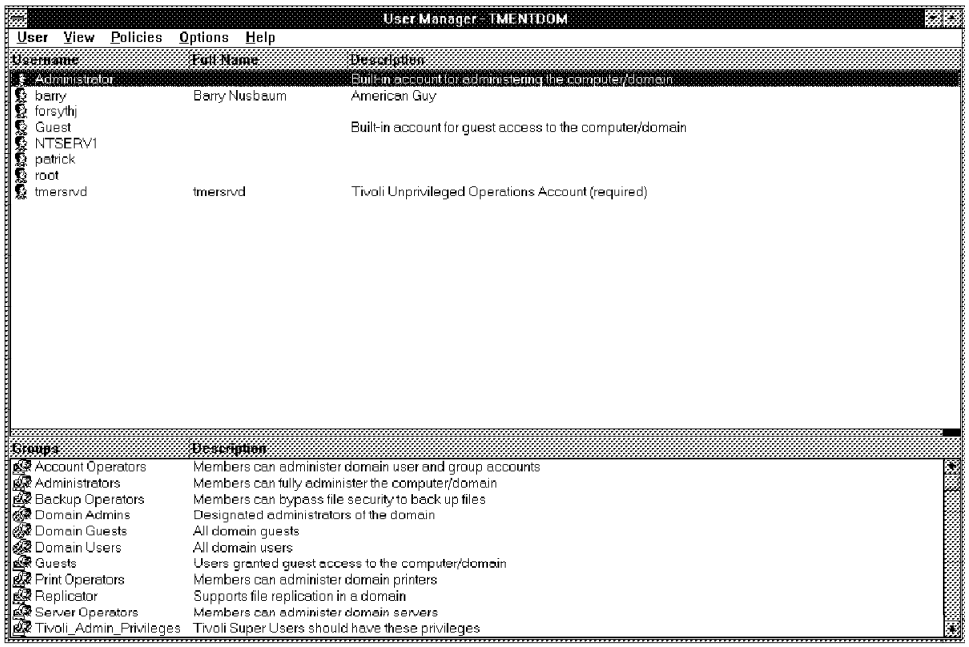


Figure 147. User Manager Main Window

In the top of the window you can see the defined users with their names. To change any user you must log on as an NT Server administrator.

In the bottom you can see the defined groups and their descriptions. Windows NT has several built-in groups. That means that some user groups are pre-defined. The groups that are built-in are:

- *Administrators* - Can administer the local computer and any domain resource (locally).
- *Account Operators* - Can administer domain user and group accounts (locally).
- *Backup Operators* - Can bypass the security restrictions on directories and files in order to back them up (locally).
- *Domain Admins* - Can administer the domain resources. These members are added automatically to the local administrator group of all domain members (globally).
- *Domain Guests* - Added automatically to the guest group (globally).
- *Domain Users* - Added automatically to the local user group (globally).
- *Guests* - Have limited access to the domain. They can't change any settings (locally).
- *Print Operators* - Administer the domain printers (locally).
- *Replicator* - Can replicate files in the domain (locally).
- *Server Operators* - Administer the servers in the domain (locally).
- *Users* - Normal user (locally).

The install of the TME causes the addition of a new group: Tivoli_Admin_Privileges. Every Tivoli administrator must be a member of this group to work with the resources. Tivoli also adds a user to the local domain

called tmesrvd during the installation of the TME. All Tivoli operations are performed under this user ID.

You can be a member of more then one group. For example, if you are the administrator of the domain, you are in the groups Administrators, Domain Admins, Domain Users and Server Operators. In the menu bar, the first option User is the most important. It permits you to:

- Add a new user, a new global or local group
- Work with the properties of a user or a group
- To select a different domain

In the following section we show the different selections from the User menu option.

4.5.2.1 New User

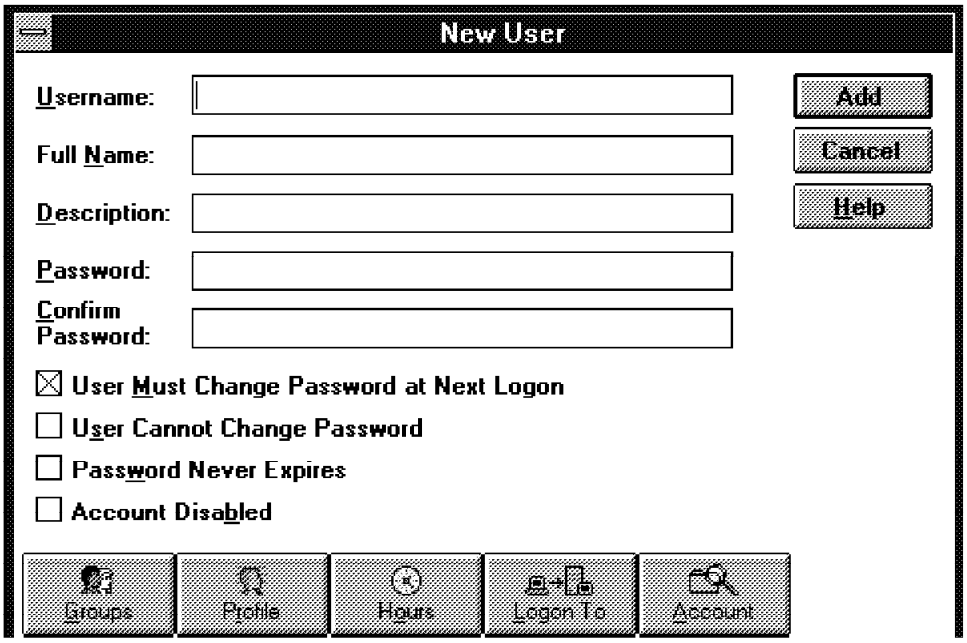


Figure 148. New User Window

The New User window is for defining a new user. You customize the following:

- Username
- Full Name (optional)
- Description (optional)
- Password (for the user)
- Confirm Password

Furthermore, there are some fields in the middle of Figure 148 where you decide about password characteristics.

On the bottom there are some buttons to help define the user:

- Groups
- Profile

- Hours
- Logon to
- Account

The functions that we spent the most time on were Groups, Profile and Account.

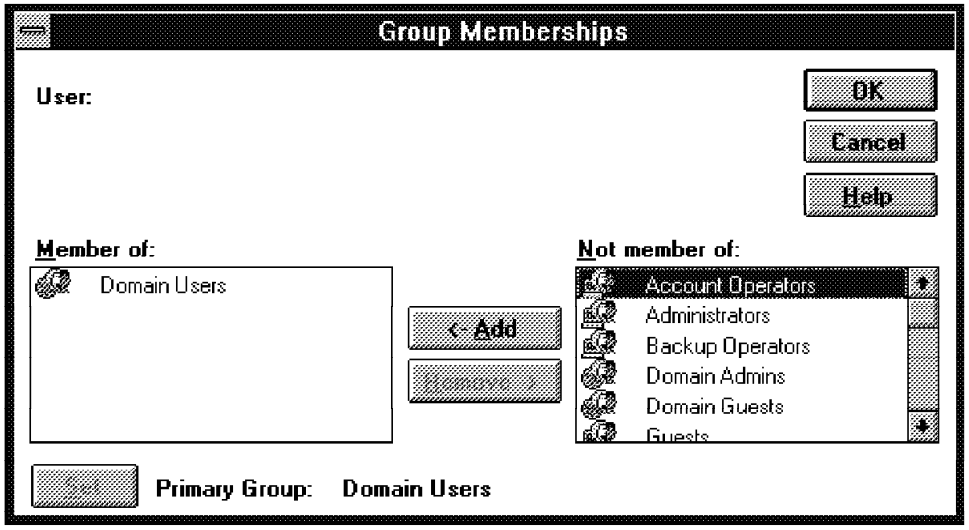


Figure 149. Group Membership Window

In the Group window you can choose user groups from the right list box and add them with the Add button to the left one. The left list box contains the current user groups the user belongs to. If the user belongs to the current domain and is just another domain user you don't have to select the Group button. You only have to work with this function if the new user belongs to a different group.

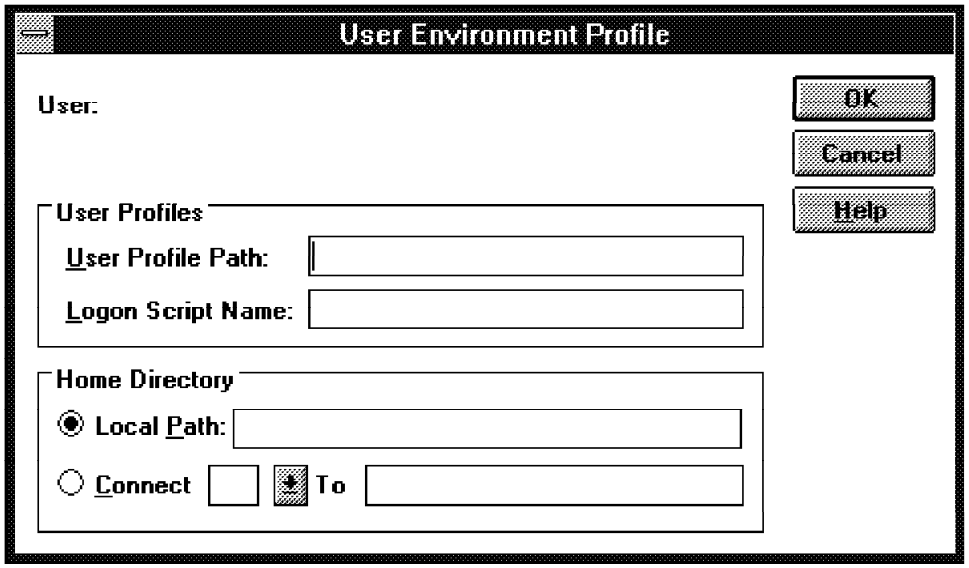


Figure 150. Profile Window

In the profile window you can set the following:

- User Profile Path
- Login Script Name

- Home Directory

Use the environment variable %UserName% for the login script name and the user profile. The %UserName% will be expanded to the name of the user during the logon process and build a fully qualified file name. The logon script runs every logon the user does. It can be a batch or an executable file. The usual directory for logon scripts is C:\WINNT35\SYSTEM32\REPL\IMPORT\SCRIPTS.

The home directory is a directory users have access to when they log on. It contains personal files, applications that have no working directory and the command prompt.

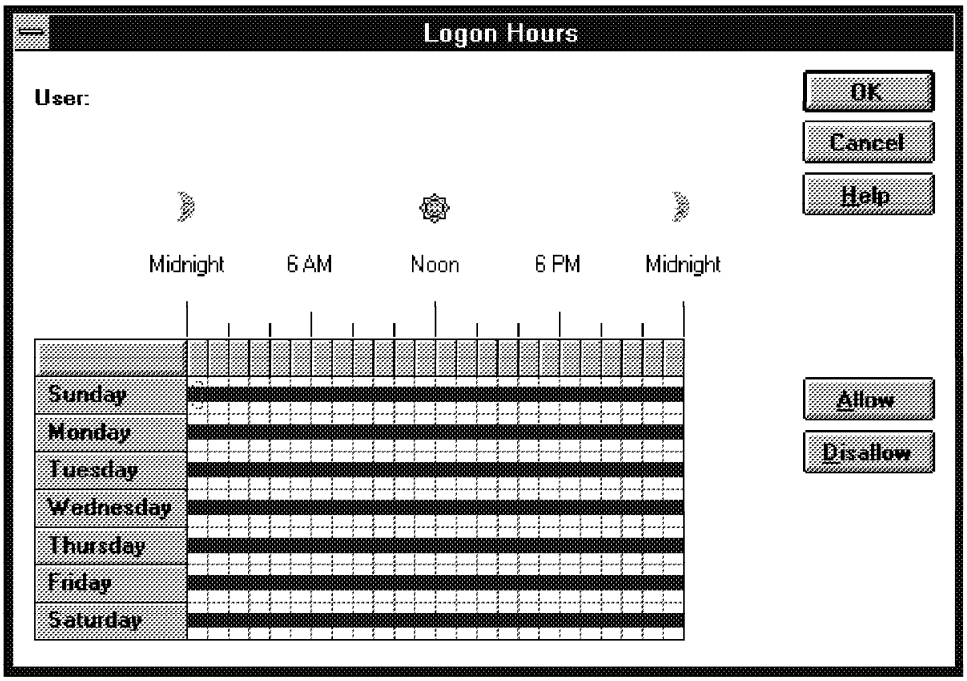


Figure 151. Logon Time Settings Window

In this window you can limit the logon hours for a user.

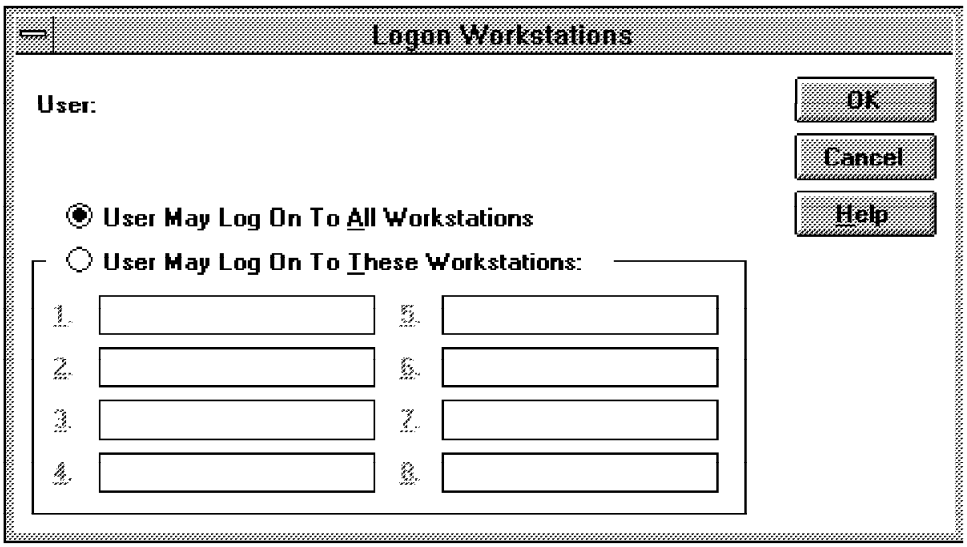


Figure 152. Logon to Window

In this window you can limit the access to eight workstations.

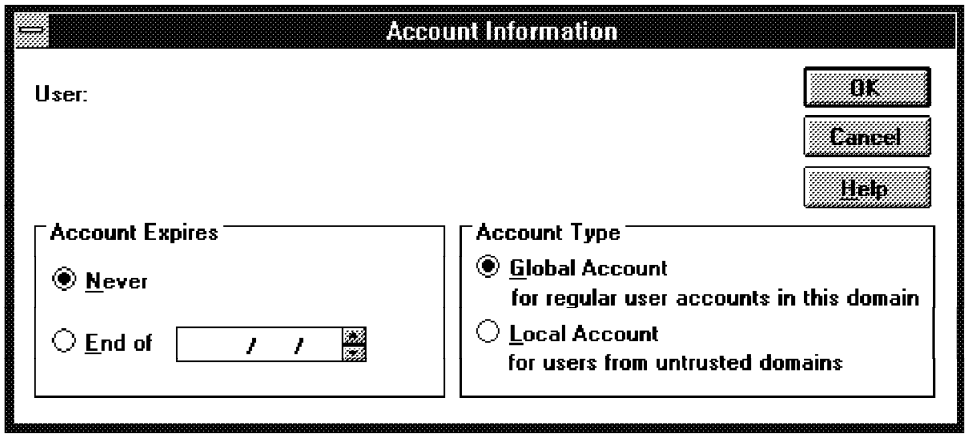


Figure 153. Account Settings Window

In this window you can specify the expiration date of the account and also what the account type is, Global or Local.

4.5.2.2 Properties

There are two ways to see the properties of a user or a user group:

- Select the **Properties** option from the User pull-down menu.
- Double-click on the user or the group.

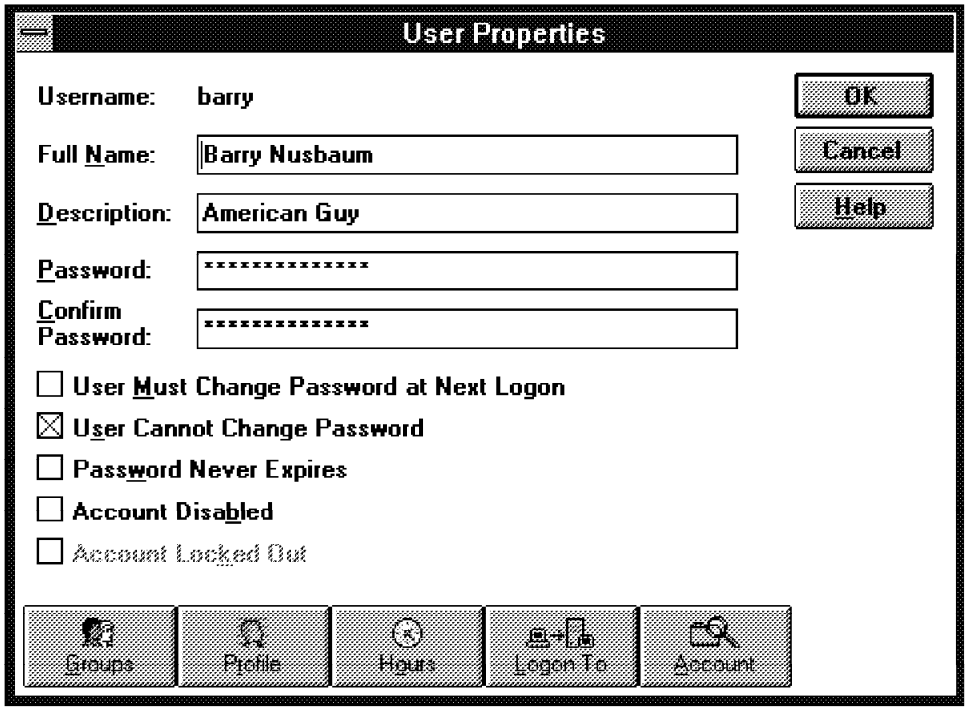


Figure 154. Properties Window

Here you can change the settings for a user or a user group.

4.5.2.3 Select Domain

If you want to work with another domain or users from another domain, you can choose the option **Select Domain** from the User menu. In this window you can fill in another domain or specific computer name.

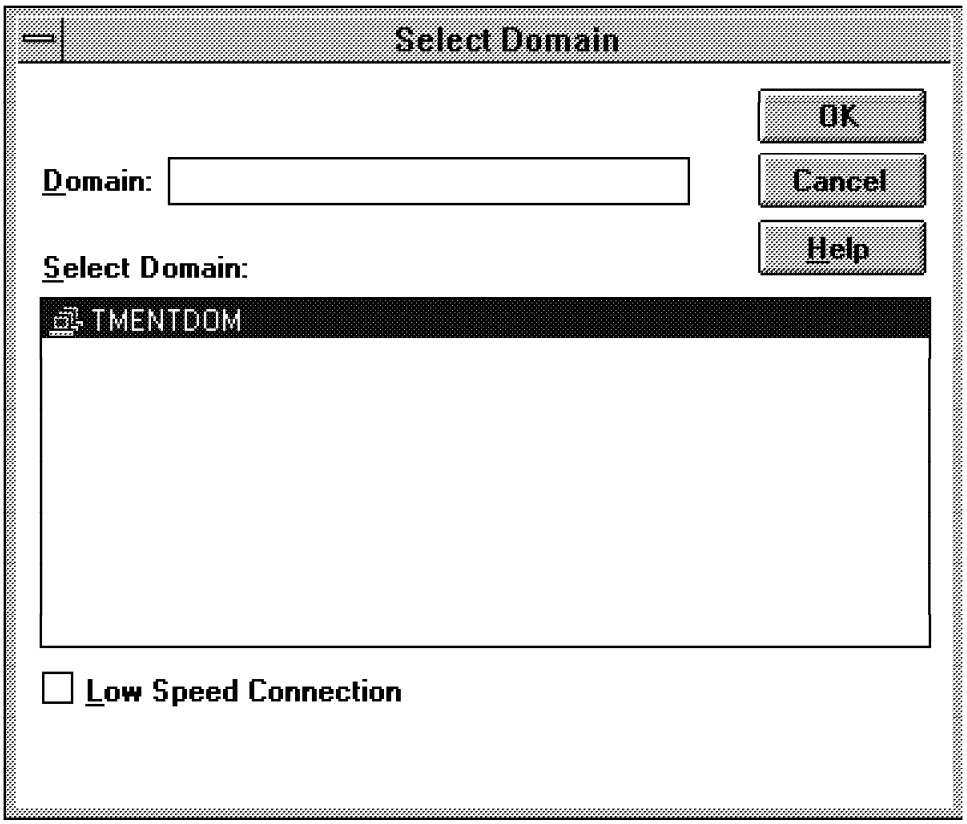


Figure 155. Select Domain Window

Chapter 5. Administration with Tivoli TME 3.0 for Windows NT 3.51

Since the installation and administration of the Tivoli-related products required some NT-based administration, this section is divided into two parts. One part covers pre-installation hints and tips and the other part is directly related to the TME platform.

5.1 Pre-Installation Planning

Before you install Tivoli TME 3.0 for NT, you have to consider several points:

- The types of machines to be managed
- Hardware and software configurations, including the operation systems
- Location of file servers
- Disk space
- Network topology and communications

It is important that you create a plan of the functions the system will be integrating. That means the requirements for the machine are different if you use it only as a TME Manager or as the TME File Server. Another very important part is the planning of the Tivoli Managed Regions (TMR). Every region has its own server (TME Server) to manage the resources. Therefore, it must be thoroughly planned how many regions you are going to install and how many servers you need. You must also have a security concept that includes the roles and the definition of the roles for the Tivoli environment. In addition, you have to decide about the integration of concepts and applications that are used in the current environment. This requires a plan for integration and migration. These are a few points to be concerned about.

5.2 Administration at the Installed System

When the Tivoli system is installed you can log on as administrator. But, if you want to give other people administrator access, you have to create new administrators under NT (User Manager) and you have to think about their roles under Tivoli TME. The role is an important part of the Tivoli framework. You can define a new administrator and give them different roles in different TMRs. You can also define different roles for resources. This might depend on what the new administrator work is focused on. Therefore, it is helpful and important that you map out all the roles before you start to implement it. Due to the complexity of this, we explain the roles and how you can create a new administrator.

5.2.1 Administrators and Other Roles

A Tivoli administrator is a system administrator that has been established as a TME administrator. The initial Tivoli administrator is added during the installation of the TME. Since root authority is required to install TME, the initial Tivoli administrator is the root administrator by default. After the TME is installed other nonroot administrators can be defined and given roles in Policy Regions. These administrators can perform assigned system management tasks without being root. Tivoli administrators can perform system management tasks and manage various regions in one or more networks. You can delegate system management tasks to administrators by:

- Assigning authorization roles to the administrators
- Copying Policy Regions between desktops
- Moving Policy Regions onto an administrators desktop
- Moving or copying system resources between Policy Regions

5.2.1.1 TMR Roles

To perform system administration activities in the Tivoli Management Environment you have to be a Tivoli administrator. Depending on what activities you are required to perform, you must have one or more roles. The different roles are:

- **Super** - An administrator with the super role can connect and disconnect TMRs and change license keys. The super role is normally required only for high-security operations and is not typically assigned to many administrators. Depending on your concept for the regions it is possible to have one administrator with the super role at each TMR.
- **Senior** - An administrator with the senior role can create and define all TME resources. The senior role is required for configuration and policy tasks such as creating an administrator, setting policy, or expiring notices.
- **Admin** - The admin role allows an administrator to perform day-to-day operational tasks and affect system configurations. The admin role is required for general system management tasks such as adding user items to a profile, pushing a file package, or changing the message of the day.
- **User** - An administrator with the user role has read-only browsing capability. The user role is required to bring up a TME desktop. Its actions are limited.
- **Restore/Backup** - The restore role is required to restore TME databases. You must have the restore role in the TMR that contains the TME server and clients that are to be restored. The same rules are valid for the backup role.
- **Install-product** - This role allows an administrator to install new products into a local TMR.
- **Install-client** - An administrator with this role can install new Managed Nodes within Policy Regions that support the Managed Node resource type.

An administrator without any role has no authorization to use TME and cannot access the desktop. The super role is automatically assigned to the administrator who installs the TME. When new administrators are added, they have no role by default. With super or senior roles you can change the roles of any administrator at any time. If you only have the admin or user role, you can only assign subordinate roles. The roles install-client and -product and backup and restore cannot assign any role.

Example (simple) - You work in an insurance company. It has 35 offices in the region near London, 20 offices in the region of Plymouth and 15 offices in the region of Wales. In every region you installed a TME 3.0 Managing Node, but you decided to place the central machine in London. You are the administrator with the roles of super, senior and admin. Your colleague in Plymouth has the roles admin and install-product and install-client. The colleague in Plymouth cannot define any role for an administrator. But the one in Wales can, because they also have the senior role.

5.2.1.2 Roles in TMRs

As administrator with the super role or the senior role you can assign roles to another administrator. With this role the administrator can perform actions that affect resources anywhere in the local TMR. If the role is other than super, the role maps across the boundaries of connected TMRs. It doesn't matter if it is a two-way or one-way connection. Only the super role doesn't map across TMRs; it maps to the role user in connected TMRs. Administrators with the super role can perform tasks that require the super role only in the TMR where the administrator was created. To assign an administrator the super role in more than one TMR you have to create separate administrators in each TMR you want to have the super role. Because of security no administrator can have the super role across all TMRs.

5.2.1.3 Resource Authorization Roles

With the super or senior role you can assign an administrator role for resources. A resource role allows an administrator to perform special management tasks on this resource.

The following table outlines the roles in installation.

Super	Senior	Admin	User	Install - Client	Install - Product	Backdrop - Restore
Update the license key for a product in the TMR			View the license key for a product in the TMR	Install TME clients (UNIX, PC Managed Nodes) and remove them from a TMR	Install a product, application, patch, upgrade in a TMR	

The following table outlines the roles in administration.

Super	Senior	Admin	User	Install - Client	Install - Product	Backdrop - Restore
	• Create/delete an administrator • Edit – An administrator's resource role - Login name - Notice group subscription - Properties – TMR authorization roles for an administrator • Set – TMR authorization roles for an administrator		Create a collection and work with resources			

The following table outlines the roles in Managed Nodes.

Super	Senior	Admin	User	Install - Client	Install - Product	Backdrop - Restore
	Move a PC/UNIX Managed Node from one Policy Subregion to another	• Add/remove an entry to the IP interface of a UNIX Managed Node • Edit an entry in the IP interface of a PC or a UNIX Managed Node • Toggle the icon of a Managed Node	• Display a PC/UNIX Managed Node properties dialog • Open a UNIX Managed Node or X terminal	Install/remove a TME client that is a PC/UNIX Managed Node		

The following table outlines the roles in TMR connections.

Super	Senior	Admin	User	Install - Client	Install - Product	Backdrop - Restore
• Disconnect TMRs • Make a remote one-/two way connection between TMRs • Make a secure one-/two way connection between TMRs	• Exchange resource information between TMRs • Schedule exchange	Drag and drop top level Policy Subregions from remote TMRs onto the local desktop	View and update the connection status			

The following table outlines the roles in Policy Subregions.

Super	Senior	Admin	User	Install - Client	Install - Product	Backdrop - Restore
• Move a profile manager from one Policy Subregion window to another • Move a profile from one profile manager window to another • Clone a profile in profile manager (also copy, create, delete) • Create/delete a subregion • Add/remove a managed resource type from/to a Policy Subregion • Delete a profile manager from a Policy Subregion • Edit a profile manager • Set policy on the records of a profile in a profile manager • Synchronize a profile in a profile endpoint		• Distribute one or more profiles from a profile manager • Subscribe a Managed Node or another profile manager • Remove a subscription	Open a Policy Subregion window			

The following table outlines the roles in the task library.

Super	Senior	Admin	User	Install - Client	Install - Product	Backdrop - Restore
	• Change the policy for a task library in a Policy Subregion • Create/delete a task library from a Policy Subregion • Set the policy for a task library in a region • Move a task library from a Policy Subregion window	• Create/delete/edit a job or task in a task library • Schedule a job for future execution				

The following table outlines the roles in notification.

Super	Senior	Admin	User	Install - Client	Install - Product	Backdrop - Restore
	• Expire notices in a notice group • Set notice expiration length for a notice group		• Combine multiple related notices in a notice group into a single listing • Display filter notices in a group • Forward notices via E-mail • Mark notices as read/unread • Sort/read/save notices			

The following table outlines the roles in scheduling.

Super	Senior	Admin	User	Install - Client	Install - Product	Backdrop - Restore
	Start and stop scheduler	• Disable, enable, edit, remove a scheduled job • Schedule a job	• Browse list of scheduled jobs • Change the information displayed in scheduled job list • Find a job in the list • Route jobs			

The following table outlines the roles in Desktop GUI.

Super	Senior	Admin	User	Install - Client	Install - Product	Backdrop - Restore
• Check integrity of the TMR database • Enable/disable diagnostic tracing • Execute maintenance actions (odadmin, odstat) • Place the TMR in single user node for maintenance			• Browse the list of installed products and applications • Send E-mail to the support provider			Backup or restore TMR database

5.2.2 Viewing Administrators

On the TME 3.0 desktop is an icon called Administrators which is a collection for the administrators in the local and the connected TMRs. To open the Administrator icon you must have the role of user assigned to your administrator access. Administrators can be viewed from the desktop only.

Figure 156. Administrators Icon on TME Desktop

When you double-click on the icon on the desktop, the Administrators window opens. Each icon in the window represents a Tivoli administrator. To work with the administrators you must have the super or senior role defined for your administrator.

Figure 157. Administrator Window

Each administrator icon has a context pop-up menu. You can open it by clicking with the right mouse button on the icon. The pop-up menu allows you to perform the following actions on the administrator ID:

Edit properties allows you to change the name, the user login name and the group name associated with the administrator. You can open an administrator by double-clicking on the administrator′s icon. The desktop, depending on the roles that are subscribed to the administrator, opens. It doesn′t matters if you are opening an administrator icon from an administrator that is not in your local TMR. This action can still run across connected TMRs.

Edit the TMR Roles allows you to change the roles of an administrator in local and in any connected TMR.

Edit Resource Roles allows you to change the roles of an administrator for resources in local TMRs.

Edit logins allows you to perform changes to login names which map to an administrator.

Edit Notice Group Subscriptions allows you to perform change actions to the notice group subscriptions of the administrator.

5.2.2.1 Creating an Administrator

You can perform this action from the desktop or from a command line interface. You need to have the senior role to perform it. There are two ways to start this action at the desktop:

- Click on the **Administrator** icon on the desktop and open the context menu. Then choose the option **Create administrator**.
- Double-click on the **Administrator** icon and the Administrator window opens. On the menu bar choose **Create**.

The window to create a new administrator appears regardless of the way you choose (see two ways above) to create it.

Figure 158. Create Administrator Window

There are three text fields in the window you must fill in:

- Administrator Name, which will be the name of the icon
- User Login Name
- Group Name

Note: Take care with your user accounts under Windows NT. Before new administrators can log on they must first be defined under Windows NT (User Manager) with the correct user rights. Then it is possible to log on locally or in a connected TMR. Remember the administrator under Tivoli doesn't depend on a machine or a system. When you finish the updates to the text fields choose the **Set TMR roles** button. This will lead you to a new window.

5.2.2.2 TMR Roles

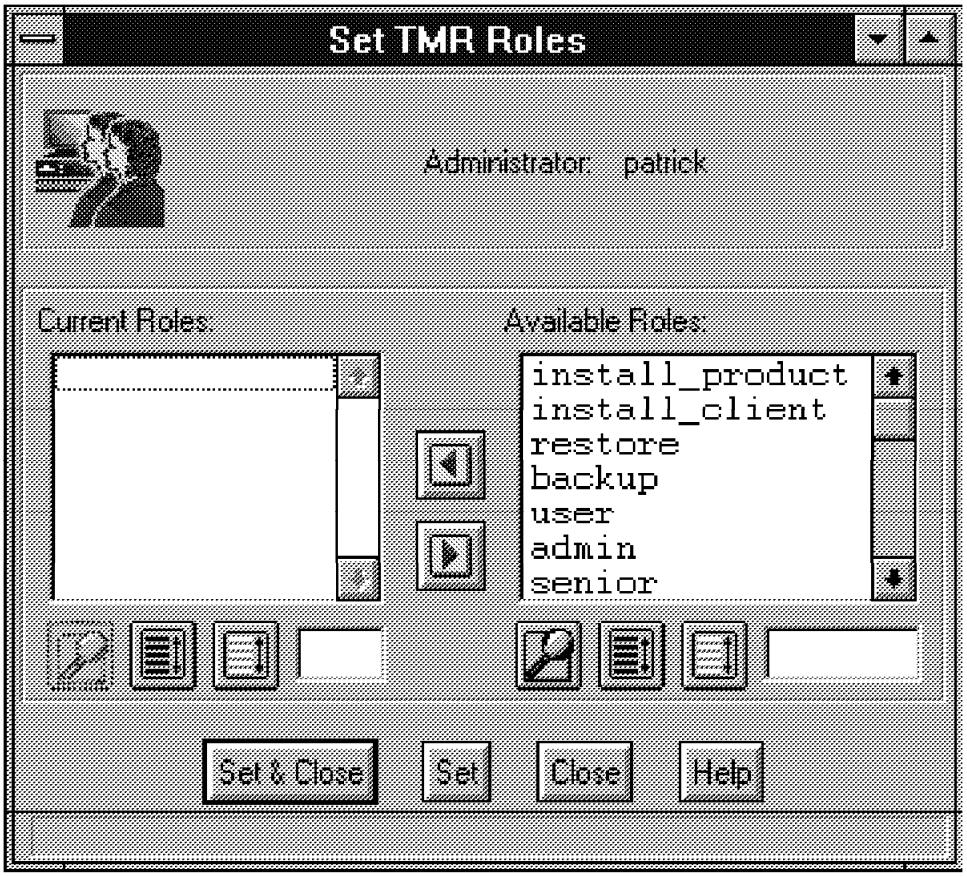


Figure 159. Set TMR Roles I

The TMR roles window is partitioned into two list boxes:

- The right one is for the available roles from which you can choose.
- The left one is for the current roles, added to the administrator.

To add a role to the administrator choose one in the right list box and add it with the bolt between the boxes to the left box. Think carefully about the roles, because often an administrator needs only resource roles and no TMR roles. After you finish your input, confirm it by selecting the **Set & Close** button.

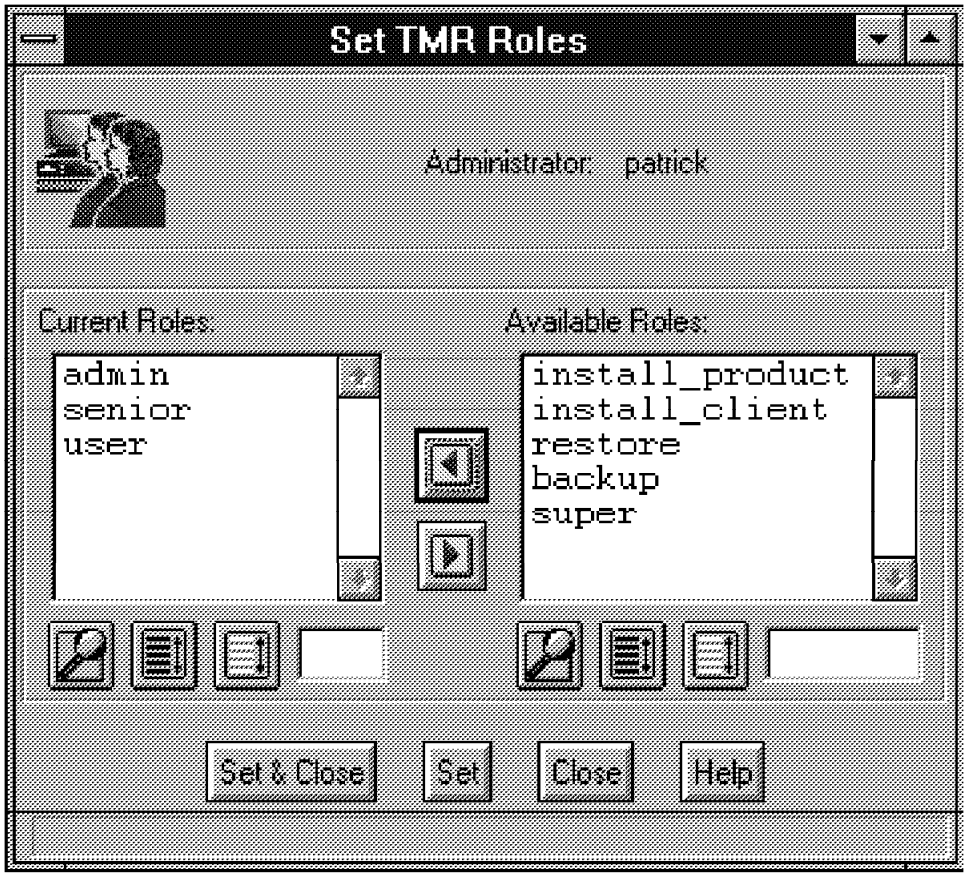


Figure 160. Set TMR Roles II

Command Line Interface: If you use the command line instead of the desktop, you have to use the wsetadmin command; it changes the properties of an existing administrator. For this action you need the senior or super role.

The following arguments are available:

- -L removes the logins for an administrator.
- -l adds the specified login.
- -N removes a subscription.
- -n adds a subscription.
- -R group removes all roles from an administrator.
- -r group,role1:role2.. changes the existing administrator roles in the specified roles.
- name - administrator name.

The command syntax looks like this:

```
wsetadmin -L/-l login -n/-N noticegroup -R group -r group,role1:role2..
name
```

To understand the command handling we show you the following example:

1. In this example the administrator tester1 will change its roles to super, senior and admin. It won't have access to the TME administration notice

group in the ntserv1-region in the future. A new login name called tester_1 will be added.

The command looks like this: `wsetadmin -r @ntserv1-region,super:senior:admin -N "TME Administration#ntserv1-region" -l tester1 tester1`

Don't forget that you will need to define the new user ID in the NT User Manager before you can add it the TMR.

The roles in the example above refer to the resource roles, not the TMR roles.

Logins: Set Logins is another push button in the Create Administrator window. After finishing the TMR roles you will automatically go back to the Create window.

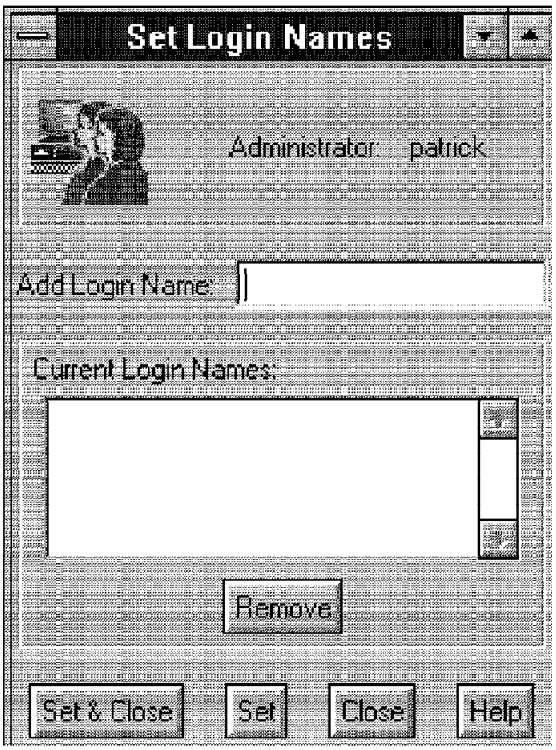


Figure 161. Set Login Window

The login window has one input field for the login name of the administrator. Fill in the correct name in the field and press the Return button on your keyboard to confirm the name. The name now appears in the list box (Current Login name). You can define more than one login name for each administrator. After finishing the input, select the **Set & Close** button. It leads you back to the Create window. To add users under NT (required for Tivoli) and then add administrators under Tivoli it is much easier to use a batch file or a REXX command file. Here is a short BAT file with the commands:

```
net user ntdom2 cm4274r /ADD /comment:"Test User" /expires:never times:all /DOMAIN

net group "Domain Admins" ntdom2 /ADD /DOMAIN

net localgroup "Tivoli_Admin_Privileges" ntdom2 /ADD /DOMAIN

wcrtadmin -l ntdom2 -r /Administrators,admin:senior:user -n "TME Authorization#ntserv2-region" -u ntdom2 ntdom2
```

Figure 162. Automatically Adding Users

Set Resource Roles: The third button in the Create window is the Set Resource button. Selecting it opens a window which has three parts:

- A resource list in the upper part of the window
- A list box with available resources on the right
- A list box with current resources on the left

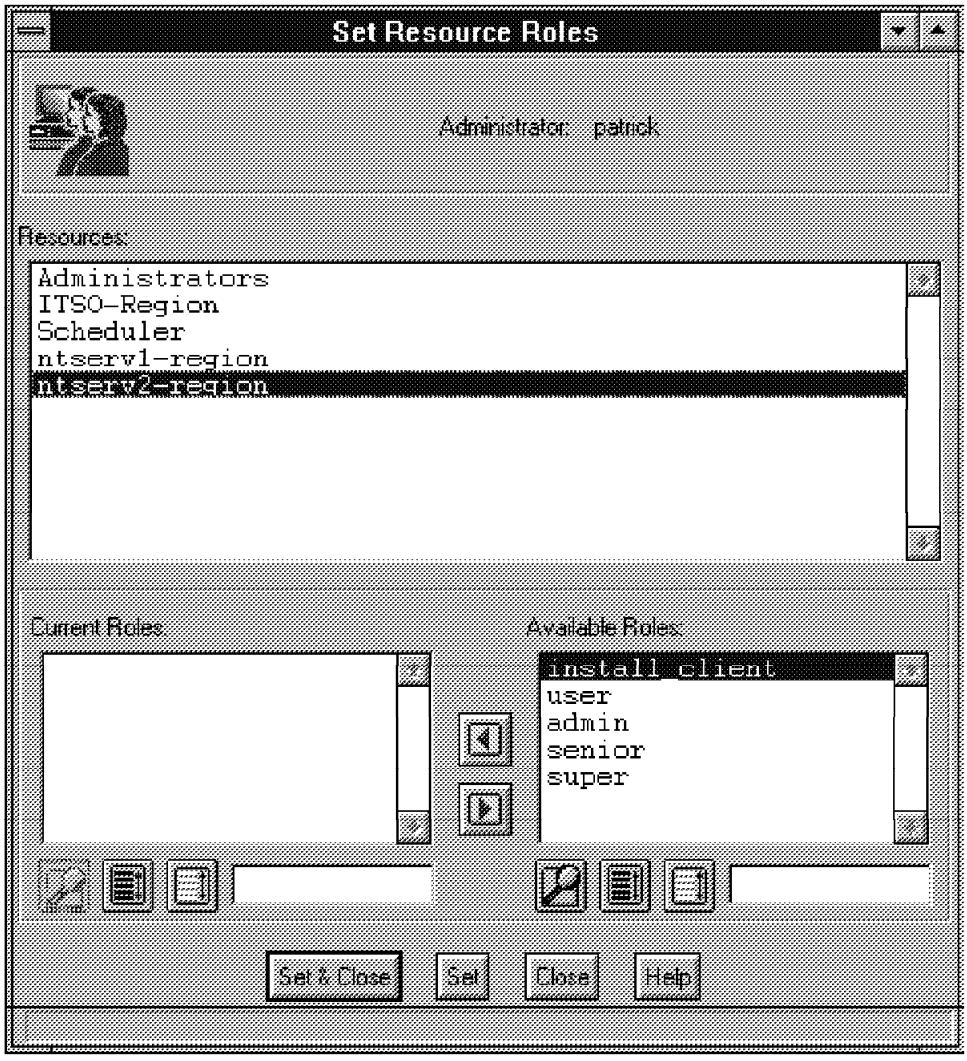


Figure 163. Set Resource Roles

First you have to choose a resource for which you want to set the administrator role. Then you can choose one or more of the available roles to add to the current roles. If you want to do this for more than one resource, you must select

the **Set** button only. When you finish, select the **Set & Close** button and come back to the Create window.

Notice Groups: The last button in the Create window is for the Notice groups. In this window you can choose the notice groups you want to notify the administrator. The notice groups are very useful because every administrator assigned to one or more of them can receive information about system management operations performed by other administrators under the TME platform. You can subscribe the administrator to a group from the desktop or from a command line.

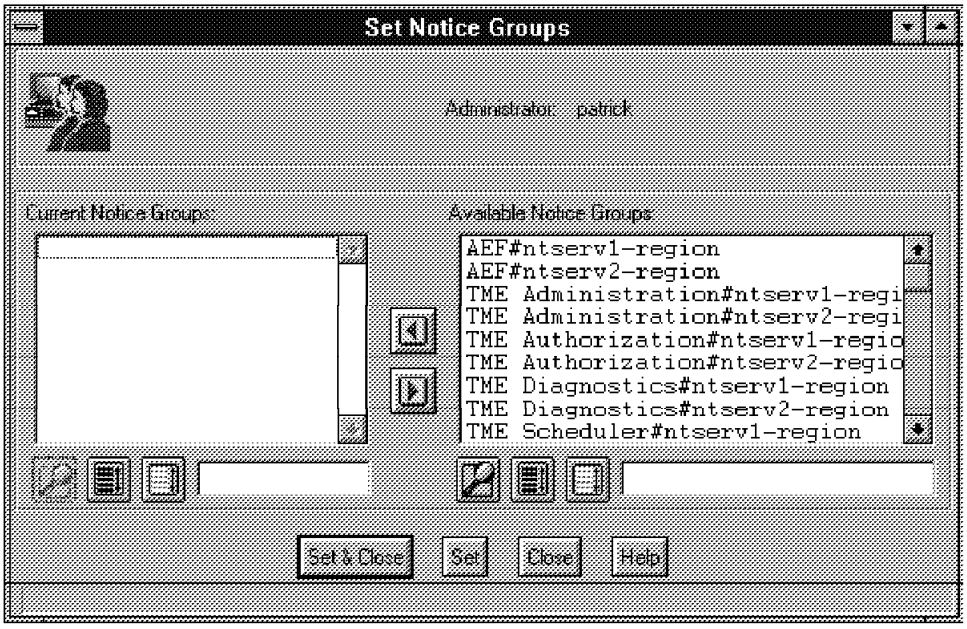


Figure 164. Set Notice Group

After you have made all of the updates to all of the fields close the Create window by selecting the **Create and Close** button. At the desktop in the Operation status box a message appears:

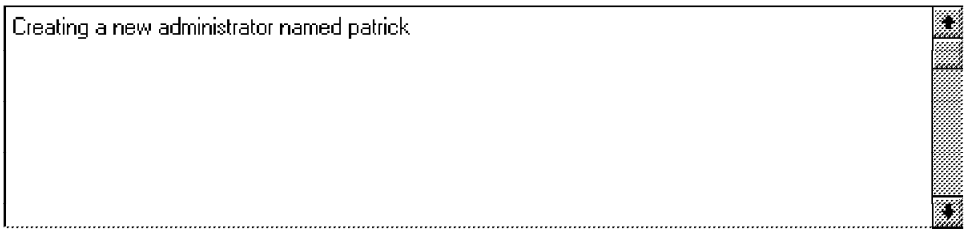


Figure 165. Message Window

Command Line Interface: The command to create an administrator from the command line is wcrtadmin. It has several options:

- -l,login - Sets up the login for the new administrator.
- -n, noticegroup - Sets up a notice group subscription for the new administrator.
- -r, group, role1, role2 - Defines the administrator roles in the specified group. To specify TMR roles, use the string *global* instead of the *group* string.

- name - Labels the new administrator.
- -u, username - Sets up the user ID for the administrator.
- -g, groupname - Sets up the groupname for the administrator.

The following two examples show how to use the command:

1. \tivoli\bin\w32-ix86\bin\wcrtadmin -l tester -r /Administrators:admin:senior:user -n "TME" Administration#ntserver1-region" -u tester1 tester1
2. \tivoli\bin\w32-ix86\bin\wcrtadmin -l tester2 -r global.admin:senior:user -n "TME Administration#ntserv1-region" -u tester2 tester2

The command is very tricky to handle because it is very sensitive. Here are some tips:

- Before you can create an administrator for Tivoli, be careful that you have defined one under the NT environment (in the User Manager). The command won't work if there is no administrator defined with the same name under Tivoli.
- If you want to define an administrator with more than one notification group, you have to enter the argument -n twice. For example, if the administrator Jan has three notice groups to subscribe to, the command will look like this:

```
wcrtadmin -l jan -r /Administrators,admin:senior -n "TME Authorization" -n "TME Administration"
```

Figure 166. Multiple Notification Groups

- Be aware of the quotation marks. You must always use them for the notice groups.
- Look for the spaces between the arguments. You have to put one between each input string.
- Take a look at the TMR you want the user defined to. Because of the Tivoli framework it is possible to define an administrator for a TMR that is connected to your local TMR. In this case you must specify the TMR name for the notice group. For example: `-n "TME authorization#ntserv1-region"`.
- You always have to place a colon between the roles you assign a new administrator. For example: `wcrtadmin -l peter -r global,admin:senior:`.
- The group you defined your administrator to must always begin with a slash (*/*). The only exception is the group *global*.
- The word global in terms of Tivoli roles stands for the TMR roles, any other word (/Administrator) stands for the resource roles.

The wcrtadmin command is found in the directory \tivoli\bin\w32-ix86\bin.

5.2.3 Adding Resources to an Administration Desktop

After creating Tivoli administrators with a set of roles, you need to allow them to access those resources from their desktop. Under Tivoli this can happen with drag-and-drop actions. Any icons (for example, Policy Regions) can be copied. Often it is easier to create a new collection and drag and drop the icons inside this new collection. For these actions the user role is required.

To open the new administrator desktop you can double-click on the **Administrator** icon in the Administrator window or you can start a second Tivoli TME platform, or restart, and log on as the new administrator. Using this object-oriented drag-and-drop technology, you can define administrators enterprise-wide from a single desktop.

5.2.3.1 Collections

A *collection* in Tivoli is a container that can be created on request and placed on the desktop. To populate collections you have to use drag-and-drop actions. The contents of a collection depends on the purpose it is defined for.

To define a new collection you have to choose the **Create** option in the menu bar on the Tivoli desktop and select **Collection**. The window for defining a new collection opens.

Figure 167. Collection Definition Window

After you fill out the Collection Name the new collection appears on the desktop. In Figure 168 it is the middle icon.

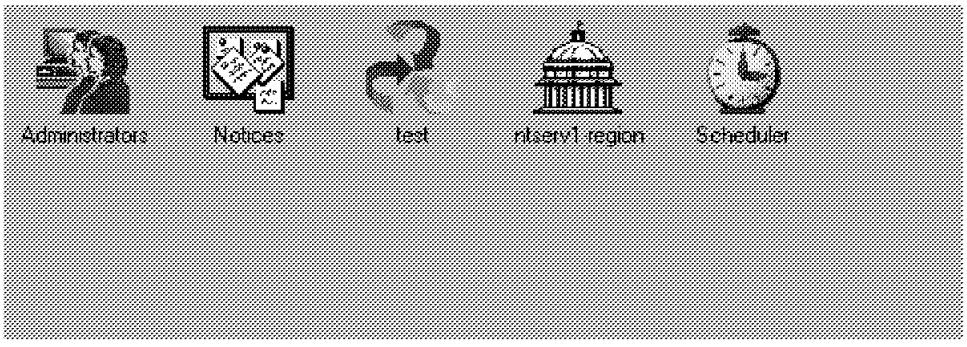


Figure 168. New Collection at the Desktop

Either use the right mouse button and select **Open** or double-click on it to open the collection main window. By default, it is empty when first created. To populate the collection use drag-and-drop actions for resources from other Policy Regions.

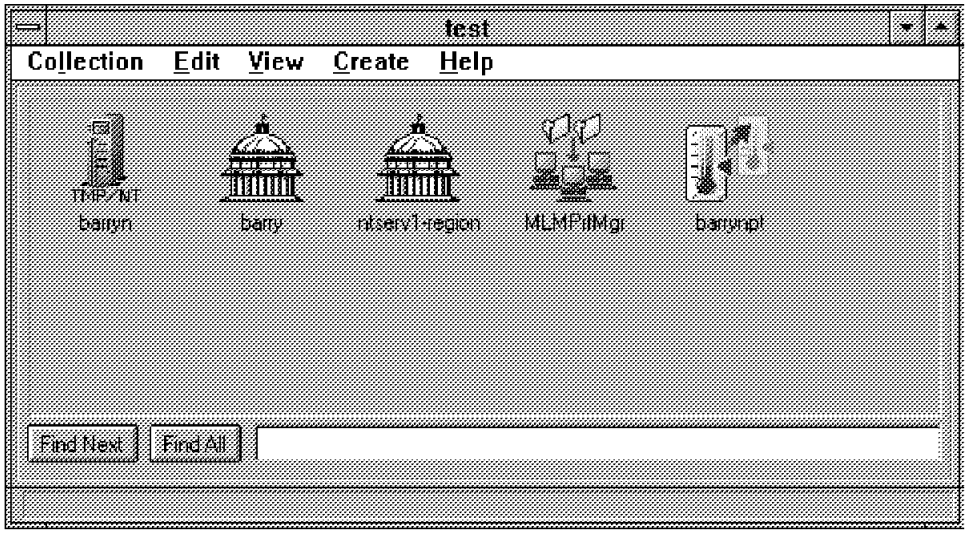


Figure 169. Collection Main Window

5.2.4 Miscellaneous

The following are some other important commands that we worked with:

- wgetadmin
- odadmin

The wgetadmin command lists all information about a specific administrator. The command lists the name, logins, roles and notice groups. The administrator name is case sensitive. If the name of an administrator is not specified, the command lists the information for the current administrator. The syntax looks like this:

```
wgetadmin -n -o administratorname
```

The -n argument lists only the name of the administrator, and the -o lists only the object ID of the administrator.

The odadmin command has a lot of power. Without an argument the odadmin command shows the region-name which is very important for connecting TMRs. An example of this is shown below.

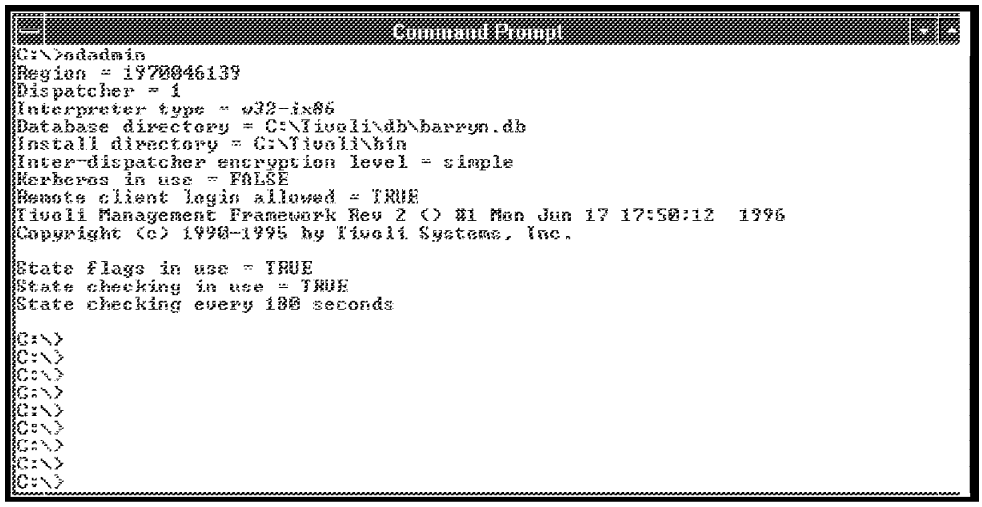


Figure 170. Output of odadmin Command

The following is an explanation of each line:

Line 1 TMR number (unique, comes with the license)

Line 2 Server or client number (If dispatcher=1, then it is the server.)

Line 3 Machine interpreter type

Line 4/5 Identifies the path of the local TME database and the TME binaries

Line 6 Shows the encryption type if it is used

Line 7 Indicates whether Kerberos is used

Line 8 Identifies the library path for TME operations

Line 9/10 Identifies the version of Tivoli

Line 11-13 Information about the TME client pings

5.2.5 Display Active Servers

To display the active servers you can still use the odadmin command:

```
odadmin odlist
```

This command shows the following output:

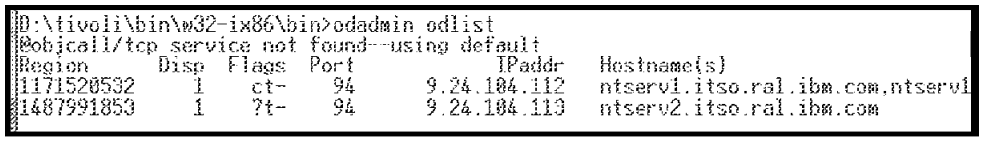


Figure 171. odadmin Command

For each server the output displays a TME number, a set of flags reflecting the connection status, the port of the connection and one or more host names. If the first flag is a c the connection is active. A ? indicates that the connection status is unknown. A - says that the connection is down.

The second flag indicates the status of trust; t means trusted, - means untrusted.

This command doesn't display the active clients.

5.2.6 Client Operations

You can shut down a TME client with the odadmin command. To do that you have to execute the following:

```
odadmin shutdown (num)
```

where num is the client number.

If you use the option clients instead of num you can shut down all clients in the local TMR. It only works for machines that have a dispatcher number. If you use the option all you can shut down all of the clients and servers.

To restart the machines you use the odadmin command again, but you use the options: start and num, clients or all. You can also use the odadmin reexec command with the same options as the start command.

5.2.7 Maintenance Mode

The maintenance mode halts all management activities in the local TMR. You can use the odadmin command to stop all clients and then perform stand-alone actions. An example of this would be when you need to back up the database. When you place a TMR in Maintenance Mode you become the only administrator in the TMR that is authorized to do anything. The desktops of all other administrators are locked.

To bring the system to Maintenance Mode you must choose the option **Maintenance** from the Desktop menu on the TME desktop. The following window appears.

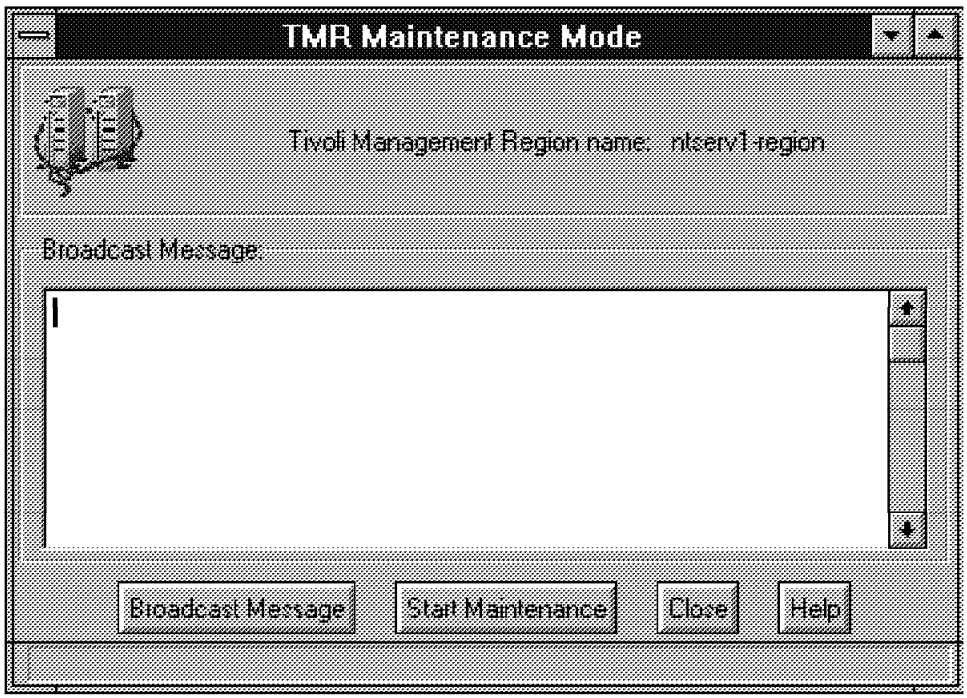


Figure 172. Maintenance Window

You can send a message to all machines in the local TMR before you go into Maintenance Mode. You do that by filling in some text in the message box and by selecting the **Broadcast Message** button. To start the Maintenance Mode you must choose the **Start Maintenance** button. A message window appears similar to the following one.

Figure 173. Maintenance Window II

This window confirms that you are in Maintenance Mode. To exit Maintenance Mode you have to choose the **Exit Maintenance** button.

If another administrator tries to perform actions in your region while it is in Maintenance Mode the message shown in Figure 174 will appear.

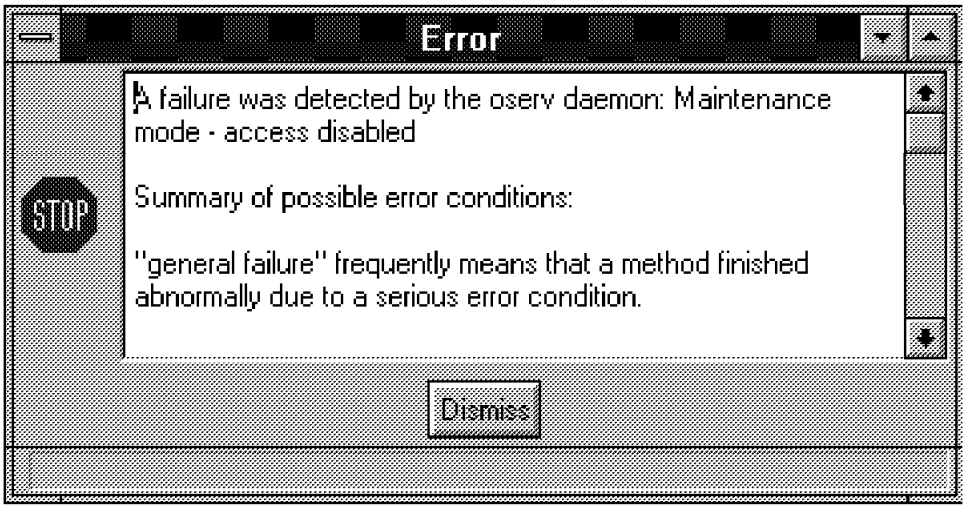


Figure 174. Failure Window

You can also perform this action from the command line by using the command wbcktmr. You need the role super to perform this action.

You have the option -p which sets the TMR into Maintenance Mode. To end the mode you must press the Ctrl+C keys.

5.2.8 Working with the Database

When you restore a TMR database you must first run the command in the local TMR and then in the connected TMR. The most important options for the command are:

- -u updates the database and fixes any discrepancies.
- -x verifies and repairs object references across TMR boundaries.

5.2.8.1 Backup the Database

Like all sensitive resources a TMR database should be backed up at regular intervals. It is possible to choose to back up only one client, a server or the whole TMR. These actions can be performed from the desktop or from the command line. Additionally, it is possible to schedule a backup action.

When a backup is started, TME creates a temporary backup file while it copies the database. From the desktop this file will be written to the database directory; from the command line the file will be written to the current directory.

To perform the backup you have to use the wbkupdb command from the command line or the Backup option from the Desktop menu on the TME desktop. When the temporary backup file is completed the TME moves the archived file to the backup directory. The name of the backup file is DB_date-time. The backup directory is stored in the directory where the database resides.

To maximize the security level for your environment, you should set up a planned schedule for when your backups will occur. For example, you may decide to back up all TME servers daily and all clients weekly. In addition, you should back up your TMR database daily.

Since Tivoli doesn't delete old backup files you have to look for older backups and delete them. Otherwise, you may eventually run into disk space problems on that drive.

To back up the database from the desktop you have to select the **Desktop** menu and then the **Backup** option. The following window appears.

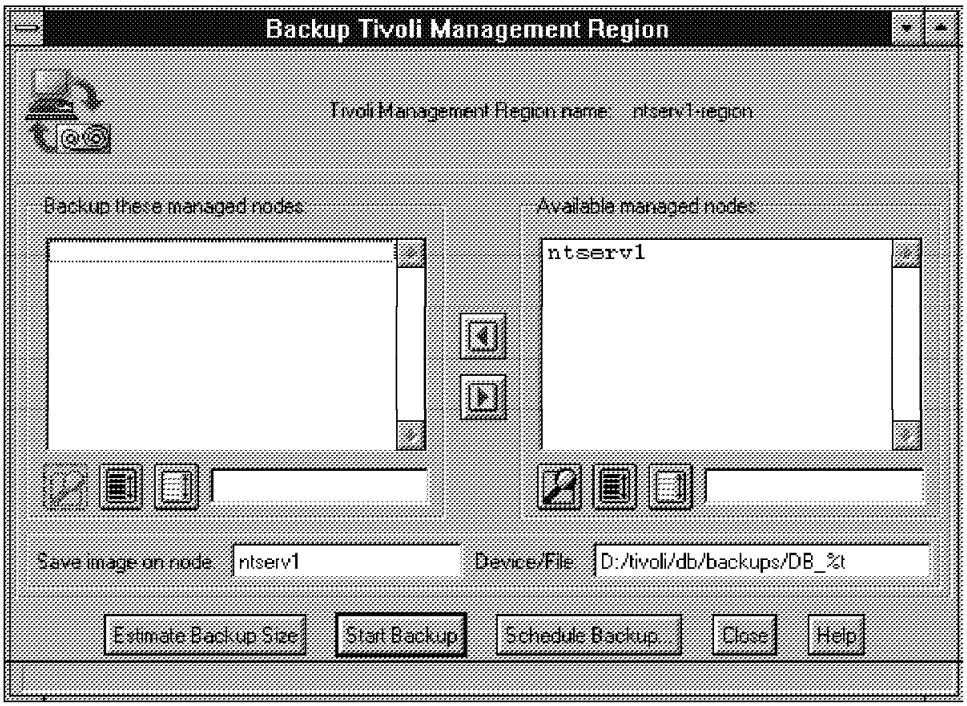


Figure 175. Backup Main Window

Figure 175 shows the available nodes for backup in the right list box. At the bottom on the right-hand side you can see the path for the backup. If you want to save it somewhere else, you need to change the field there.

To back up a node you have to choose one from the right list box. As with other functions, you move it from the right side to the left side to select it.

Before you make a backup, it is useful to know the backup size. For this purpose you can use the Estimate Backup Size push button at the bottom of the window. Choosing it opens a new window.

Figure 176. Backup Main Window

This window shows the disk space estimated for a backup of the TMR database.

If you decide that there is enough space, you can choose either the **Start Backup** button to start it right away, or you can choose the **Schedule Backup** button and Figure 177 on page 151 opens.

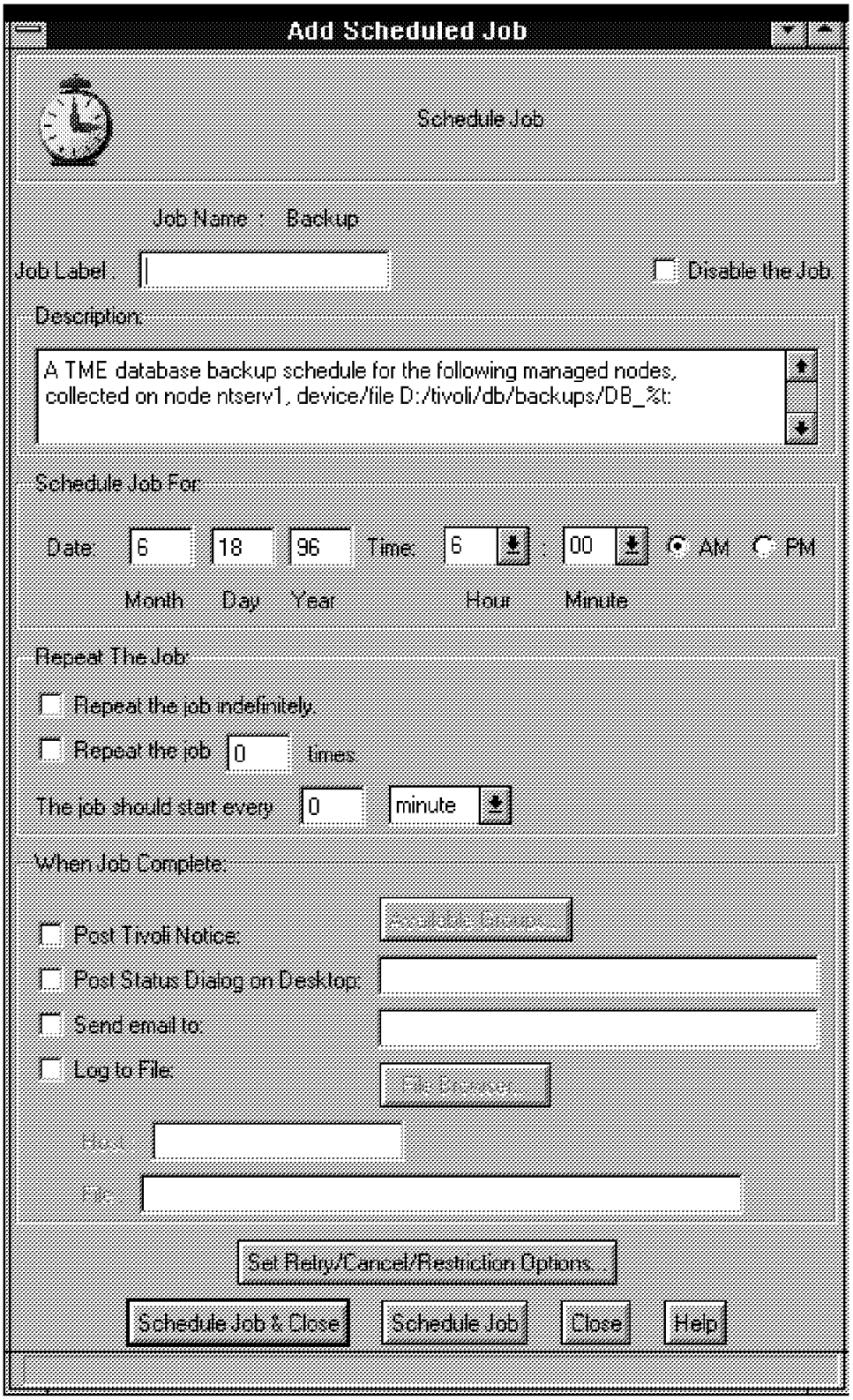


Figure 177. Backup Main Window

In this window you can define:

- The name of the schedule job.
- The date and time for backup.

- How often the job should run.
- What to do after completing the job. An example might be to send E-mail to a user ID for notification purposes.

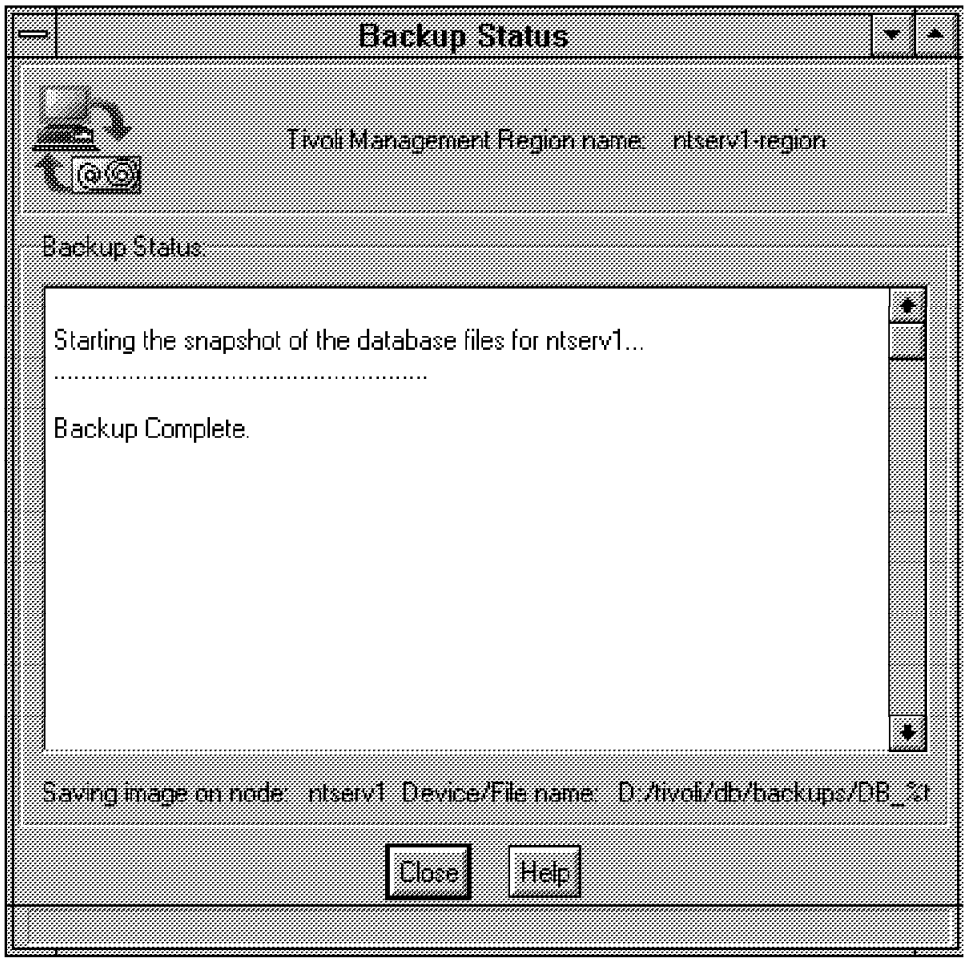


Figure 178. Backup Main Window

This window shows the progression and the status of the backup of the database.

You can also start the backup in a connected TMR if you have the role backup assigned to you. To do this you have to go to the Administrator desktop of the remote region and start the backup.

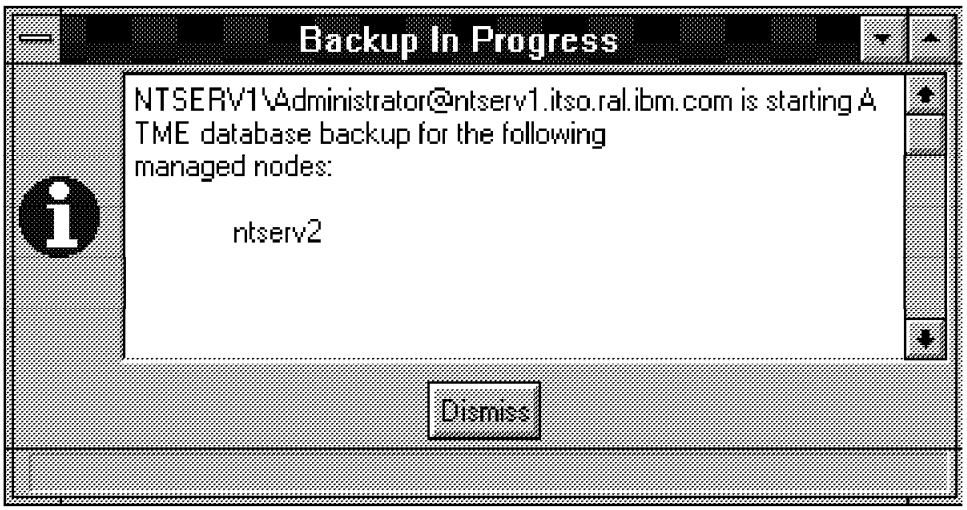


Figure 179. Backup Main Window

Figure 179 appears at the server of the connected TMR.

From the command line you have to type the `wbkupdb` command. This command has several options:

- -e nodename - Estimates the size of the backup of a node.
- -d device - Specifies the file or the device of the backup.
- -f - Overwrites previous backup files with the same name.

An example of the command would be: `wbkupdb -d D:\tivoli\db\test -f .`

5.2.9 Restoring

To restore a database you have to type the `wbkupdb` command again, but this time with different options:

- -r nodename - Restores the database for a special node.
- -d device - Specifies the device of the backup file to restore.

An example of this would be: `wbkupdb -r -d D:\tivoli\db\test ntserv1`.

After a restore the Tivoli TME desktop shuts itself down. You have to restart it again.

Note

You can only do a backup for the database and only for the TMR server that includes the database. Other backups must be made with a different tool, such as ADSM. And you can only restore from the command line. There is no desktop-based application available now.

5.2.10 Communication

TME provides a distributed environment. Each machine has a daemon called oserv, or object dispatcher, that communicates with other Tivoli daemons. The requirements for communications between Tivoli clients is a running TCP/IP environment. You can use the odadmin command to view the behavior of the clients. The command must be used with the following syntax:

```
odadmin odinfo 1 (clientnumber)
```

If you need to, you can enable or disable the use of the Tivoli daemon ping. Therefore, you have to use the Tivoli odadmin command again. The daemon ping is something integrated in Tivoli. The daemon on the TME server occasionally pings connections to other Tivoli daemons to determine if the client is still connected. The server pings only under the following special conditions:

- If the client daemon is located in the local TMR.
- If the client daemon has not sent a request to the TME server during the timeout period.
- If another client daemon has sent a request for the client daemon during the timeout period.

You can also use the wping command interactively, or as part of a command script.

5.2.10.1 Enable/Disable

To enable/disable the use of the Tivoli ping cache use the odadmin command in the following way:

```
odadmin set_keep_alive on/off
```

You can see the difference between the two states if you use the command odadmin odinfo. If the status is on, you can see lines 11-13 of the output. If the status is off, you cannot see the last three lines. There is only one line that says `State flag in use=FALSE`. By default, the use of the Tivoli daemon ping cache is enabled.

To enable or disable the internal ping mechanism of the Tivoli daemon you must use the odadmin command with the option set_keep_alive off or on. If the ping is on, the clients are periodically polled.

To set the polling interval you have to use the odadmin command with the `time` option. The option `time` sets the interval for the ping time.

5.2.11 Working with IP Addresses

IP addresses and the protocol TCP/IP are the basics for Tivoli to work with clients. It is possible that you might have to change the addresses of a client if they are mobile or if they move. You can change IP addresses from the Tivoli desktop or from the command line. To change TMR IP addresses you need the super role. To change client addresses you need the admin role.

5.2.11.1 Desktop

If you change the IP address from the desktop you have to go to the Policy Region icon of your local or connected TMR. Double-click on the icon and the Policy Region window appears. Here you can change the IP address of a node.

Figure 180. Policy Region Window

Using either the pull-down menus or the right mouse button, you can click on one of the resources and then edit its properties. If you choose to do this, you will end up with a window like the one shown in Figure 181.

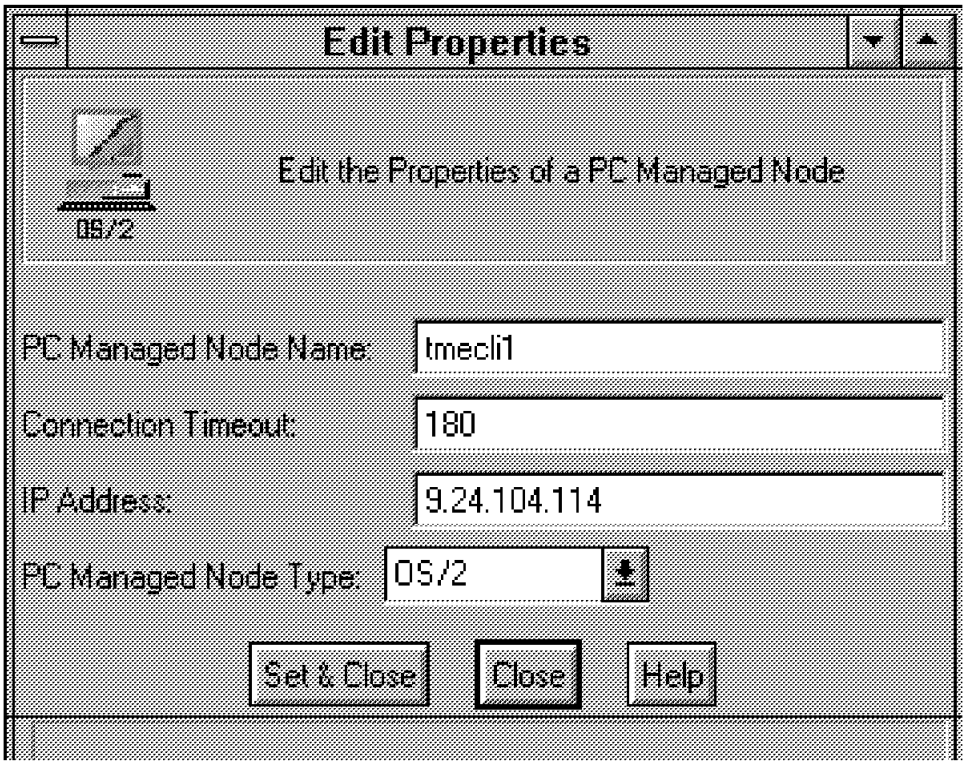


Figure 181. Editable Properties

If you look at the properties of a server instead of a PC Managed Node you will see slightly different fields. The following figure shows what a server (and in this case also a Managed Node) looks like.

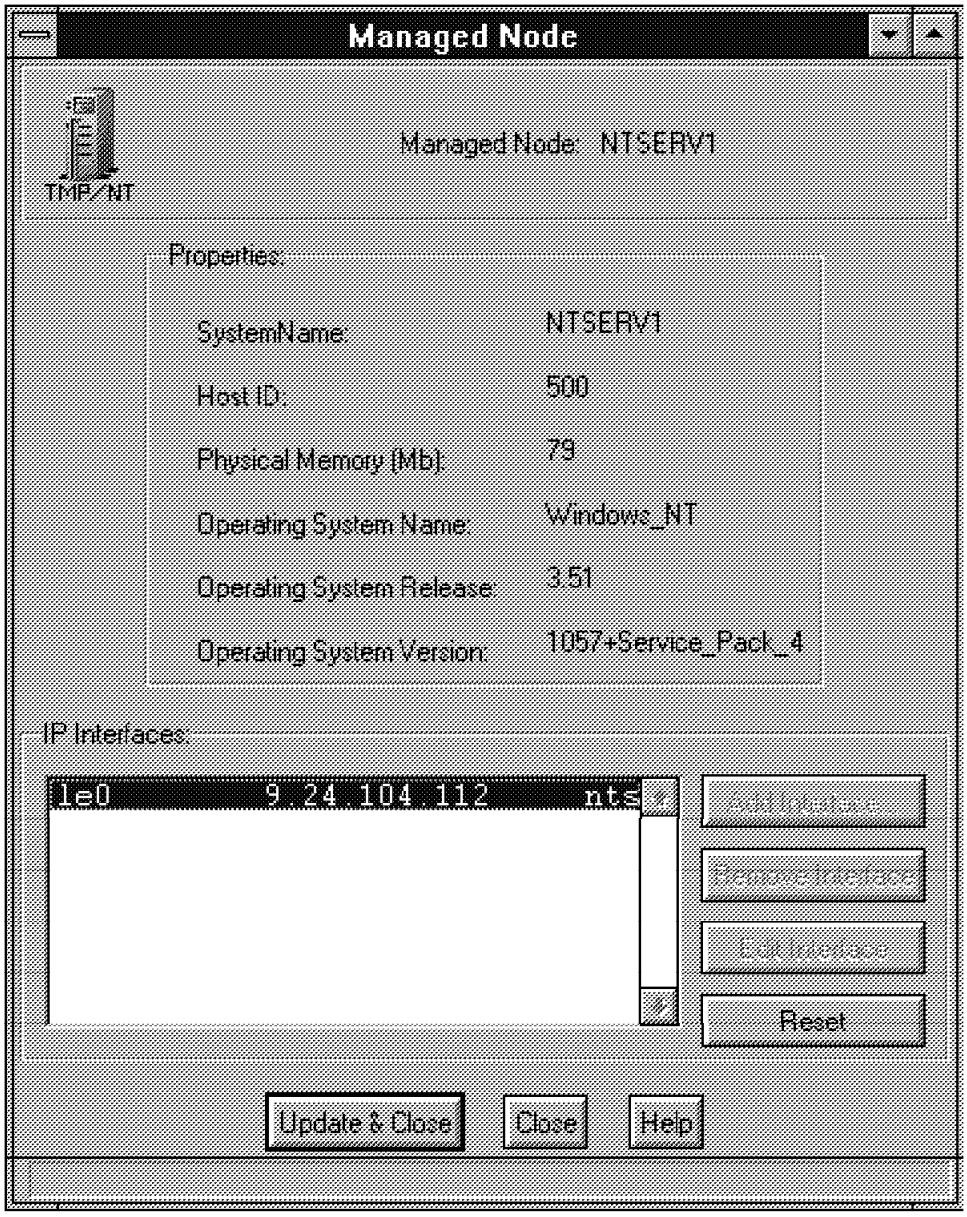


Figure 182. Properties Server

5.2.11.2 Command Line

If you want to change the address of your server, it is often much easier to shut down all clients before you make the change. Therefore, you have to use the odadmin command:

```
odadmin shutdown clients
```

To change the IP address you have to use the odadmin command again:

```
odadmin odlist change_ip 1 9.111.104.114 FALSE
```

.

This command changes the IP address of the server (Dispatcher 1) to the new IP address. The FALSE option specifies that a gethostbyaddr should not be

performed since the IP address has not changed. After this action you can restart the clients and the server again.

```
odadmin start 1(server)/clients
```

It is possible to add or delete an alias with the odadmin command:

```
odadmin odlist add_ip_alias/delete_ip_alias objectdisp. ip-adrr
```

The command can be performed in a similar way to add or delete a hostname alias.

5.2.12 Removing Objects

Another part of the administration is adding and removing objects from the desktop.

The easiest way to create or add a new object to the desktop is with the Create option from the menu on the TME desktop. You can define several objects depending upon the window you are currently in. For example, from the desktop you can create Policy Regions and collections. From the Policy Region window you can create subregions, task libraries Profile Manager, Managed Nodes and PC Managed Nodes. We show two examples, one with a Policy Region and one with a PC Managed Node.

On the desktop you choose the **Create** option from the menu and the **Region** option to create a new region.

Figure 183. Create a Policy Region

You can only fill in the name of the new Policy Region.

On the Policy Region window you also have to choose the **Create** option from the menu to add an object to your local desktop. Here you can choose the option **PC Managed Node** to create a new PC node. The following window appears.

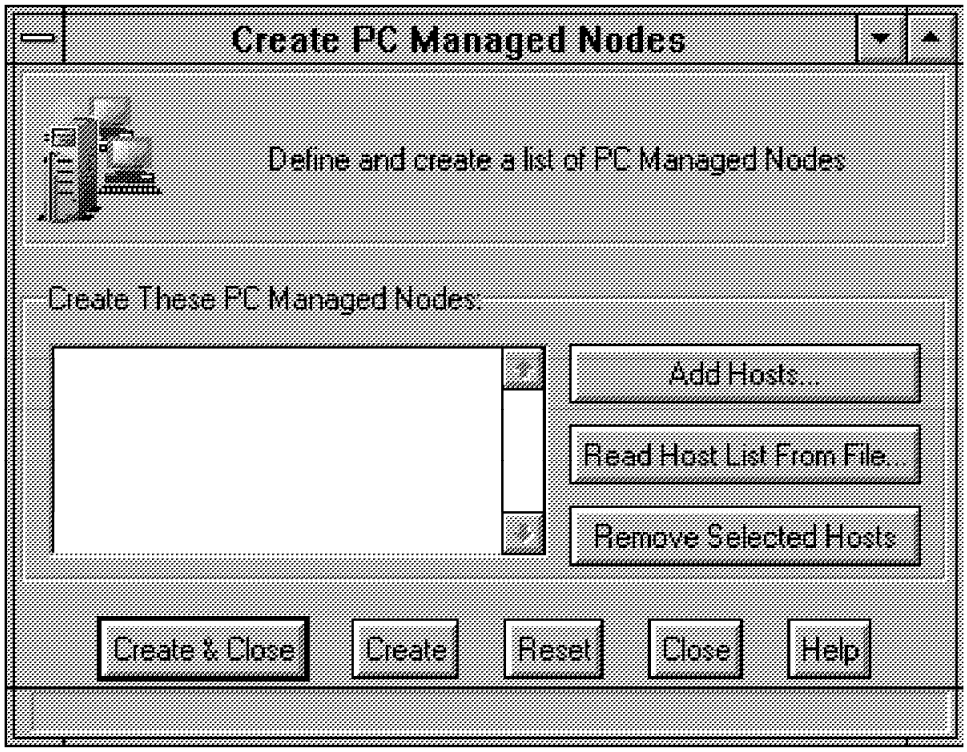


Figure 184. Create PC Managed Node

With the Add Hosts button you can add a new host. A window for defining the new host would look like the following:

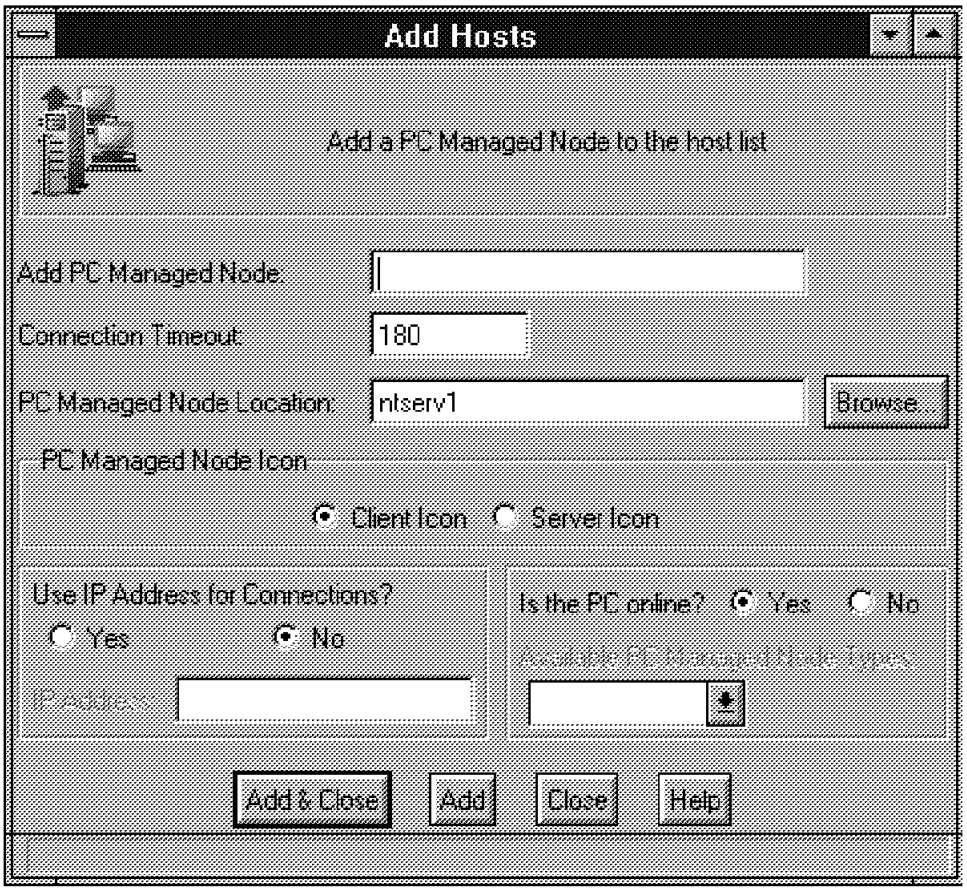


Figure 185. Create PC Managed Node

In this window you can fill in information about:

- The name of the node
- A new connection timeout period (ping interval)
- The location of the PC Managed Node
- The type of the icon representing the node
- The IP address (if needed)
- The status of the PC (on/offline)

After you set the values you can close the window by selecting the **Add & Close** button. In the first window the just defined node will appear in the list box. You can now select this node and choose the **Create&Close** button to add the node to the desktop.

Another way to add nodes to a local desktop is to drag and drop them from another connected TMR Policy Region to the local TMR.

5.2.12.1 Removing/Deleting an Object

You can remove an object be selecting the **Edit** option from the menu bar. There are two options:

- Remove
- Delete

Remove deletes the object only from the local desktop and not for the whole environment. The object is still in the database.

Delete deletes the object from the desktop and the database.

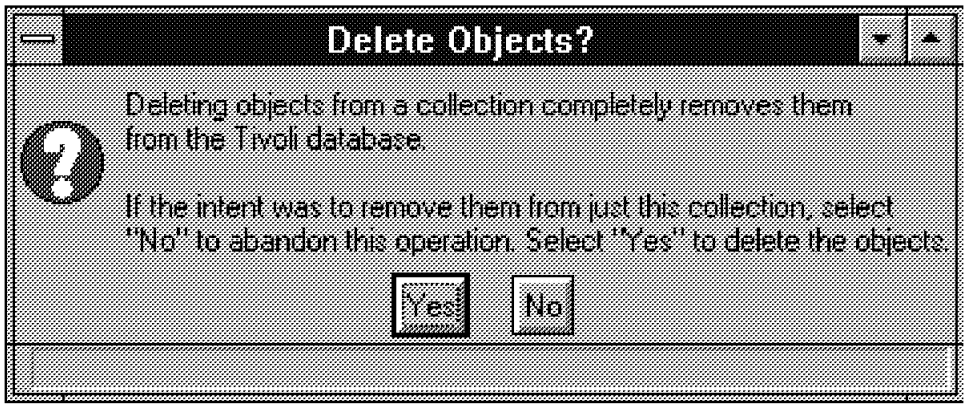


Figure 186. Create PC Managed Node

In some windows, such as the Policy Region window, you can only delete objects. In the Administrator window you can still remove and delete.

To remove a node from the command line you have to use the wrmnode command.

Another useful command is the odstat command. It shows the currently running operations and the 200 previous operations. A piece of the output from the command follows.

```
daemons:
  pid  flags  count     uid     gid         oid         name
  348     MO      0                  1970046139.1.157  \w32-ix86\TAS\SCHEDULER\TMF_sched
  352     MO      0     root         1970046139.1.509  \w32-ix86\TME\SENTRY\sentry_engine
  132     MO      0                  1970046139.1.178  \w32-ix86\TAS\Administrator\Admin_prog1
    1     MO      0 Administ         1970046139.1.604  \UI\Extd_Desktop
  464      M      0     root                           \w32-ix86\TMF\NOTIF\NtfServer
  180     MO      0                  1970046139.1.353  \w32-ix86\TAS\InterRegion\InterRegion_ui_prog1
  700     MO      0                  1970046139.1.168  \w32-ix86\TAS\Administrator\AdminCol_prog1
  364     MO      0                  1970046139.1.713  \w32-ix86\TAS\TGC\GenCol

n_active = 13  n_free = 187
   tid type  State   Err    Method
     1 SYS
     2 O+bhdoq   run        1970046139.1.157#TMF_Scheduler::scheduler# start
        service: "201 attr 'BDBPG:RestrictionsDatabase:0' dat:"  err=0  pid=348
     3 O+bhdoq   run        1970046139.1.509#SentryEngine::engine# run_engine
        service: "202 attr 'sentry_engine_state' dat:a."  err=0  pid=352
     6 SYS
    20 O+hdoqs   run        1970046139.1.604#TMF_UI::Extd_Desktop# uiserver
        service: ""  err=0  pid=1
    23 O+hdoq   run        1970046139.1.178#TMF_Administrator::Configuration_GUI# launch
        service: "201 attr 'members' dat:"  err=0  pid=132
   176 O+hdoq   run        1970046139.1.353#TMF_InterRegion::Connection# ui_remote_connect
        service: "201 attr 'pres_object' dat:"  err=0  pid=180
   380 O+hdoq   run        1970046139.1.353#TMF_InterRegion::Connection# ui_secure_connect
        service: "201 attr 'pres_object' dat:"  err=0  pid=180
   673 O+hdoq   run        1970046139.1.353#TMF_InterRegion::Connection# ui_get_connections
        service: "201 attr 'pres_object' dat:"  err=0  pid=180
  2415 O+hdoq   run        1970046139.1.168#TMF_Administrator::Collection_GUI# launch
        service: "201 attr 'pres_object' dat:"  err=0  pid=700
  3100 O+hdoq   run        1970046139.1.353#TMF_InterRegion::Connection# ui_get_connections
        service: "201 attr 'pres_object' dat:"  err=0  pid=180
  5686 O+hdoq   run        1970046139.1.713#TMF_TGC::CollectionGUI# launch
        service: "237 attr '' dat:"  err=0  pid=364
  6123 O+      run        1970046139.1.2 query odstat
        service: ""  err=0  pid=0

---- history ----
  5936 O+     done        1970046139.1.2 query whoami
        service: ""  err=0  pid=0
  5937 O+     done        1970046139.1.2 query interp
        service: ""  err=0  pid=0
  5938 O+     done        1970046139.1.2 query db_dir
        service: ""  err=0  pid=0
```

Figure 187. odstat Output

Chapter 6. Policy Management

This chapter shows how to set up policies, describe the function of policies and Policy Regions as well as subregions. It also addresses top level Policy Regions and goes into some of the functions that can be issued from the command line.

6.1 Policy Management

The Tivoli management environment uses policies to enable customization of the environment to suit different organizations. Within the TME a policy is essentially a rule defined for a system, that when deployed, TME administrators can use for management operations. These rules can be simplistic or structured such that they fit an overall enterprise-specific set of rules for systems or operational management.

For example, a large organization has ten major sites around the world. By structuring the top level Policy Regions for this organization such that they reflect the geographic locations, subregions can be created within each top level region to reflect the different departments at each location. Each subregion for a specific department within the organization, which exists at multiple locations, can have the same policies deployed. Sales departments at each of the ten major sites could be an example of this.

These policies are managed and maintained by what are known to the TME as *Policy Regions*. A Policy Region within the TME is a collection of resources that share a set of policies. Policy Regions can be defined to show the overall organizational hierarchy by structuring these Policy Regions within the enterprise network.

Policy Regions contain:

- Managed resources
- Policy Subregions

The managed resources contained in a Policy Region exist in only one Policy Region. They can be moved to other Policy Regions from the Policy Region in which they were created, but they can only be part of a single Policy Region at a time. The managed resources that are available to the Policy Region are dependent upon the Tivoli products and applications that are available to the TME. The basic TME platform provides the following managed resources.

- PcManagedNode - A network node with a TME agent
- ManagedNode - A network node acting as a TME server
- TaskLibrary - A repository for frequently run jobs and tasks
- ProfileManager - A group of profiles which are subscribed to

Each of these resources is known as a TME managed resource class.

Policy Subregions are Policy Regions located within Policy Regions. Initially, policy subregions inherit the resource and policy properties of the parent Policy Region. The properties can be changed after subregion creation to reflect the requirements of that Policy Subregion.

Tivoli Policy Regions check for conflicts before managed resources begin any interaction. They also allow for policy enforcement when new resources are added or configured, so that no resource can contain policy information that will create conflicts with other resources. Policy Regions also allow for a default set of resource properties to be initiated upon resource creation.

6.1.1 Tivoli Policies

Policies within TME consist of a set of rules that can be applied to systems such that the TME enforces this rule as a management operation. TME policies enforce that system resources are created according to some specific management criteria specified within the methods attached to the resource class for that Policy Region. Within the TME there are two possible types of policies that can be established:

- Default policy
- Validation policy

An example of a default policy would be establishing a value for the connection timeout to be set when creating PcManagedNode resources. The value that gets set would be a constant and each time a resource of this type is created, the policy default ensures this connection timeout parameter is set as the default each time.

6.1.1.1 Tivoli Default Policies

A default policy is a set of default resource property values that are assigned to a resource when that resource is created. These can be the resource property values that came as a default with the TME for each resource, or the resource properties can be edited to suit specific needs. These policies can be based on Tivoli profiles or they can be based on other factors. Nonprofile-based policies can be given a constant value or they can be associated with a script.

6.1.1.2 Tivoli Validation Policies

A validation policy enforces rules when creating or altering resources that do not conform to the validation policy employed by that Policy Region in which the resource was created. The validation policy, as would be expected, also ensures that any amendments made to a resource will comply with the validation policy used, hence, no resource should be nonpolicy compliant.

An example of a validation policy would be a script which validates the value entered for the connection timeout each time a PcManagedNode is created. This policy would ensure that even after the default creation of the resource any amendments would have to satisfy the criteria of the script in question.

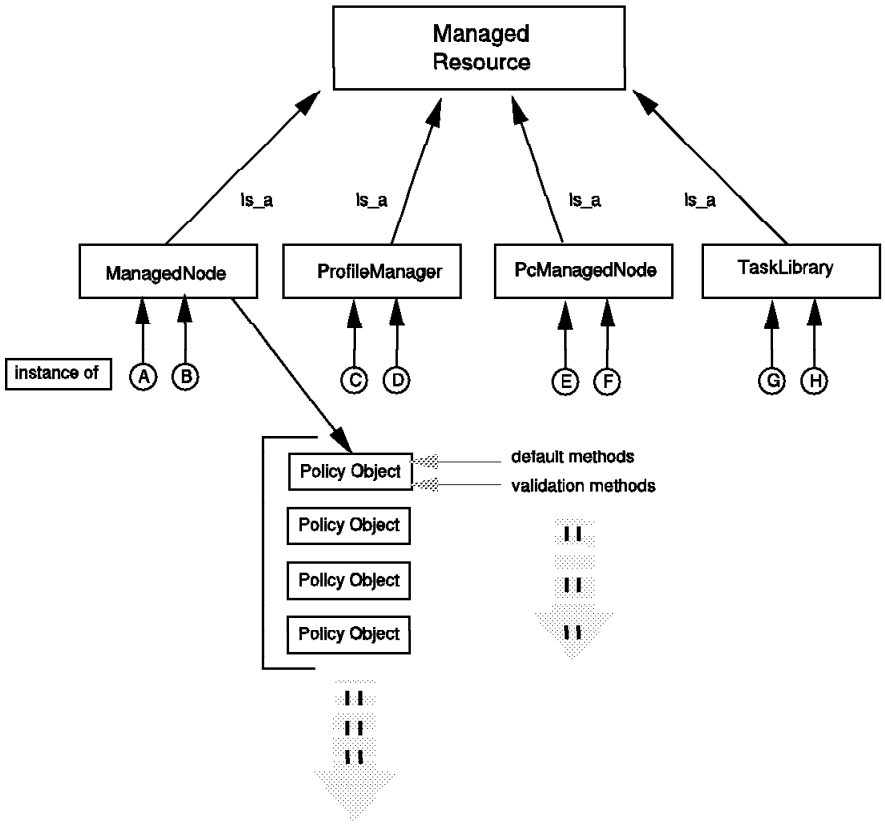


Figure 188. Policy Management

6.2 Policy Regions

Tivoli Policy Regions can be divided into two types:

- Top level Policy Regions
- Policy Subregions

This is analogous to a hierarchical directory structure, with the top level Policy Region at the root with a number of Policy Subregions hanging off the top level Policy Subregion. The top level Policy Region is not unique. There can be, and most often are, many top level Policy Regions, which are analogous to drives on any PC system. These Policy Regions, be they top level or subregions, are a set of managed resources from one or more connected TMRs grouped together to form the Policy Region.

Within the TME, Policy Regions, whether they be top level or subregions, are recognized by the following icon.

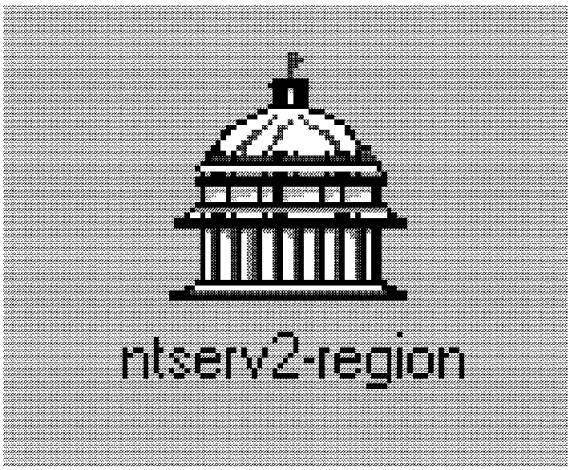


Figure 189. Tivoli Policy Region Icon

Within the next series of diagrams, we can see the structure that Policy Regions follow, from top level Policy Regions through Policy Subregions. We also see the visibility of other top level Policy Regions from connected TMRs.

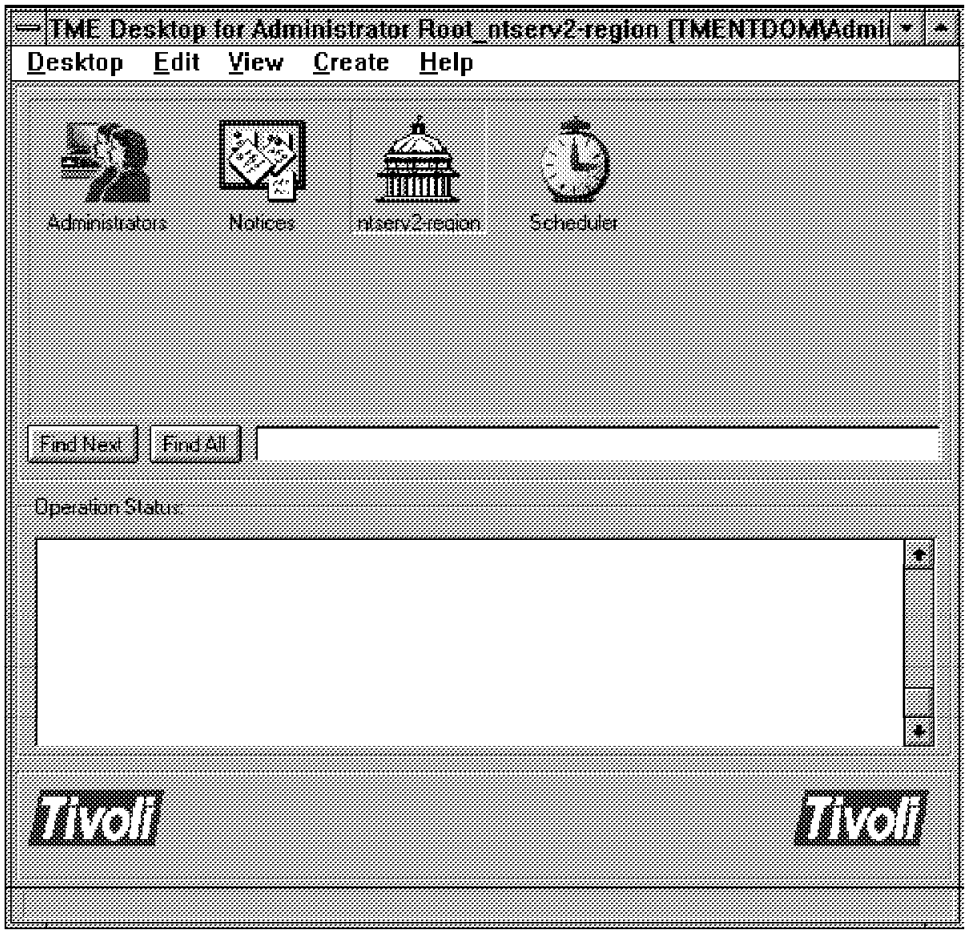


Figure 190. Tivoli Policy Regions - Desktop with Top Level Policy Region

Figure 191. Tivoli Policy Regions - Inside the Top Level Policy Region

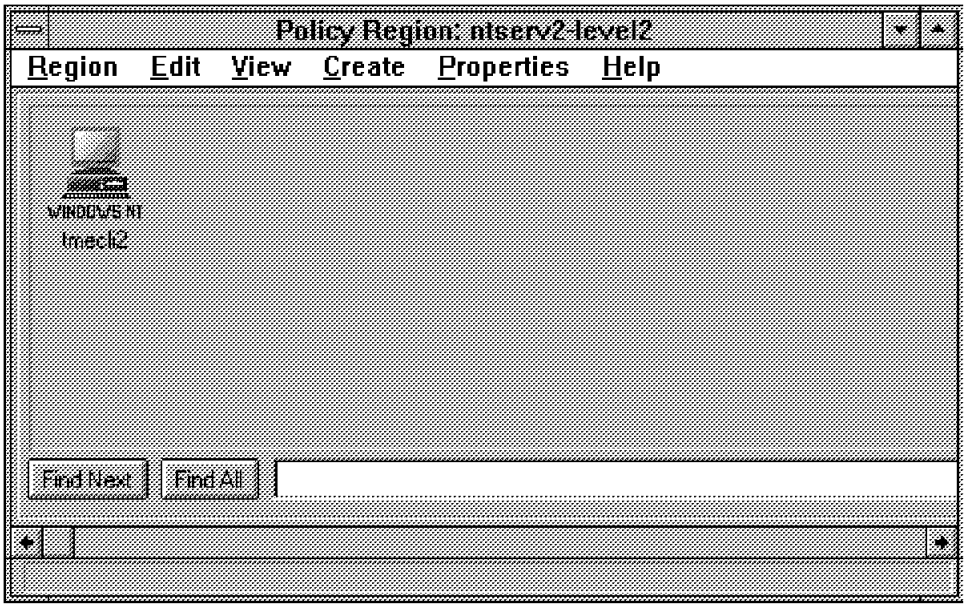


Figure 192. Tivoli Policy Regions - Inside the Policy Subregion

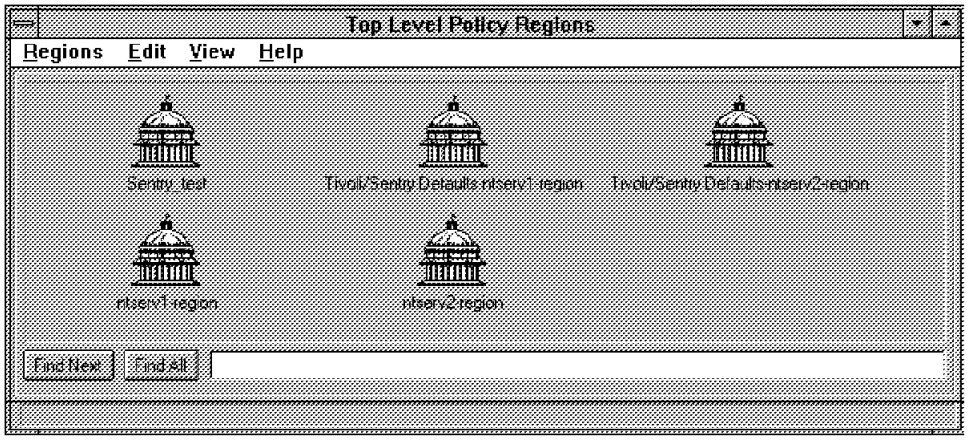


Figure 193. Tivoli Policy Regions - Top Level Policy Regions across TMRs

6.2.1 Top Level Policy Regions

Top level Policy Regions are, as there name suggests, policies viewed from the top level of the Policy Region. They can be used to mirror an organization′s broad structure and depict that structure within the TME. An example of this would be a large organization with a number of departments. These departments could be shown using top level Policy Regions, as it is likely the resources within these departments will be utilized similarly.

Top level Policy Regions are also visible across TMRs, thereby providing the ability for the resources to be remotely managed.

6.2.1.1 Creating Top Level Policy Regions

Top level Policy Regions are created from the desktop interface by selecting the **Desktop** pull-down from the TME desktop, then clicking on **Create** and **Region**. The TME then displays the Create Policy Region dialog as shown in the following figure:

Figure 194. Tivoli Policy Regions - Creating Top Level Policy Regions

In this dialog, entering the name of the new Policy Region in the Name field creates a new top level Policy Region. The name must correspond to a unique name within the local TMR. Selecting **Create & Close** creates the new Policy Region.

6.2.1.2 Changing Top Level Policy Region Properties

Top level Policy Regions consist of only one property, their Policy Region name. Changing this property, involves selecting the top level Policy Region at the desktop, with the right mouse button. A submenu is presented with the following options:

- Open
- Region Properties
- Managed Resources
- Managed Resource Policies

The first option will open the Policy Region selected, displaying all the TME managed resources defined within the region. The second option allows for the altering of the top level Policy Region name.

The Managed Resources option displays all the TME managed resources being used by the top level policy and those available to be used. An example of that follows:

Figure 195. Tivoli Policy Regions - Managed Resources Options

The Managed Resource Policies option displays all of the policy default and policy validation policies for all of the currently managed resources within the top level Policy Region.

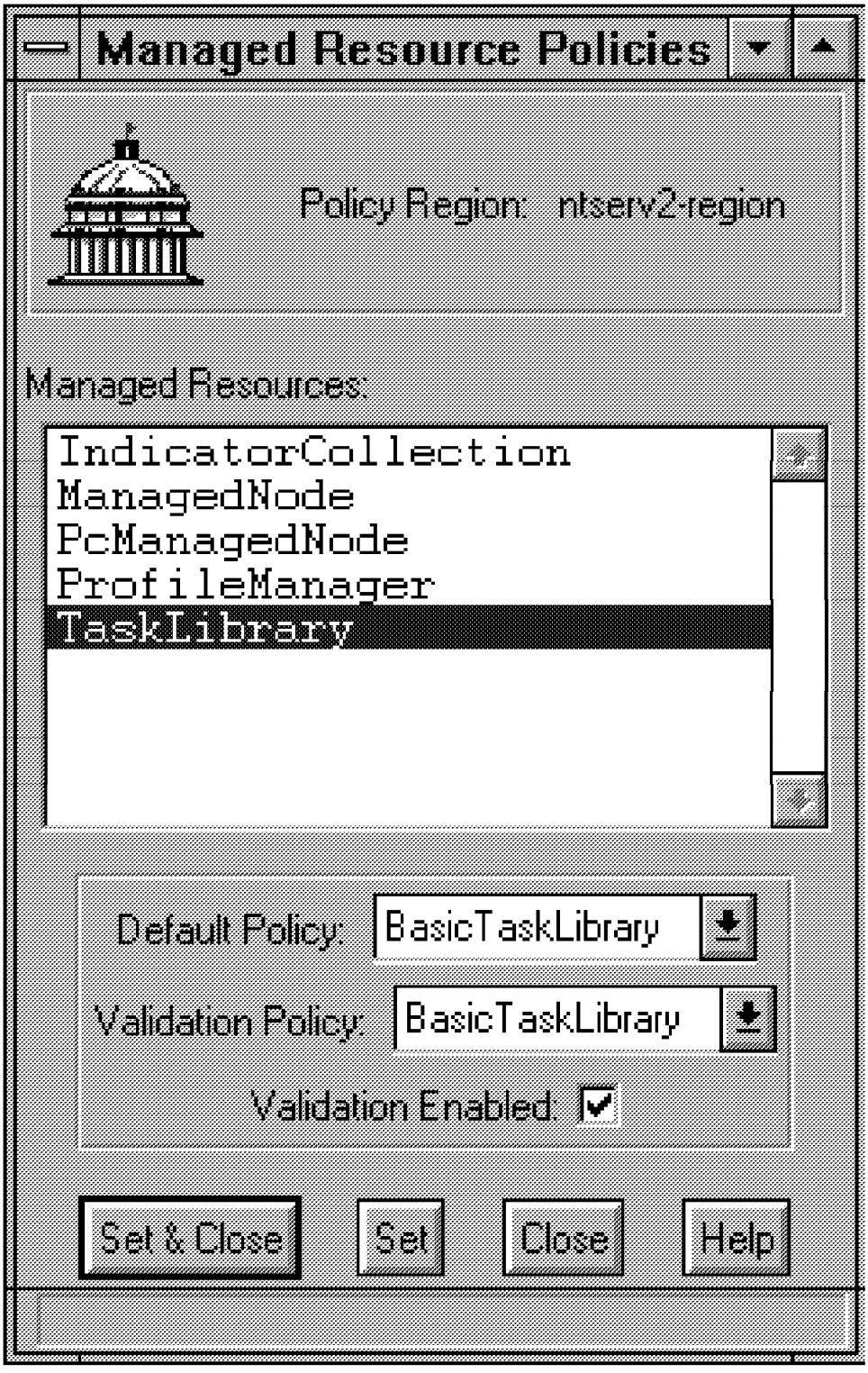


Figure 196. Tivoli Policy Regions - Managed Resource Policy Options

6.2.2 Policy Subregions

Policy Subregions are created from within top level Policy Regions. When a top level Policy Region is opened a subregion may be created by selecting **Create** and **Subregion**. A dialog identical to the dialog used when creating top level Policy Regions is displayed. The procedure to create the subregion is identical to that used when creating top level Policy Regions.

6.2.2.1 Changing Policy Subregion Properties

Policy Subregions consist of only one property, a Policy Region name. Changing this property involves selecting the Policy Subregion icon within the parent Policy Region with the right mouse button. A menu is presented with the following options:

- Open
- Region Properties
- Managed Resources
- Managed Resource Policies

The first option will open the Policy Region selected, displaying all the TME managed resources defined within the region. The second option allows for the altering of the Policy Subregion name.

The Managed Resources option displays all the TME managed resources being used by the subregion, and those available to be used. The Managed Resource Policies option also displays all of the policy default and policy validation policies for all of the currently managed resources within Policy Subregion. This is exactly as was described for top level Policy Regions.

From within Policy Subregions it is possible to alter the properties, Managed Resources and those policies defined for Managed Resources for the Policy Subregions parent Policy Region. This supports the hierarchical object structure of policy management. This is not available to Policy Regions which are at the top level.

6.3 Changing Managed Resource Types

Within each Policy Region there is a defined list of managed resources that can be defined or are valid within that region. Within the TME platform, we have the four basic managed resources as mentioned before. As more and more Tivoli applications are added, then more managed resources become available. These managed resources can be made available or withheld from Policy Regions by adding or deleting them.

The following diagram represents a Policy Region on ntserv1.

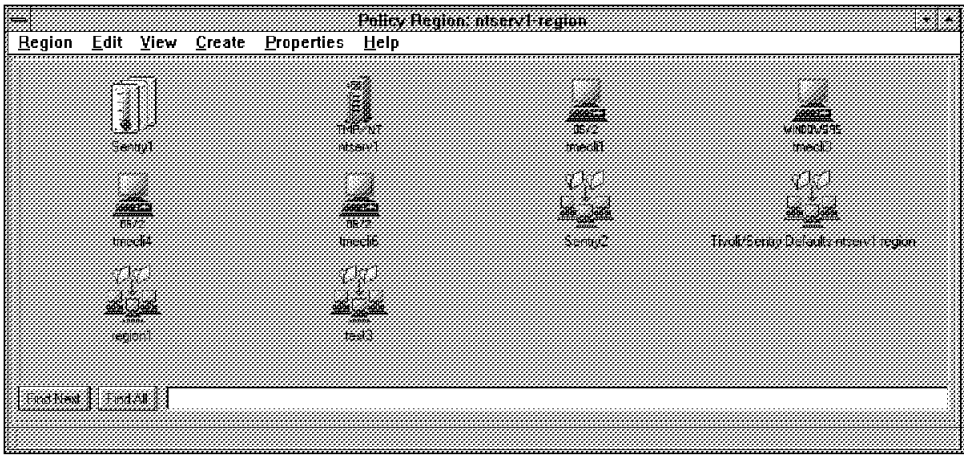


Figure 197. Tivoli Policy Regions - Example Policy Region

By selecting **Properties** and **Managed Resources**, a window representing the current and available resources is presented. This window shows the valid managed resource types available for this Policy Region and those that can be made available.

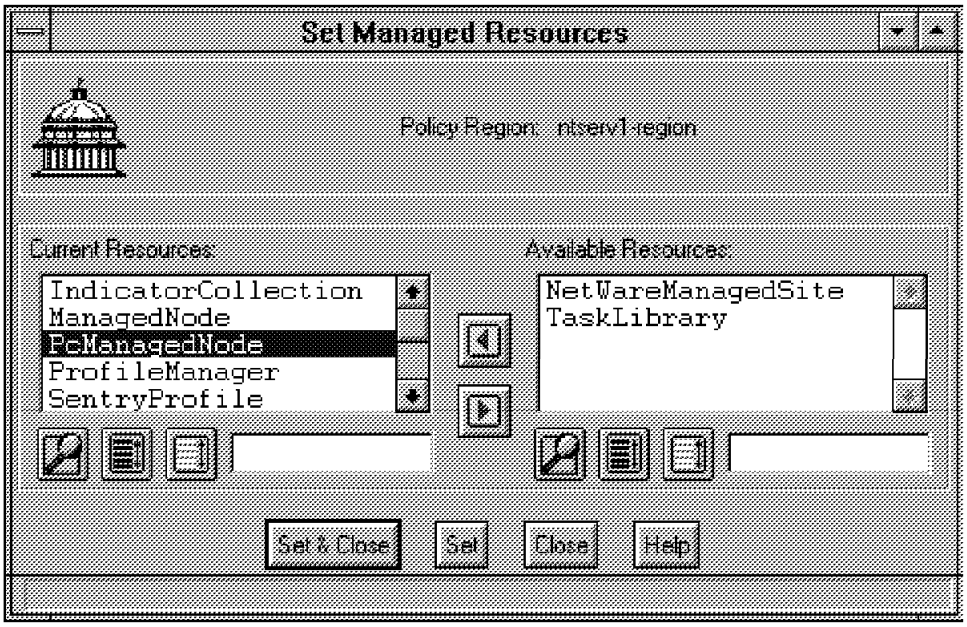


Figure 198. Tivoli Policy Regions - Setting Policy Region Resources

By using the push buttons beneath the window panels, resources can be added or removed one at a time, or all at once. There is also a browser button to list all the resources. The following picture displays the available buttons. The leftmost button is the browser, the central button selects all from the list and the rightmost button de-selects everything selected.

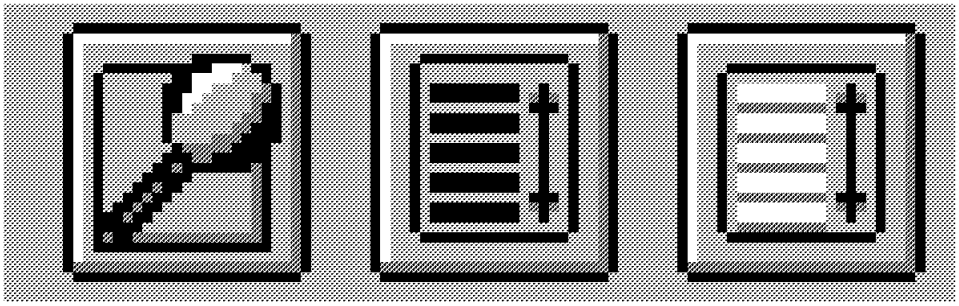


Figure 199. Tivoli Policy Regions - Setting Policy Region Resources

Selecting the browser button presents a textual listing of all the available resources.

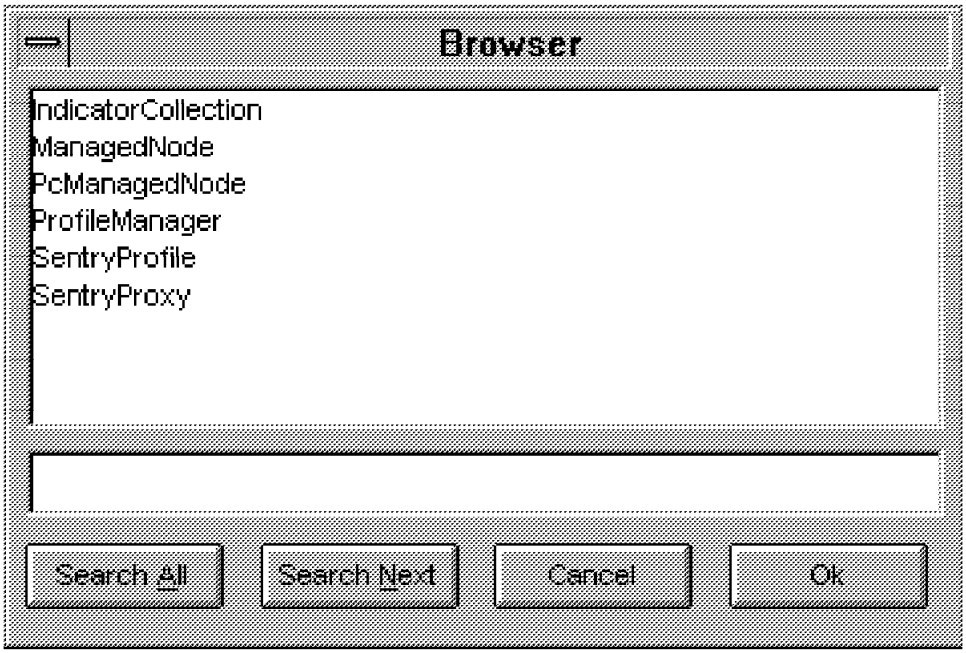


Figure 200. Tivoli Policy Regions - Setting Policy Region Resources

6.4 Assigning Policies to Resources

Within each Policy Region, the managed resources have default policies and can have validation policies assigned to them. These policies can be assigned to managed resources from within the Policy Region by selecting **Properties** and **Managed Resource Policies**. This will present the following window.

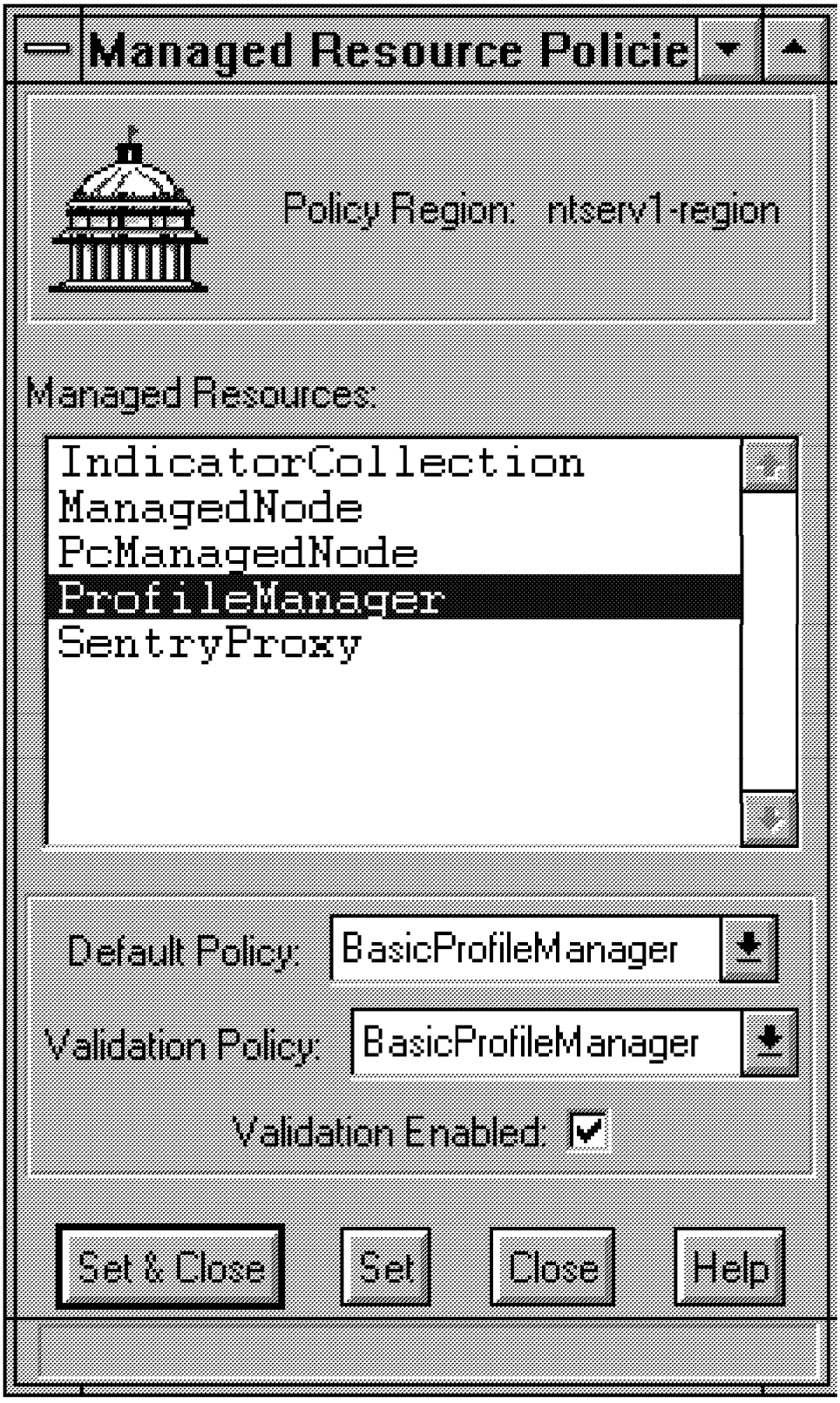


Figure 201. Tivoli Policy Regions - Assigning Policy to Resources

This displays all the available resource within this Policy Region and the current default policy and validation policy for that resource. These policies can be

changed by selecting other created policies from the available list of defined policies for each type.

There is also a check box that enables or disables the validation policy for that managed resource type. A changed default policy for a managed resource will be used by the TME the next time a managed resource of that type is created. A changed validation policy for a managed resource will be used the next time an operation is performed on that managed resource, for example, changing the properties of a managed resource.

6.5 Checking Policies

A policy within a Policy Region can be checked against the managed resources current state to reflect whether those resources conform to the current policies for that region. Moving resources from Policy Region to Policy Region does not validate resource properties for the moved resource within its new region. Therefore, it is important that moved resources should be policy checked to make sure that all the managed resources within each Policy Region conform to the correct validation policies laid out for that region.

It is also the case that when validation policies are modified after new instances are created or modified, then those resources may not conform to the altered policy within the region.

A policy can be checked by selecting from within the desired Policy Region **Region** and **Check Policy**. An example of that follows.

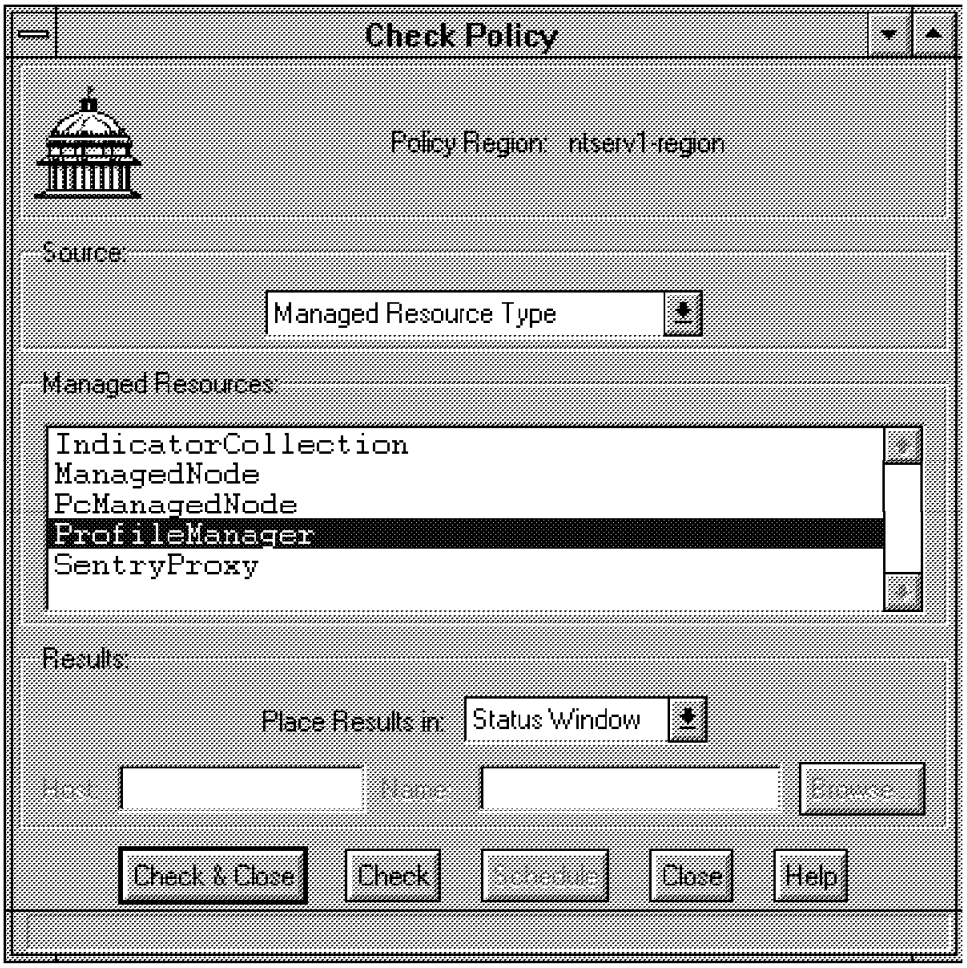


Figure 202. Tivoli Policy Regions - Checking Policy

It is possible to check a policy against a specific managed resource type or against all members of the Policy Region simultaneously, by selecting the appropriate source option from the Source section.

The results of a policy check can be output to any of the following:

- A status window
- A text file
- A collection on the desktop

Policy checks can also be scheduled to occur.

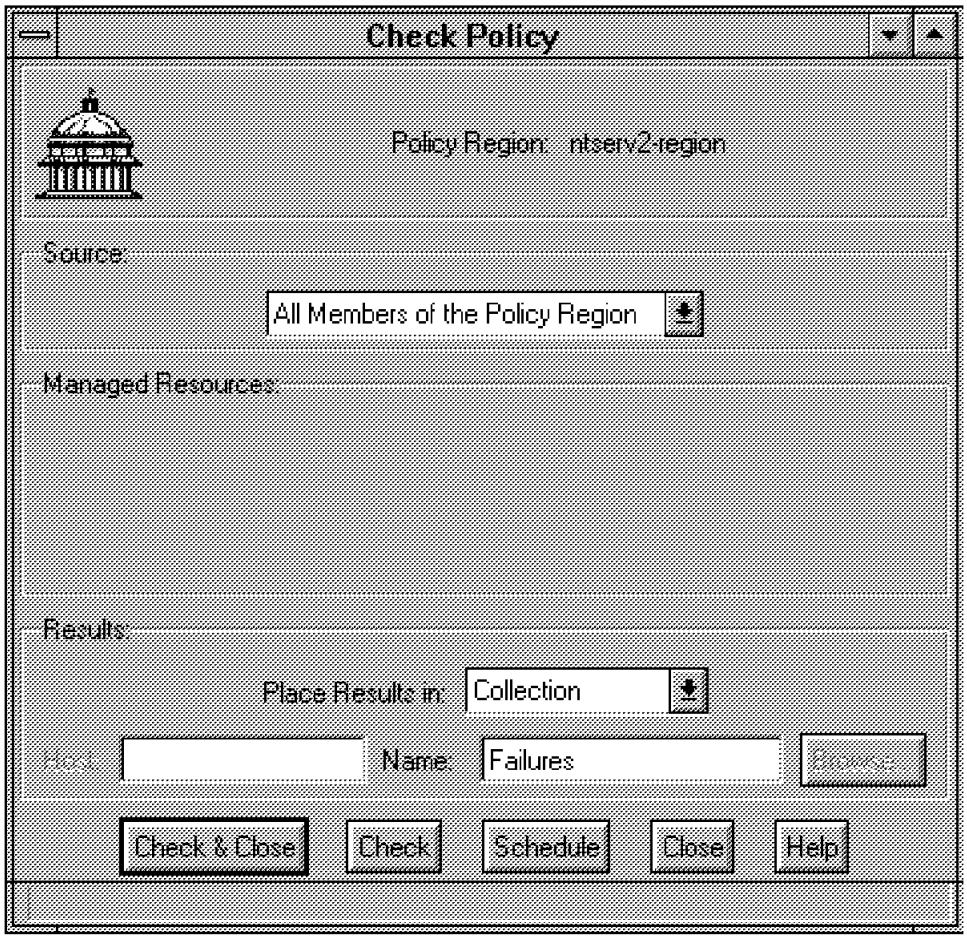


Figure 203. Tivoli Policy Regions - Checking Policy to a Collection

Figure 204. Tivoli Policy Regions - Resources Failing Check in Failures

6.6 Commands Available for Policy Management

The subset of Tivoli commands for policy management allows for the manipulation of Policy Regions, policy objects and policy methods.

6.6.1 Policy Region Commands

The TME command line interface, allows for the following operations to be performed on Policy Regions:

- Creating Policy Regions

  NAME: wcrtpr

  PURPOSE: Creates new Policy Regions

  SYNOPSIS:

  – wcrtpr [-a administrator] ... [-s PolicyRegion] [-m managed_resource] ... name

EXAMPLES:

1. wcrtpr pr_newone will create a new Policy Region with the name of pr_newone. The Policy Region will be empty and not exist on anyone's desktop.
2. wcrtpr -a Root_ntserv2-region pr_newone will create a new policy region with the name of pr_newone. The Policy Region will be empty and will exist on the desktop of the administrator identified with the -a flag. Multiple instances of valid administrators may be supplied.
3. wcrtpr -a Root_ntserv2-region -m tmecli4 pr_newone will create a new Policy Region with the name of pr_newone. The Policy Region will be populated with managed resources identified with the -m flag. Multiple instances of managed resources may be supplied. The Policy Region will exist on the desktop of the administrator identified with the -a flag.
4. wcrtpr -a Root_ntserv2-region -m tmecli4 -s ntserv2-region pr_newsub will create a new policy subregion within the region identified with the -s flag. The region identified with the -s flag is the parent region of the new subregion. Multiple instances of policy subregions may be supplied. The policy subregion will exist on the desktop of the administrator identified with the -a flag, beneath the parent subregion identified with the -s flag.

- Deleting Policy Regions

  NAME: wdelpr

  PURPOSE: Deletes Policy Regions

  SYNOPSIS: wdelpr PolicyRegion

  – wdelpr PolicyRegion

  EXAMPLE:

  1. wdelpr pr_newone will delete the Policy Region specified. The Policy Region specified can be either a top level Policy Region or a subregion.

- Listing Policy Region properties

  NAME: wgetpr

  PURPOSE: Lists Policy Region properties

  SYNOPSIS:

  – wgetpr PolicyRegion

  EXAMPLE:

  1. wgetpr @PolicyRegion:pr_newone will list all the managed resource properties available to this Policy Region:

     ```
     TaskLibrary
     ProfileManager
     ManagedNode
     PcManagedNode
     ```

- Changing Policy Region properties

  NAME: wsetpr

  PURPOSE: Changes Policy Region properties

  SYNOPSIS:

  – wsetpr [[[-d policy_default] [-v policy_validation] [[-e] || *lbrk.-E]]] || [-r]] managed_resource PolicyRegion

  EXAMPLES:

1. wsetpr IndicatorCollection pr_newsub will add the IndicatorCollection managed resource to the Policy Region pr_newsub.
2. wsetpr -r IndicatorCollection pr_newsub will remove the IndicatorCollection managed resource from the Policy Region pr_newsub.
3. wsetpr -d BasicProfileManager -e ProfileManager pr_newsub will set the default policy for managed resource ProfileManager to be BasicProfileManager, within the Policy Region pr_newsub. The -e flag will enable policy validation. Replacing -e with -E will disable policy validation. The other available flag, -v, allows a validation policy to be associated with a managed resource.

6.6.2 Policy Object Commands

The TME command line interface allows for the following operations to be performed on policy objects:

- Creating policy objects for TME resources

  NAME: wcrtpol

  PURPOSE: Creates policy objects for TME managed resources

  SYNOPSIS:

  – wcrtpol [-d | -v] class new_name [parent]

  EXAMPLES:

  1. wcrtpol -d ProfileManager pr_newpol creates a new policy default object for resource class ProfileManager called pr_newpol. To create a new policy-validation object for a managed resource class use the -v flag.
  2. wcrtpol -d ProfileManager pr_newpol BasicProfileManager functions similarly to the previous command, except the new policy default object inherits initial methods and attributes from the policy object BasicProfileManager.

- Deleting policy objects for TME resources

  NAME: wdelpol

  PURPOSE: Deletes policy objects for TME managed resources

  SYNOPSIS:

  – wdelpol [-d | -v] class name

  EXAMPLE:

  1. wdelpol -d ProfileManager pr_newpol deletes the policy-default object defined for the managed resource class pr_newpol. The -v flag can be used to delete policy-validation objects.

- Listing a default policy object

  NAME: wgetdfpol

  PURPOSE: Lists the default policy-default/policy-validation object for a TME managed resource class

  SYNOPSIS:

  – wgetdfpol [-d | -v] class

  EXAMPLE:

1. wgetdfpol -v ProfileManager will list the default policy validation object for the managed resource ProfileManager. Using the -d flag will list the default policy-default object for this managed resource.

- Setting a default policy object

  NAME: wsetdfpol

  PURPOSE: Sets the names of the default policy default/policy validation objects for a specified TME managed resource class

  SYNOPSIS:

  – wsetdfpol [-d | -v] class name

  EXAMPLE:

  1. wsetdfpol -d ProfileManager pr_newpol will set the policy default object for the resource class ProfileManager to pr_newpol. Using the -v flag will set the policy validation object for this resource class.

- Listing available policy objects for a TME resource

  NAME: wlspol

  PURPOSE: Lists the possible policy default/policy validation object names for managed resources

  SYNOPSIS:

  – wlspol [-d | -v] class

  EXAMPLE:

  1. wlspol ProfileManager lists all the policy objects available for the managed resource class ProfileManager. The -d flag lists the policy default objects for the resource class, while the -v flag lists the policy validation objects for the resource class.

- Checking Policy Region members against policy

  NAME: wchkpol

  PURPOSE: Checks the members of a Policy Region, and verifies that they comply with the Policy Region′s enabled policy

  SYNOPSIS:

  – wchkpol [-c collection] [-f file] [-a] [-r resource] [-l member] ... policy-region

  EXAMPLE:

  1. wchkpol -c failed -a pr_newsub will check that all the members of the Policy Region pr_newsub comply with the policy enabled for the region. Those that fail are placed in a collection called failed that is created within this region. The failures can also be directed to a file using the -f flag. The command can be used to check policies for specific resources using the -r flag and for specific members using the -l flag.

6.6.3 Policy Method Commands

The TME command line interface allows for the following operations to be performed on policy methods:

- Listing of policy methods for a TME resource

  NAME: wlspolm

  PURPOSE: Lists all the policy methods assigned to a resource

  SYNOPSIS:

  – wlspolm [-d | -v] class

  – wlspolm [-d | -v] profile

  EXAMPLE:

  1. wlspolm -v ProfileManager lists all the policy methods assigned to the resource ProfileManager. The -v flag lists all the validation policy methods and the -d flag lists all the default policy methods.

- Listing the body or constant value of a policy method

  NAME: wgetpolm

  PURPOSE: Retrieves the body or constant value of a policy object

  SYNOPSIS:

  – wgetpolm [-d | -v] class name policy

  – wgetpolm [-d | -v] profile policy

  EXAMPLE:

  1. wgetpolm -v ProfileManager BasicProfileManager pm_val_subscribers returns the body of the policy method pm_val_subscribers from the policy validation object defined for the TME resource ProfileManager. Once again the -d flag may be used to obtain a default policy method.

- Replacing a policy methods body

NAME: wputpolm

PURPOSE:

SYNOPSIS:

- wputpolm [-d | -v] [-C | -c value] class name policy
- wputpolm [-d | -v] [-n | -C | -c value] [args=a1,a2...] profile policy

6.7 An Example of Policy Management - PcManagedResource

The following script was used to display all the policy default and policy validation objects, as well as all the policy default and policy validation methods for the four base managed resources within the TME platform. The script can be run using:

```
bash pol_obj_meth.sh <Managed Resource>
```

under NT, since the bash shell is available for use under TME NT.

- pol_obj_meth.sh

```
#####################################
#
# Lists policy validation and policy-methods for any TME
# managed resource passed at the command line.
#
#####################################

d_pol_objs="wlspol $1"
d_pol_methods="wlspolm $1"
v_pol_objs="wlspol -v $1"
v_pol_methods="wlspolm -v $1"

echo Resource: $1
echo ---------
echo
echo Policy Default Objects
$d_pol_objs
echo
echo
echo Policy Validation Objects
$v_pol_objs
echo
echo
echo Policy Default Methods
$d_pol_methods
echo
echo
echo Policy Validation Methods
$v_pol_methods

exit 0
```

- bash pol_obj_meth.sh PcManagedNode

```
Resource: PcManagedNode
---------

Policy Default Objects
BasicPcManagedNode

Policy Validation Objects
BasicPcManagedNode

Policy Default Methods
pc_def_connection_timeout

Policy Validation Methods
pc_val_connection_timeout
```

- bash pol_obj_meth.sh ManagedNode

```
Resource: ManagedNode
---------

Policy Default Objects
BasicManagedNode

Policy Validation Objects
BasicManagedNode
```

```
Policy Default Methods
Policy Validation Methods
```

- bash pol_obj_meth.sh ProfileManager

```
Resource: ProfileManager
---------

Policy Default Objects
BasicProfileManager

Policy Validation Objects
BasicProfileManager

Policy Default Methods
pm_def_profile_managers
pm_def_profile_types
pm_def_subscribers

Policy Validation Methods
pm_val_remove_subscribers
pm_val_remove_subscription
pm_val_subscribers
pm_val_subscription
```

- bash pol_obj_meth.sh TaskLibrary

```
Resource: TaskLibrary
---------

Policy Default Objects
BasicTaskLibrary

Policy Validation Objects
BasicTaskLibrary

Policy Default Methods
tl_def_dist_mode
tl_def_man_nodes
tl_def_prof_mgrs
tl_def_set_gid
tl_def_set_uid

Policy Validation Methods
tl_val_man_nodes
tl_val_prof_mgrs
tl_val_set_gid
tl_val_set_uid
```

The above examples display the policy management associated with the default managed resources. We now take a look at the associated policy methods with the managed resource PcManagedNode. As can be seen, it has one policy default method. We can obtain the body of this method by:

```
wgetpolm -d PcManagedNode BasicPcManagedNode pc_def_connection_timeout
```

This returns:

```
180
```

Figure 205. Default Managed Resource Policy Timeout Value

This says that each time a resource of type PcManagedNode is created using the policy BasicPcManagedNode, it will be created with a default value of 180 seconds before the TME times out waiting for connection. This value can be set by entering:

```
wputpolm -d -c 90 PcManagedNode BasicPcManagedNode pc_def_connection_time
```

Therefore, this sets the default connection timeout for all resources of this type to that value.

The resource for PcManagedNode also has one policy validation method. The body of this method is obtained in the same manner as before except the -v flag is used. The *-v* means validation.

```
wgetpolm -v PcManagedNode BasicPcManagedNode pc_val_connection_timeout
```

This returns:

```
#!/bin/sh
##############################################
#
# $Id: pc_val_con_time.sh,v 1.2.2.1 1995/08/01 17:28:58 johnson Exp $
#
# This script implements the "pc_val_connection_timeout" policy
# method for the PC Managed Node class.   It currently ensures
# that the connection timeout value is not less than 5 seconds and
# not greater than 5 minutes
#
# To debug your changes you could add the lines:
#
#  set -xv
#  exec > /tmp/debug.output 2>&1
#
# These lines will allow you to see any errors that occur by looking
# in the /tmp/debug.output file.
#
##############################################

connection_timeout=$1
shift 1

if [ $connection_timeout -lt 5 ]; then
 echo FALSE
 exit 0
fi
if [ $connection_timeout -gt 300 ]; then
 echo FALSE
 exit 0
fi
echo TRUE
exit 0
```

Figure 206. Get the Policy Value for Connection Timeouts

This says that each time a resource of type PcManagedNode is created using the policy BasicPcManagedNode, its value for TME connection timeout has to be within 5 and 300 seconds otherwise the creation will fail. The connection timeout can be edited, but there are rules governing its boundaries.

This policy validation can be altered by entering:

```
wputpolm -v PcManagedNode BasicPcManagedNode pc_val_connection_timeout <
t.sh
```

The script script.sh could be another script with different values for connection timeout validation.

Chapter 7. Configuration Management

Today's machines and systems that are working on common tasks are often grouped together in an office or workgroup. With TME you can logically place common configuration information for machines used for similar purposes into a centralized area. This makes it easier to access, manage and prevent duplication of resources. TME provides a centralized point to control data distribution to groups of systems. You can develop prototypes, known as *profiles*, distribute them across the network and apply them to heterogeneous machines. Information about the profiles is stored in a central location, but can be distributed to other locations.

7.1 Profiles and Profile Managers

A profile contains a collection of application-specific information. Each item in a profile contains system configuration information. The information in a profile is specific to the particular profile type. Profile records are stored in a platform-independent format that allows it to be distributed to different environments without having to go through a translation process. The profile hierarchy contains two pieces:

- Profile Manager - A set of profiles that are subscribed to as one unit by individual profile endpoints.
- Profile Endpoints - A system resource that is the final destination of a profile.

When you modify profiles in a profile manager, the modifications are stored in the associated profile database. The profiles can then be distributed to subscribers. These distributions are copies of the original profiles, which remain on the profile manager site. When the information is distributed to a profile endpoint, TME converts the platform-independent data into the appropriate format for the registered application method that resides at the profile endpoint.

7.1.1 Creating Profile Managers

The first thing you have to do before you start working with profiles is to create a new Profile Manager. In order to perform this action the role of senior is required. You can define a Profile Manager from the desktop or from the command line. We documented both ways.

To create a Profile Manager you have to open your Policy Region on the TME Desktop. In the menu bar you have to choose the option **Create** and then **Profile Manager**. The window for the Profile Manager definition appears. You then fill in the name of the new Profile Manager and confirm it with the **Create & Close** button.

Figure 207. Creating a Profile Manager

A new icon appears in your Policy Region window.

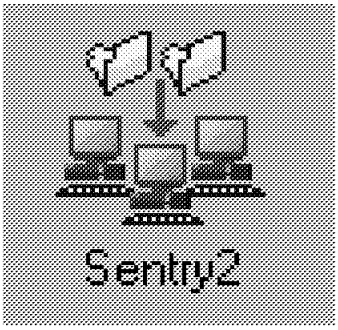


Figure 208. Profile Manager Icon

From the command line you have to use the wcrtprfmgr command to create a new Profile Manager. The syntax of the command looks like the following:

```
wcrtprfmgr pol-region name
```

The parameters are:

- pol-region stands for the name of the Policy Region where you want to create the new Profile Manager.
- Name is the name of the new Profile Manager.

Examples:

```
wcrtprfmgr @ntserv1-region Sentry2
```

The *@* sign is for addressing a sub-object of the TME desktop. Any extension or another sub-object can be added with colons to the first object that includes the addressing sign.

After you create the new Profile Manager you can open the main window by double-clicking on the icon of the Profile Manager.

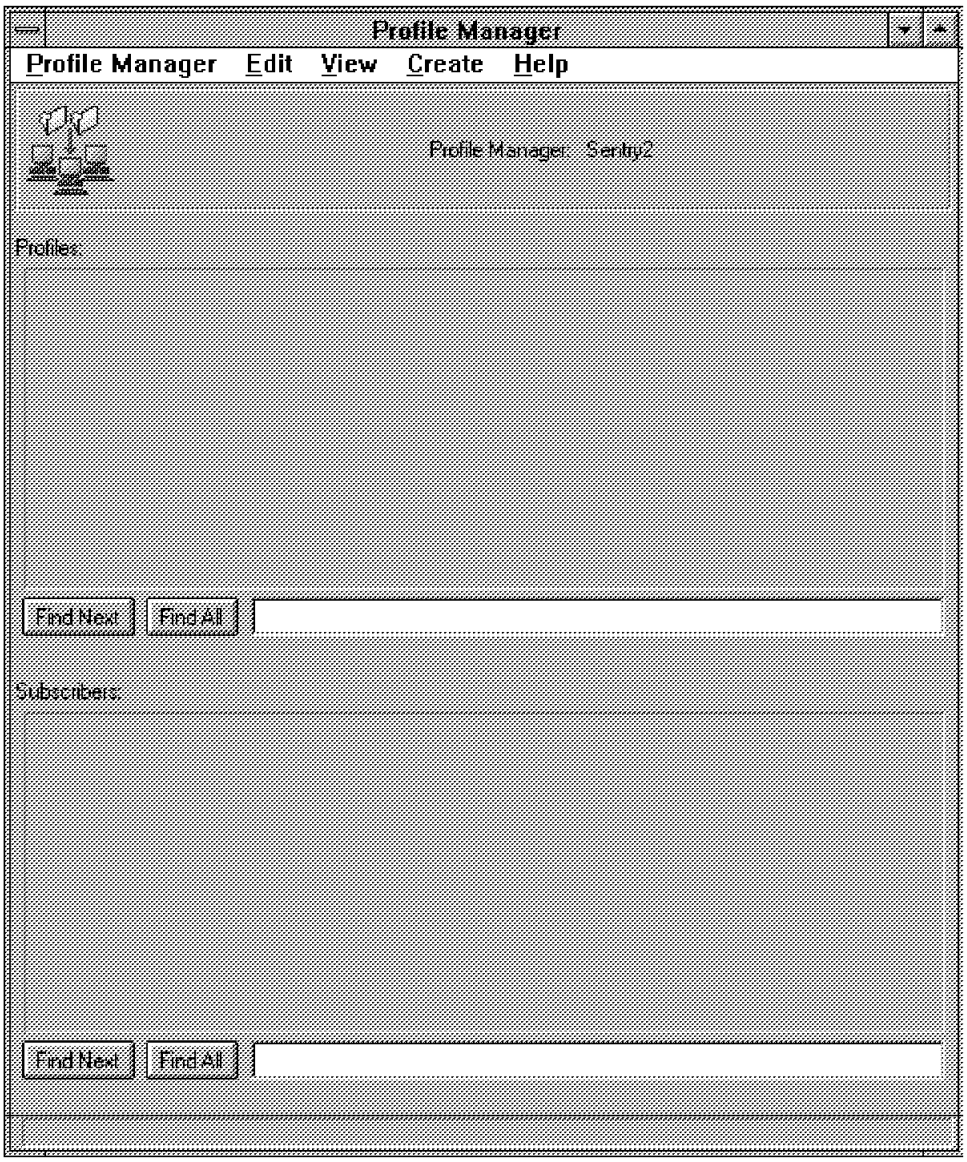


Figure 209. Profile Manager Main Window

The initial look of the window is not very exciting, because it is empty. In order to have resources included in the Profile Manager, you have to populate the Profile Manager with resources and profiles. You can do the following in the Profile Manager window:

- Create new profiles
- Edit existing profiles
- Add subscribers and delete subscribers

7.1.2 Defining Subscribers

The first thing you should do is to define the subscribers. Subscribers are systems which will receive profiles from the Profile Manager that they are defined to. Subscribers can be endpoints or other Profile Managers. All subscribers must conform to the validation policies. We re-address the policies later in this chapter. To create a subscription you need the role *admin*. There are several ways to add subscribers. You can do it in the following ways:

- Through the context menu of the Profile Manager icon
- Trough the menu bar in the Profile Manager window
- Drag and drop from the Policy Region

When you open the context menu (click with right mouse button on the icon) of the Profile Manager you can do several things:

- Open the Profile Manager
- Distribute profiles
- Get a new copy of a profile
- Set subscribers
- See subscriptions for the Profile Manager

When you choose **Subscribers** the Subscribers window opens as shown in Figure 210.

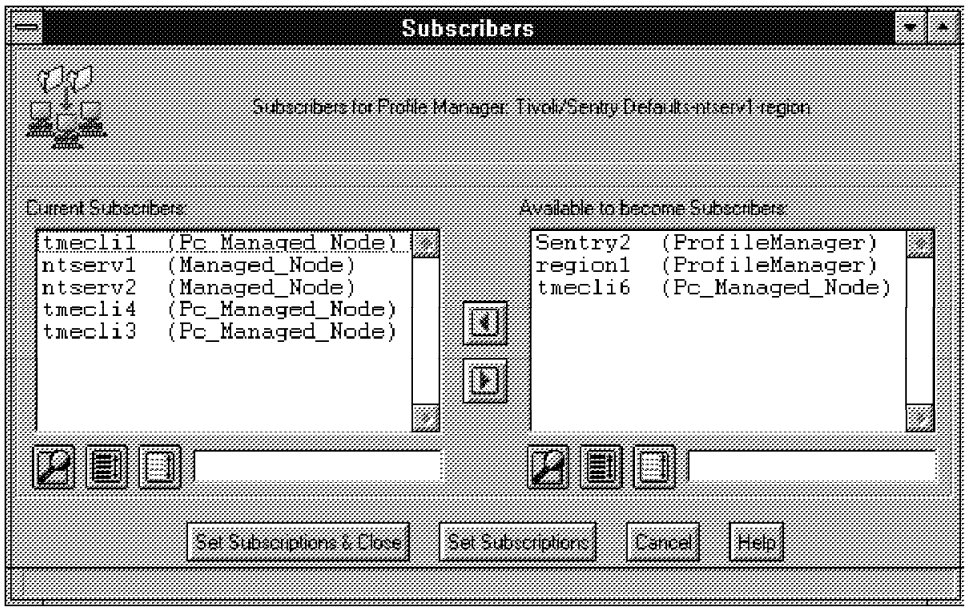


Figure 210. Subscribers Window

The right list box shows the available, potential subscribers. The left one shows the current subscribers. You can choose resources from the right list box and add them to the left with the two buttons in the middle between the two boxes. After you have finished your subscriptions choose the **Set Subscription & Close** button to confirm your definition. The new subscribers appear in the upper part of the main window of the Profile Manager.

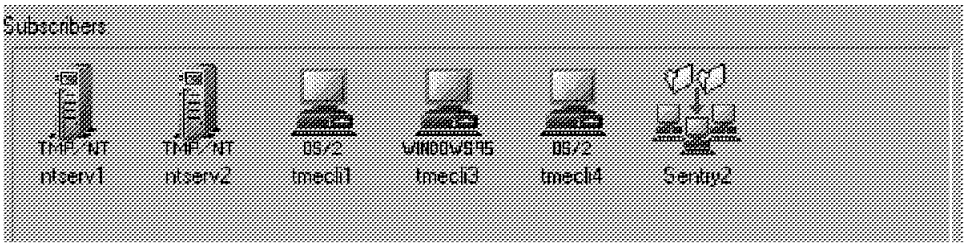


Figure 211. Types of Subscribers

The second way to add or delete subscribers is through the menu bar in the main window of the Profile Manager. You would use the pull-down from the **Profile Manager**. When you open the menu there is an option to select **Subscribers**. You will see the same window for the subscriber list that you saw in the previous example.

Another way is to *drag and drop* from the Policy Region. You can select a Profile Manager or a system in your local TMR or in any connected TMR and copy it using drag and drop on to the Profile Manager. The option subscription shows the subscriptions of the current Profile Manager to other Profile Managers. This option is also available through the context menu on the desktop GUI. You can also use the command line to set subscribers. The command is the wsub command. The format of the command is:

```
wsub name subscriber
```

The parameters are:

- Name - Name of the Profile Manager
- Subscriber - Name of the resource added to the Profile Manager

Examples:

```
wsub @Sentry2 @PcManagedNode:tmecli1
```

```
wsub @Sentry2 @PcManagedNode:tmecli6 @ProfileManager:region1
```

7.1.3 Remove Subscriber

To remove a subscriber you can use the command line interface or the desktop. From the desktop you have to open the **Subscriber** window and remove resources. The process is the reverse of adding them, where you just select them and use the right arrow to move them back to the available, but not current, list. Another way is the use of the command line. The command is the winsub command. The following is the format of the command:

```
wunsub -a -l name subscriber
```

The parameters and options are:

- -a - All subscribers are removed.
- -l - Maintain s the database tool.
- Name - Name of the Profile Manager.
- Subscriber - Name of the subscriber.

Examples

- `wunsub -l @Sentry2 @Managed Node:tmecli4`
- `wunsub -l @Sentry2 @ProfileManager:ntserv2`
- `wunsub -a -l @Sentry2`

Note

Keep in mind the following:

- Always use the *@* at the start of an address of a node or a resource in the Policy Regions.
- Use the correct resource type: PcManagedNode and ManagedNode.

7.1.4 Profiles

The next important part of the Profile Manager are the profiles themselves. As mentioned above, profiles contain items for monitoring specific system resources on other machines. The profiles are used for distributing system management information to endpoints and other profile managers and to take care of their configurations. This includes changes. To work with profiles you have to look for the following prerequisites:

- Tivoli TME itself has no profiles pre-defined or included in the framework. Thus you have to install an application to work with the profiles. An example of this would be Sentry.
- You have created a Profile Manager to work with the profiles.

To create a new profile you can use the desktop or the command line. To do it from the desktop interface requires you to start from the Profile Manager main window. Choose the option **Create** in the menu bar. There is only one item in the pull-down, the profile. After you have selected the profile to create, the window for the definition appears.

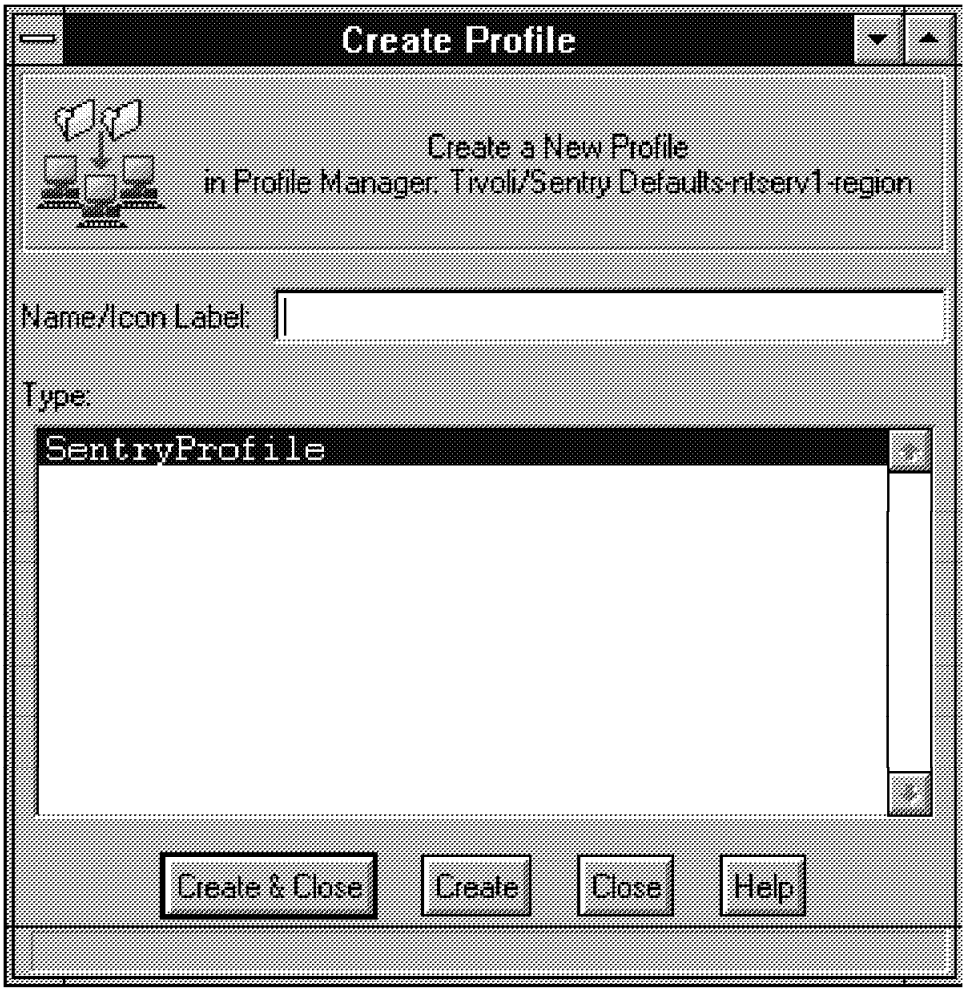


Figure 212. Defining a New Profile

You have to fill in a name for the new profile and select a type from the the list box. The more applications you install for the TME desktop, the more profiles are available to use. To confirm the definition select the **Create & Close** button. After that, a new profile appears in the Profile Manager window.

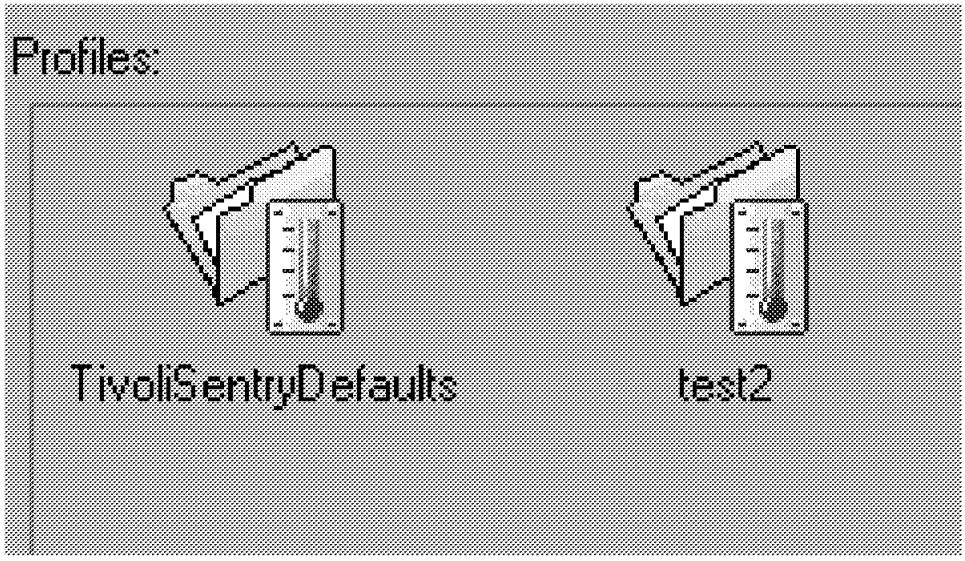


Figure 213. Profile Icons in the Profile Manager

From the command line use the wcrtprf command:

```
wcrtprf -c source profile_manager type profile_name
```

The parameters and options are:

- -c source - Specifies the name of an existing profile to clone from
- profile_manager - Name of the profile manager to which the profile will be created
- Type - Type of profile (here:SentryProfile)
- profile_name - Name of the new profile

The difference between cloning and creating a profile is that cloning copies the policies of the original profile. Creating it makes a completely new profile without any specified policies.

Examples:

```
wcrtprf @ProfileManager:Sentry2 SentryProfile CPU
```

```
wcrtprf -c @SentryProfile:CPU @ProfileManager:Sentry2 Sentry Profile harddisk
```

You can perform cloning from the desktop. To do this select the option **Edit** from the menu bar. In the pull-down select **Profiles**. In the submenu for Profiles you can find the clone option. To clone a profile you must first select a profile from the desktop, then you can choose the clone option. A window, similar to the Create window, opens.

Figure 214. Cloning Window

In the window you have to fill in the name of the new profile and you have to select a destination Profile Manager where you want to clone to. This gives you the chance to clone to and from connected TMRs.

7.1.4.1 Architecture of a Profile

When you open the new profile in the Profile Manager window you get the following:

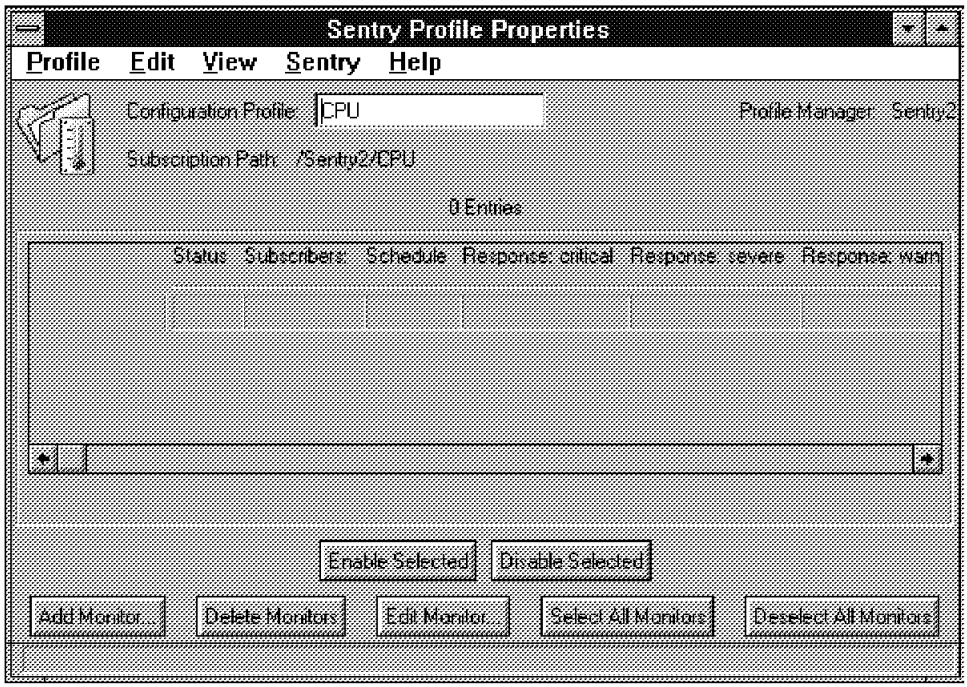


Figure 215. Profile Icons in the Profile Manager

To explain the profiles and their contents we start with an example. To understand a profile you must see it work. We installed Tivoli Sentry as the application. It gives us the base to work from for our profiles. See 7.1.12, “NT Sentry Examples” on page 241 for an example of a monitor that we set up. The following section just shows how to distribute a profile, not how to set up the monitors.

7.1.5 Distributing Sentry Profiles

To show how to work with profiles we use the Tivoli/Sentry application again. After creating a profile, setting subscribers and populating the profile with monitors, you can start a distribution of the profile to the subscribers. To perform these actions you need the admin role. Open the profile you want to distribute, then select **Distribute** from the Profile menu. The window for the distribution settings opens.

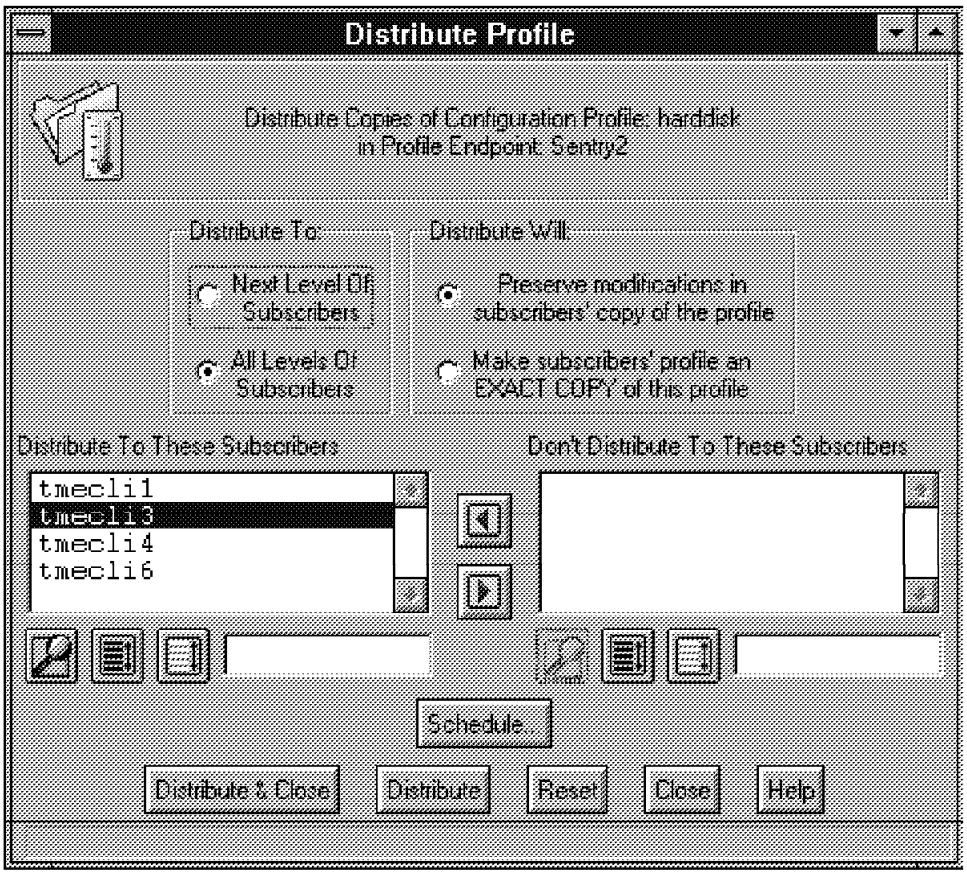


Figure 216. Distribution Window

There are several actions to be taken:

1. First you have to decide to which level you want to distribute. You can distribute to the Next Level of Subscribers, which means that only subscribers of the current profile manager are included in the distribution, or you can choose the All Levels of Subscribers option, which allows you to distribute the profile to all machines.
2. Then you have to decide about the distribution will. The option Preserve modifications... takes care of modifications made locally to the profiles from the user. The EXACT COPY option overwrites existing modifications.
3. In the list boxes you can see the subscribers you distribute to in the left list box. If you want any station excluded, mark it and use the buttons between the list boxes to change its state from distribute to don't distribute.
4. You can set times for the distribution or you can distribute them right away. To schedule the distribution, select the **Schedule** button. The window for setting the schedule opens.

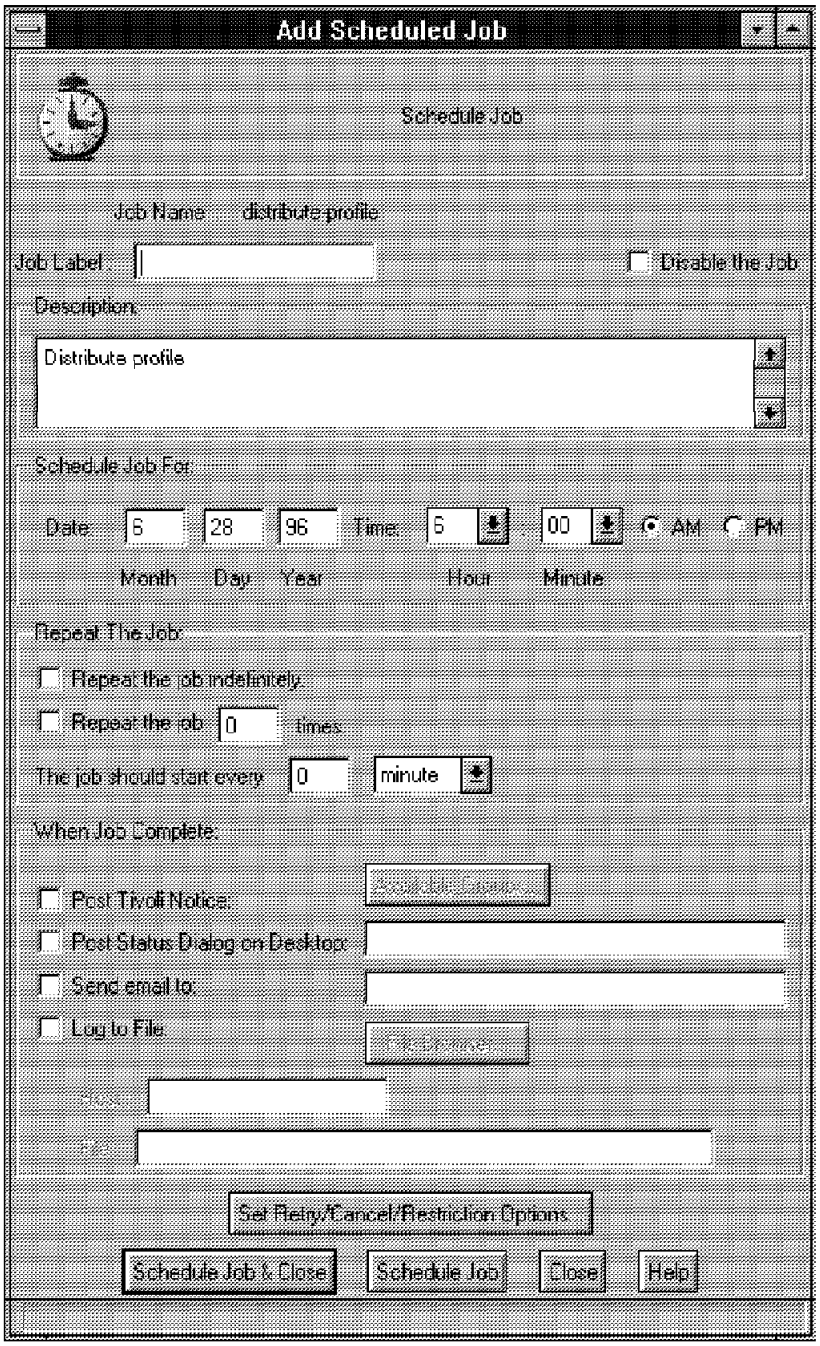


Figure 217. Schedule Window

You have to provide a name for the new job and decide about the following fields:

- When to schedule it
- How often to repeat the job
- What to do when the job is finished

For further definitions, choose the **Set Retry/Cancel** button. In this window you can set options for retrying and cancelling of the job. You can also set restrictions for when the job should start.

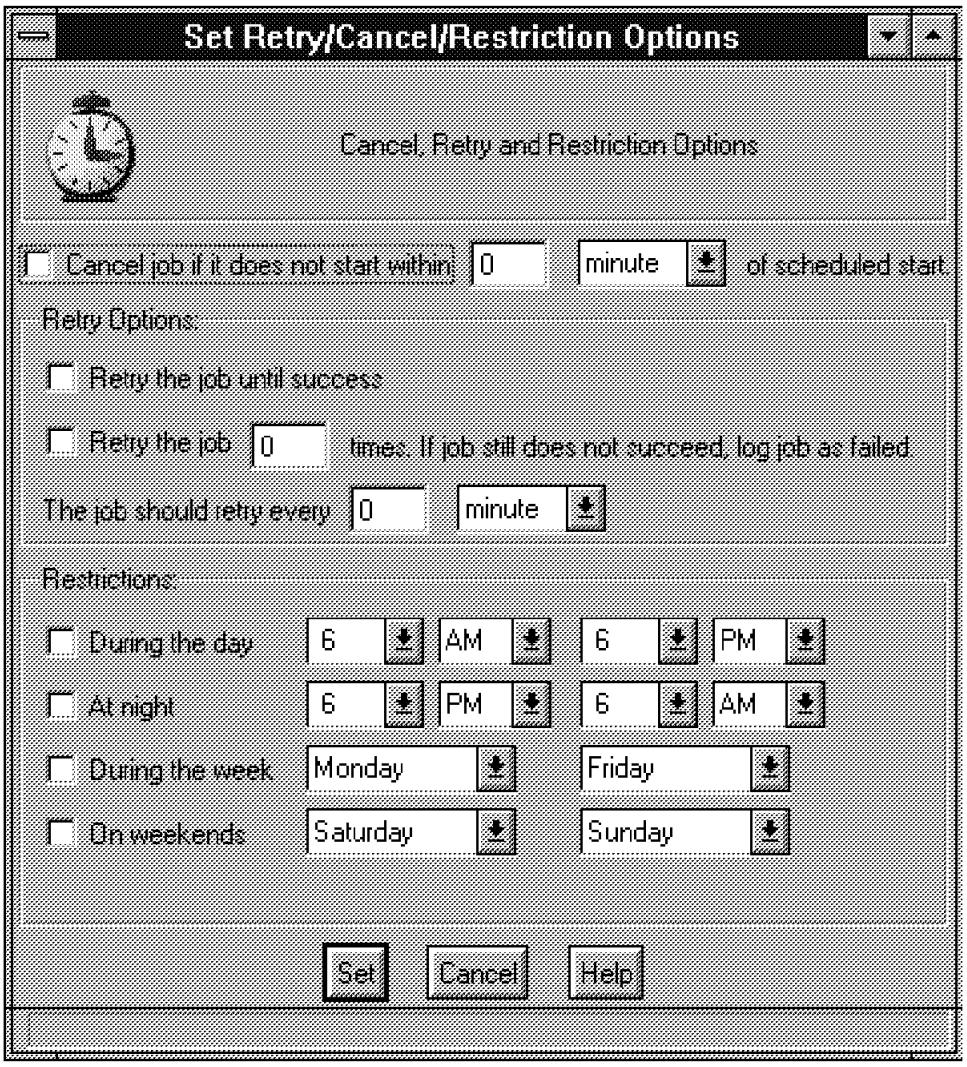


Figure 218. Set Retry/Cancel/Restriction Options

After you determined what options you required, select the **Distribute & Close** button to confirm.

Note: You can only distribute to *Managed Nodes* or *proxy endpoints*. That means the subscriber must have a Sentry installation - Server/Engine or Client. If there is no Sentry software installed on the machine, you will get the following error message:

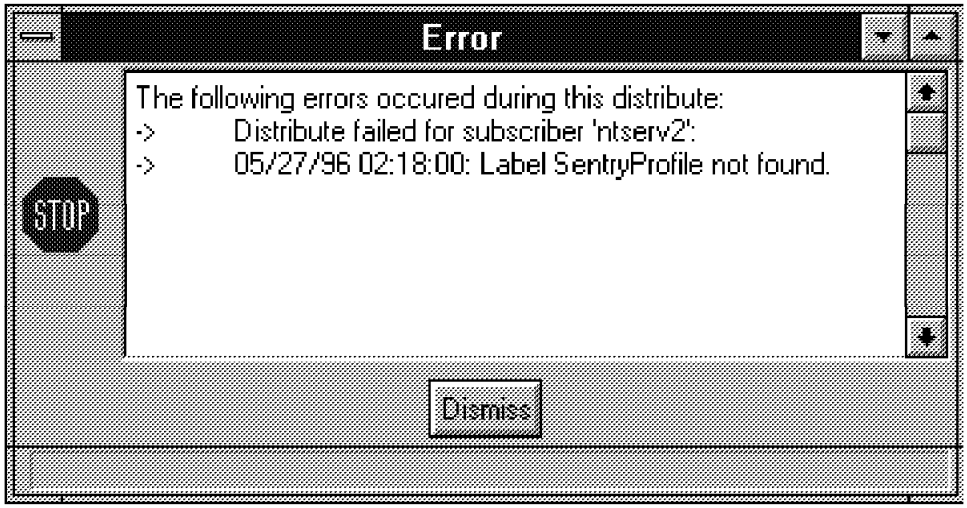


Figure 219. Error

For OS/2 machines that are subscribers to the current profile manager you have to choose **Next Level** and then EXACT COPY or **Preserve modifications** in the Distribute window. Other combinations with the All Level option cause the following problems:

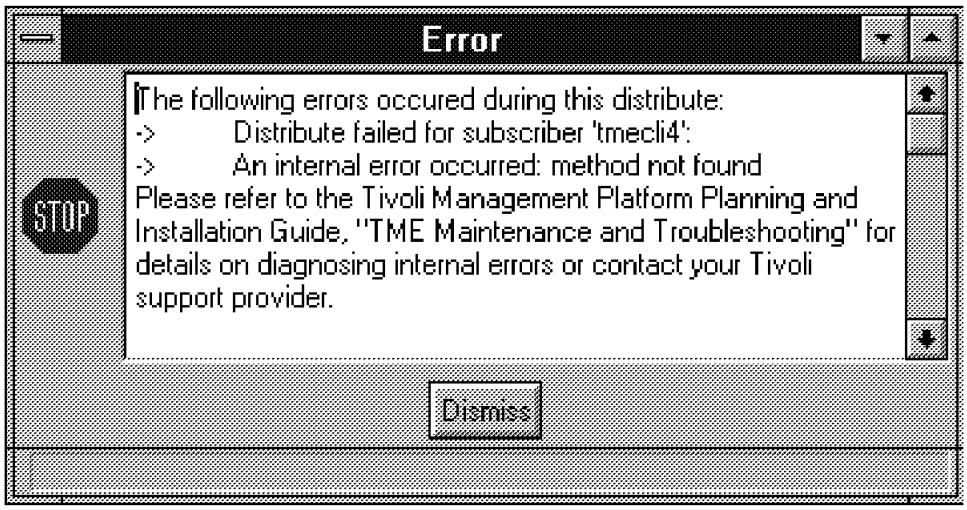


Figure 220. Error

It works the same way for other endpoints (Windows 95, 3.11) because they don't have any subscribers since they are not a server platform. Therefore, you must be careful which systems you distribute your profiles to.

From the command line you can use the command wdistrib to start a distribution. The syntax is:

```
wdistrib -l maintain over_all -m name subscriber..
```

The parameters are:

- -l specifies the distribution level (see maintain and over_all).
- -m specifies a multi-step distribution.
- Name is the name of the profile to distribute.

- Subscriber is the name of the subscriber(s) to distribute to.
- over_all overwrites local modifications.
- Maintain keeps local modifications.

Example: wdistrib /Regions/ntserv1-region/Sentry2/harddisk @ManagedNode:ntserv2 wdistrib -l over_all @SentryProfile:harddisk @ManagedNode:ntserv2 wdistrib -l over_all @SentryProfile:harddisk @PcManagedNode:tmecli1

7.1.5.1 Setting Default Policies

You can set a default user ID and group IDs from the Sentry Profile Properties **Edit**, then **Set User & Group ID** pull-down menu for ownership of the following:

- Monitors
- User-defined scripts
- Log files
- Executable files

These values are relevant for all current and all future monitors in the current profile.

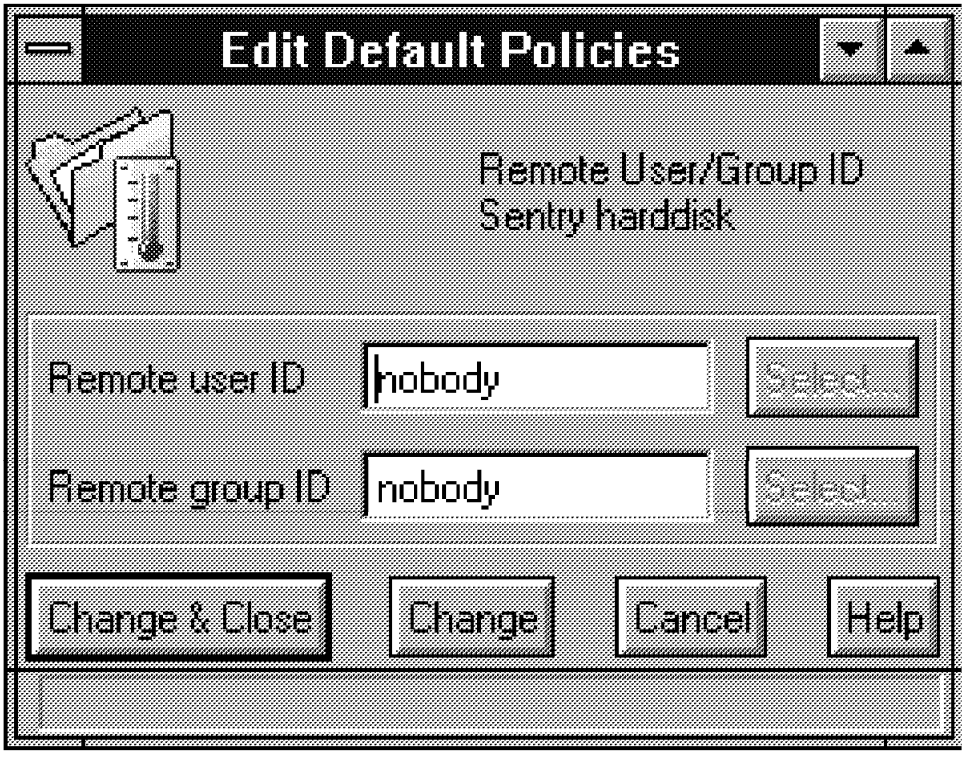


Figure 221. Default Policies Window

No group IDs exist on the NT platform. The value can be distributed but it is ignored.

The command line for this action uses the wsetsntid command:

```
wsetsntid user-id group-id profile-name
```

The parameters are:

- user-id - User ID value
- group-ID - Group ID value
- profile-name - Specifies the profile for the settings

Example: `wsetsntid admin Administrators harddisk`

7.1.5.2 Default Monitoring Schedule at the Profile Level

This setting controls how often all the monitors in a profile check the monitored resources. The schedules set at the monitor level override those set at the profile level. To set the profile schedule open a profile window and select **Set Default schedule** from the Edit menu. The dialog that appears is similar to that which we have discussed for the monitor. The Monitoring Schedule screens look like the following two screens.

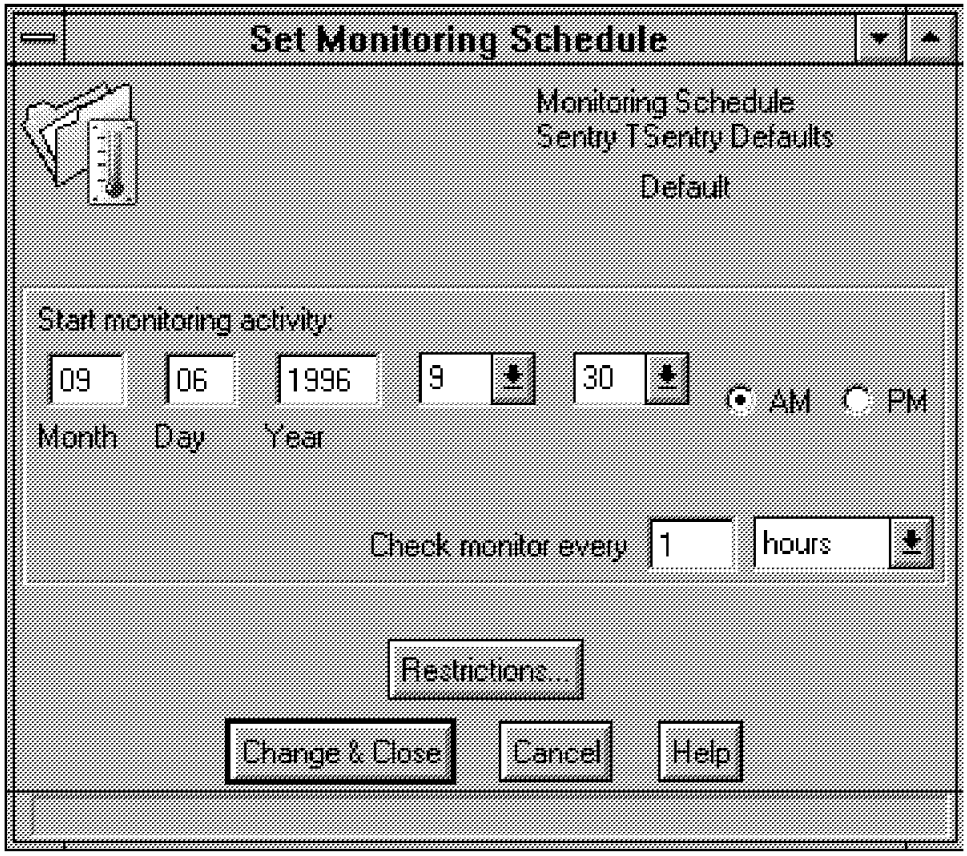


Figure 222. Set Monitoring Schedule

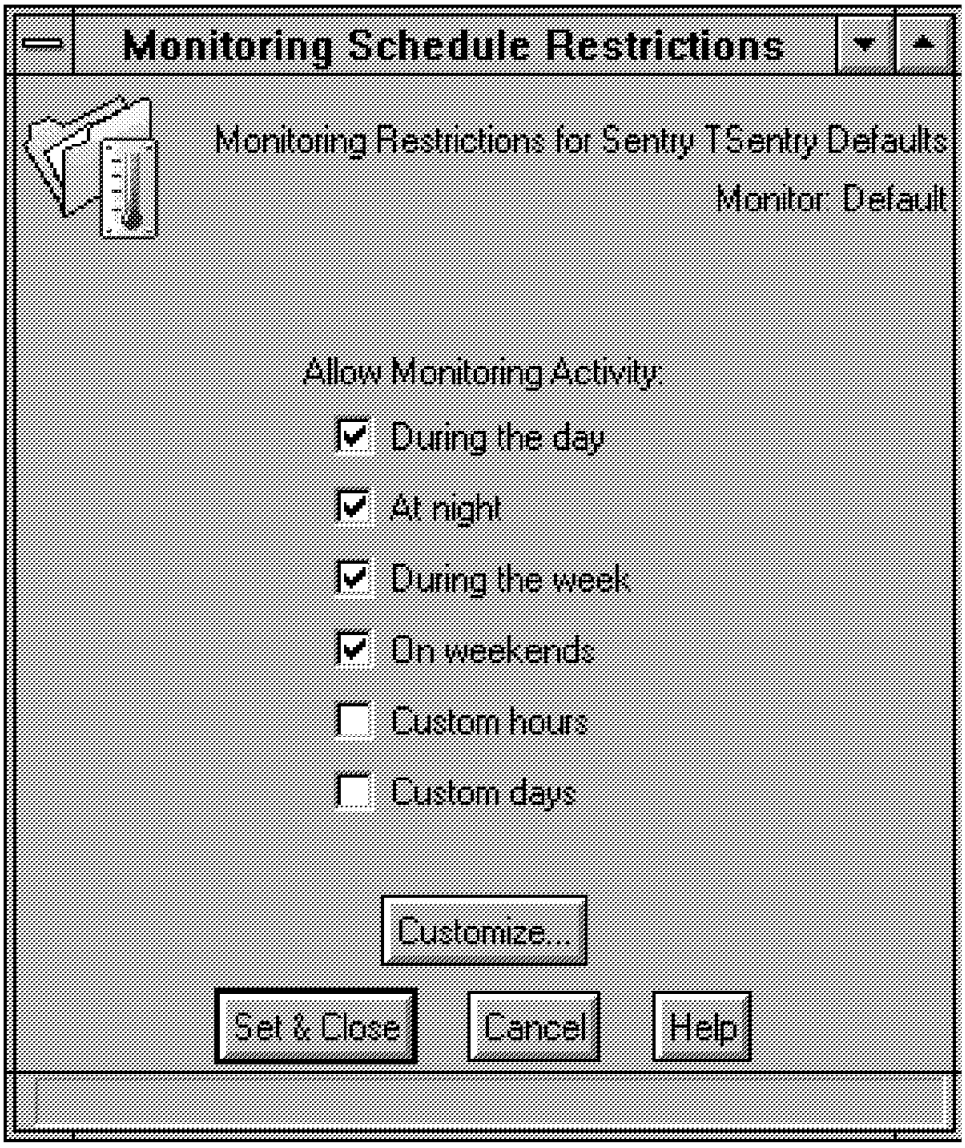


Figure 223. Restrict the Monitoring Schedule

7.1.5.3 Distribution Actions

From the Sentry Profile Properties window, select the **Sentry** pull-down window, then **Edit Distribution Actions**.

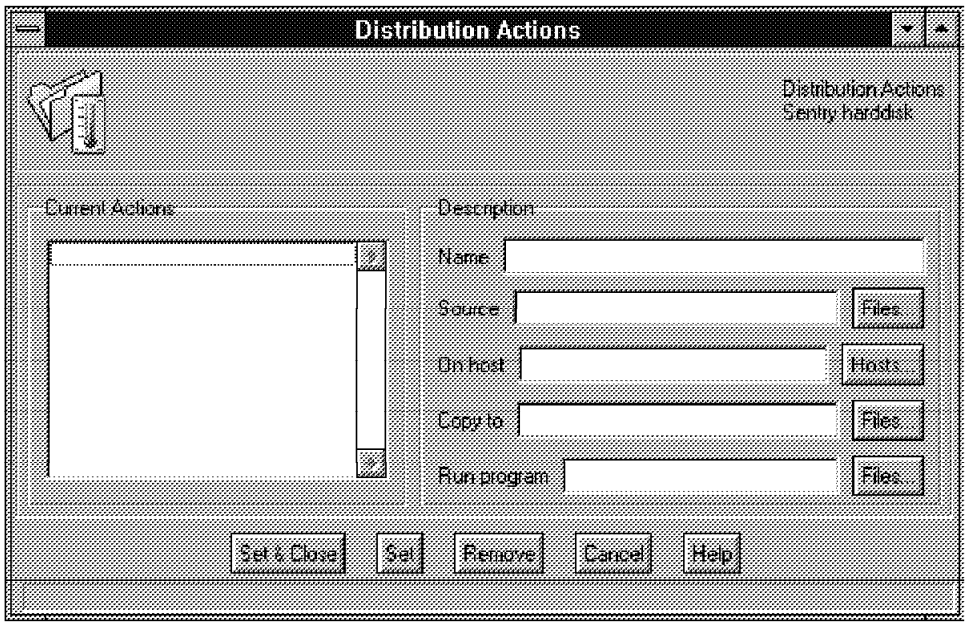


Figure 224. Distribution Actions Window

Distribution actions are performed once a distribution is started. These can be actions such as running a script file or copying a file. The user and group ID from the Edit Default Policies dialog is used for the distribution.

To set up an action you have to do the following:

- Fill in a name for the action.
- If you want to copy a file, enter the path name of the source in the Source field. You can select the Files button to see the available disks.

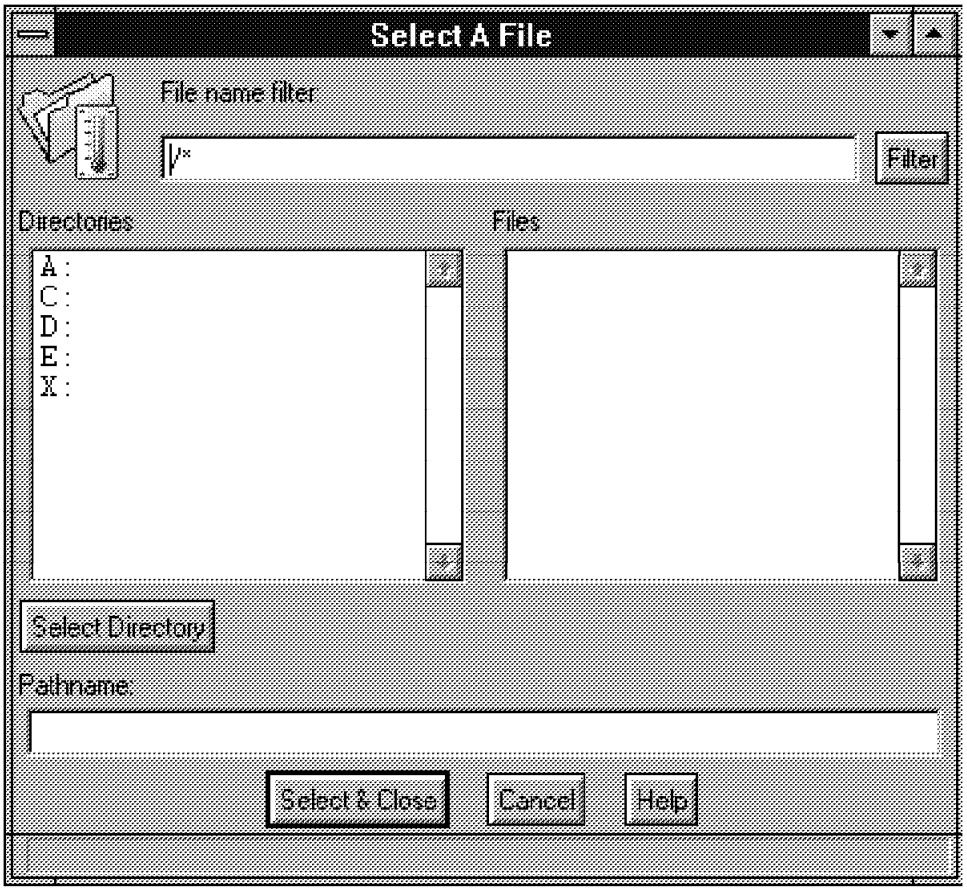


Figure 225. File Dialog

- If you want to copy the file to a special host, use the input field On host in the Distribution Actions window to specify one. The Hosts button leads you to the following selection window.

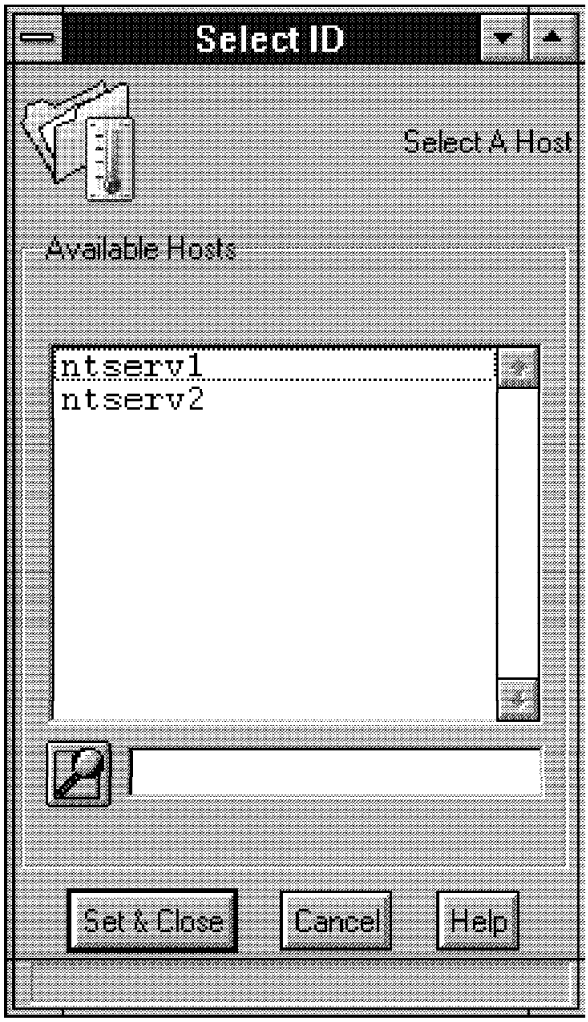


Figure 226. Selection Window

- Use the Copy to field to specify the destination path for a copy. Again the **Files** button can help you to find the right place.
- Put in the path and name of program in the Run program field to run a script or other executable file. The Files button can help you with this task.

To finish, select the **Set & Close** button.

There are some other actions you can do with profiles and monitors that are worth knowing about:

- You can define a Default Source Profile for a profile.

  This means that when you create a new monitor it is filled in with the default entries of the source profile. This is true if you select the **Add with defaults** button. When you go to this window, it is a little misleading as to which is the default profile. The top item is always highlighted, but the text near the bottom of the window indicates what the current default source profile actually is.

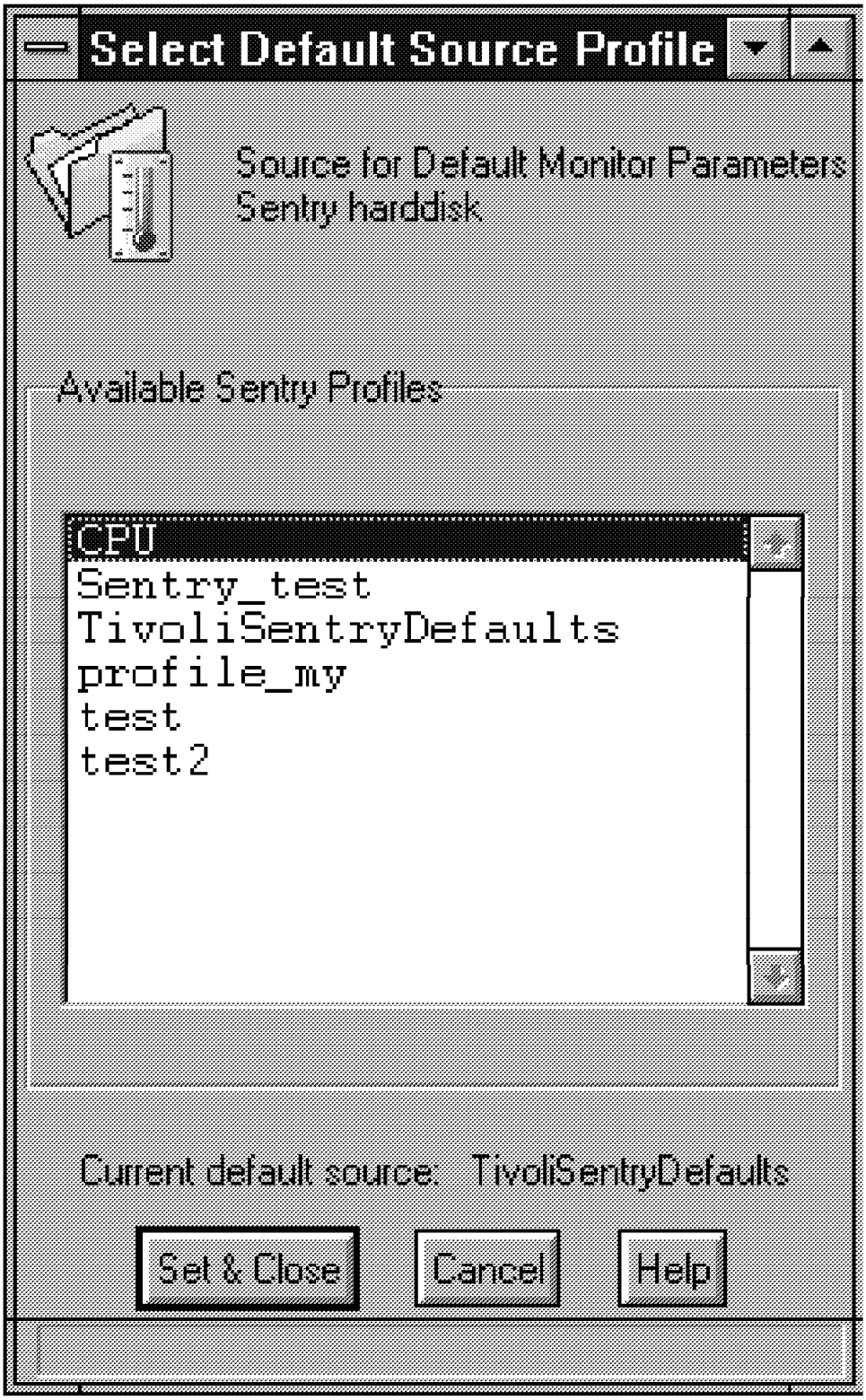


Figure 227. Default Source File

- Cloning is a method to create a new profile.

  Using this method, settings such as the distribution actions or the default policies are copied, but no monitors are copied. For more information refer to 7.1.4, “Profiles” on page 194.

- To delete a profile you have to choose the option pull-down **Profiles** and then **Delete** from the Edit menu in the Profile Manager window. Before you perform the action you have to select a profile to delete first.

Figure 228. Deleting a Profile

From the command line you can use the wdel command:

```
wdel label
```

For example:

```
wdel @SentryProfile:profile_my
```

- Locking/unlocking monitors enables an administrator to prevent a subscriber from editing or deleting a monitor. To lock or unlock a monitor, mark the monitor in the profile window and select **Unlock Selected monitors** or **Lock selected monitors** from the Edit menu.

Note

Another important function in the Profile menu is the option Go To Profile At. This enables you to look at profile copies to subscribers.

Figure 229. Go to Profile

You can select a subscriber from the list and the select the **Go To & View** or the **Go To & Edit** button. The first one enables you only to see the profile. The second one allows you to edit the profile if your access rights permit that.

This is the way to use profiles and monitors. The functions can be different between different Tivoli applications, but Sentry is a good example to learn about the basic functions. We will show how to work with Indicator Collections and how to set up proxy endpoints.

7.1.6 Sentry Basics

The installation for Sentry is described in Chapter 2, “TMR and Client Installation” on page 11.

Tivoli/Sentry monitors the status of network resources. To get information about the systems and to perform actions on remote systems, Sentry uses profiles. Sentry has two main components to perform these functions:

- The Sentry engine must be installed on each client to monitor its status directly. Another possibility is to monitor the resources through a proxy endpoint.
- The monitoring collections contain code that defines how data from a resource is collected.

Sentry provides three main functions:

1. Remote Performance Capabilities allows an administrator to distribute tasks to a remote system and to send messages back before a problem effects the whole system.
2. Centralized Configuration Mechanism can monitor specified resources of systems and you can define messages for changing one of these components.
3. Data Collection/Automatic Response can collect data about the whole system and define actions to perform based on this data.

The Sentry profiles are empty when they are created as shown in Figure 230.

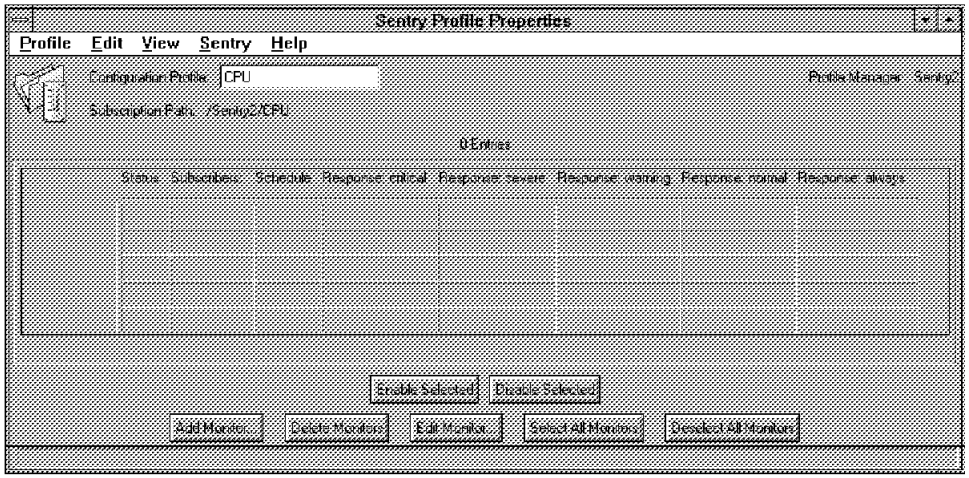


Figure 230. Initial Sentry Profile Window

By default, no monitors or responses are defined. It is your job to define them. We will explain the main functions in the Profile window before we show the details of how to define the monitors later on in this chapter.

- Add Monitor - This is the first action to take if there are no monitors defined. You can choose monitoring resources in a separate window.
- Delete Monitor - This deletes a monitor that has been previously defined.
- Edit Monitor - In a separate dialog you can edit the properties of a monitor that has already been defined.
- Enable Selected - When a monitor is active (enabled) you can monitor the activities in the TMR. The default for a monitor is to be enabled.
- Disable Selected - This leaves the monitor definition in the database, but it disables the specific monitor. For example, you may set up a monitor to monitor a MIB variable that required lots of research. Instead of deleting the monitor when you don't wish to capture any more data, you can disable it, and then turn it on at a future time.

7.1.7 Setting Up Sentry

After the installation of Sentry you must assign the Sentry Managed resources to the Policy Region. You can do that through the context menu of the Policy Region icon or through the desktop in the Policy Region.

In the first case you must click with the right mouse button on the icon. The context menu opens. Select **Managed Resources** from the menu and the following window will appear:

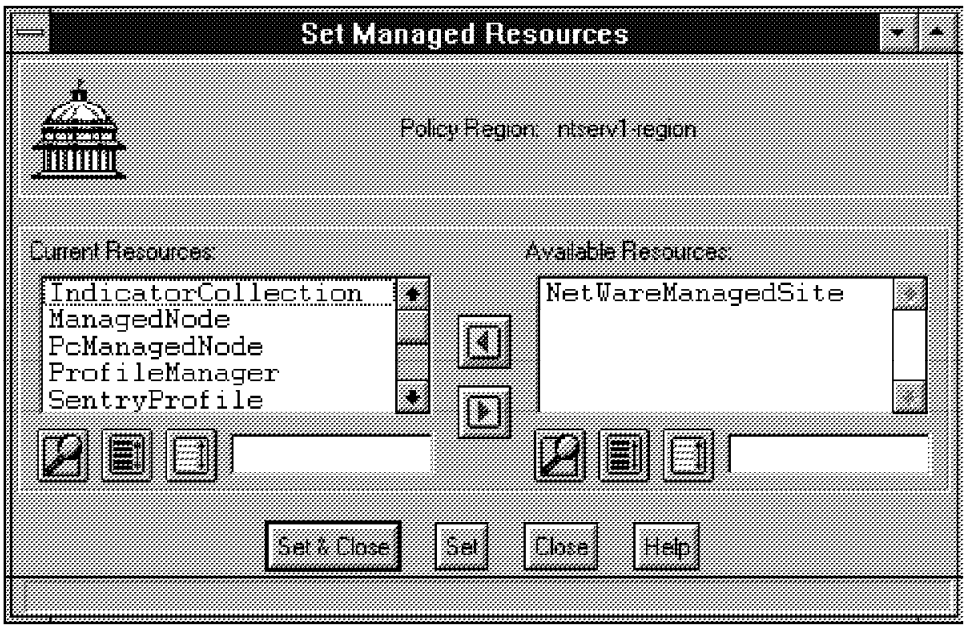


Figure 231. Profile Window

In the right list box you can see the available resources. The Sentry installation added three new resources to the list:

- IndicatorCollection
- SentryProfile
- SentryProxy

To perform the add action you need the senior role. You must add the three components to the current resources. There is a way to do this from the command line. You have to use the wsetpr command for this action:

```
wsetpr SentryProfile Region
```

For example:

```
wsetpr SentryProfile @ntserv1-region
```

After this you can start to define a new Profile Manager and to populate the new Profile Manager with profiles and subscribers as indicated above.

7.1.7.1 Indicator Collection

You can also create an Indicator Collection. You can create an Indicator Collection for each Policy Region. An Indicator Collection is a collection to view alarms and Sentry icons in an efficient way. You have to associate the profiles you want to monitor with the Indicator Collection. You can see:

- Log information for each Sentry policy in the region.
- Save a log file or send an E-Mail.
- Reset icons from triggered to non-alarmed.

To create an Indicator Collection go to the Policy Region main window and choose the option **Create** from the menu bar. There is an option called Indicator Collection under the Create menu item.

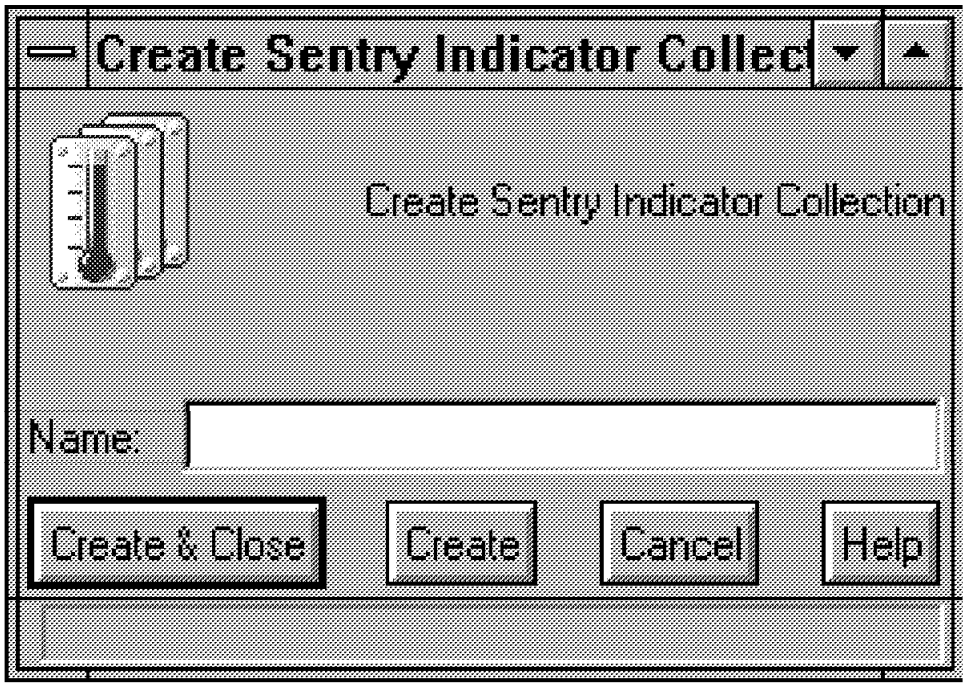


Figure 232. Indicator Collection Defining Window

In this window you have to label the new Indicator Collection. After you have finished your input and the **Create & Close** button is chosen, a new icon is added to the Policy Region.

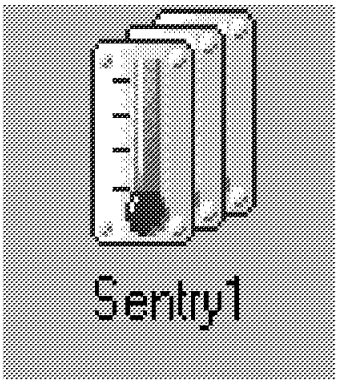


Figure 233. Indicator Collection Defining Window

This action can also be performed from the command line. The command is the wcrtsntcoll command. The format of the command is:

```
wcrtsntcoll policy-region name
```

For example:

```
wcrtsntcoll ntserv1-region Sentry1
```

The Indicator Collection is still empty. When you open the collection, an empty window appears. As mentioned above, you have to associate Sentry profiles with the Indicator Collection to activate it. Only after the association with some Sentry profiles can Indicator Collection display the profiles response. To perform this action you must have the role admin.

To associate a profile you have to open the Profile Manager and the profile you want to associate to the Indicator Collection. At this point you can see one of the advantages of the Indicator Collection: you can add/associate different profiles from different Profile Managers in your TMR. You don't have to open several Profile Managers if an alarm is generated.

From the Sentry Profile Properties window open the Sentry menu and choose **Select Indicator Collection**. A window with the available Indicator Collection opens. Two initial collections are available when you first install the product. Others will get added as you create connections to other regions. The initial two are:

- DefIndCol - Standard for Sentry
- (none) - No collection

The collection you added before is also available. Choose the one you created or another if one still exists. Select the **Set & Close** button to close.

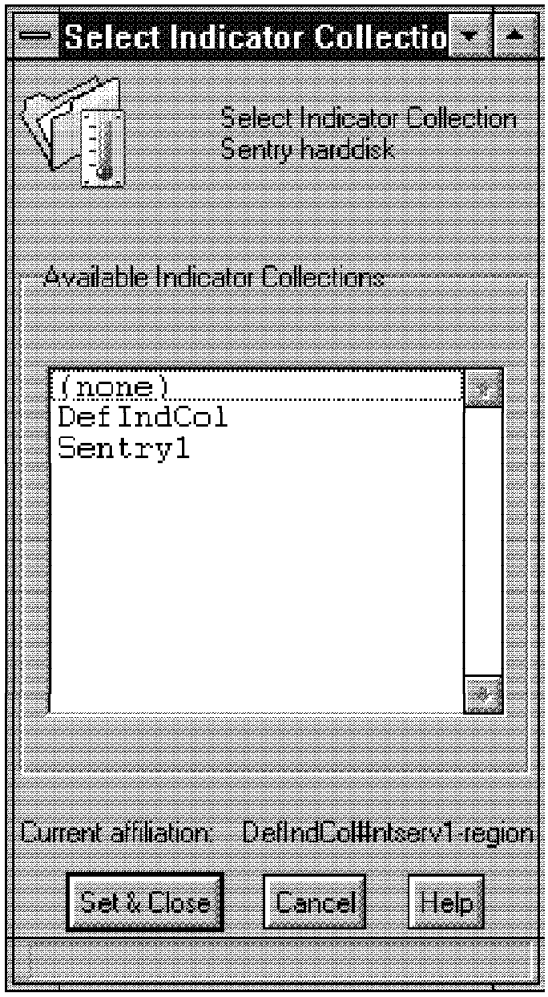


Figure 234. Indicator Collection Defining Window

If you now look at the Indicator Collection in the Policy Region there is an entry for the profile. You can click on **Refresh** from the View pull-down menu.

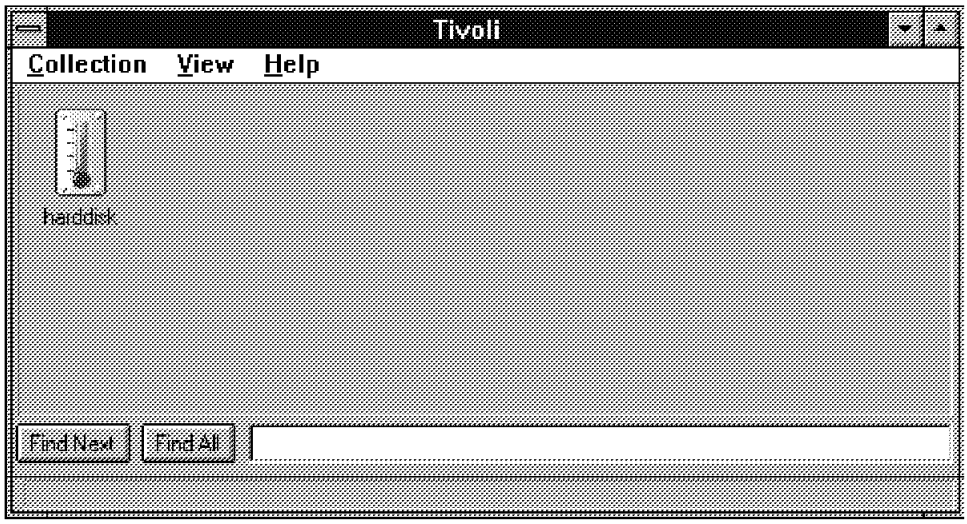


Figure 235. Indicator Collection Defining Window

When you open the collection for the profile you can see the log for the profile.

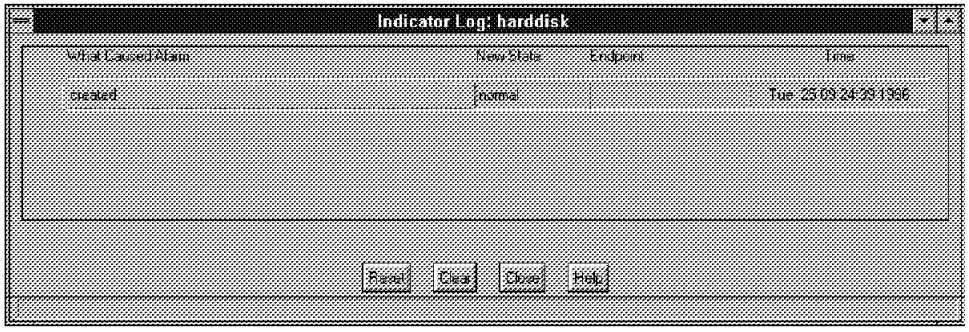


Figure 236. Indicator Collection Defining Window

There is also a command to set up the association from the command line, wsetcoll. The command looks like:

```
wsetcoll collection profile_name
```

For example:

```
wsetcoll Sentry1 harddisk
```

If you have defined monitors and you have distributed them, the indicators will fill with data. First the message that you set in the profile appears.

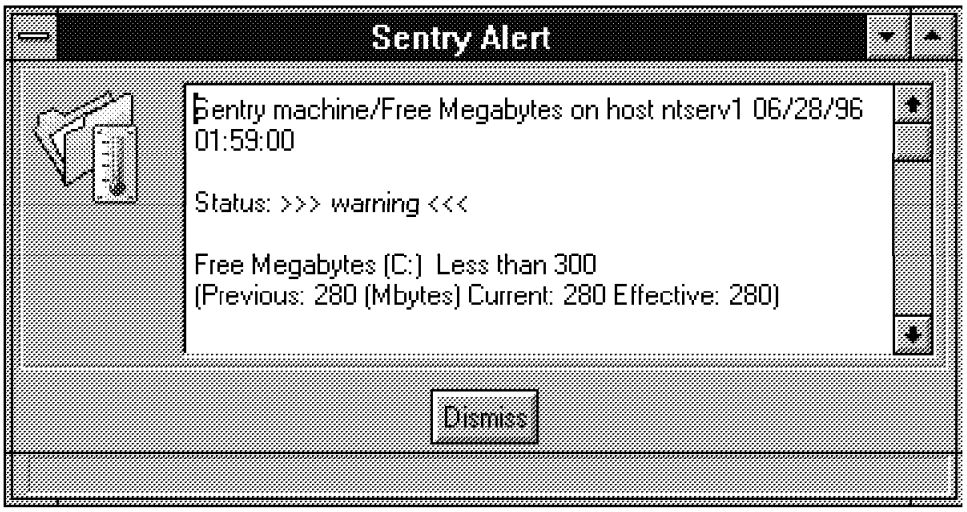


Figure 237. Message Pop-Up Set in the Profile Properties

Then you can open the Indicator Collection, and have a look at the different thermometer indicators.

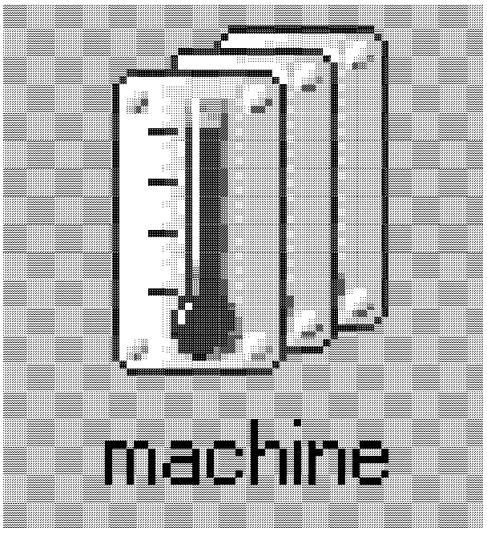


Figure 238. Indicator Collection

In the collection you can have a look at the indicator log.

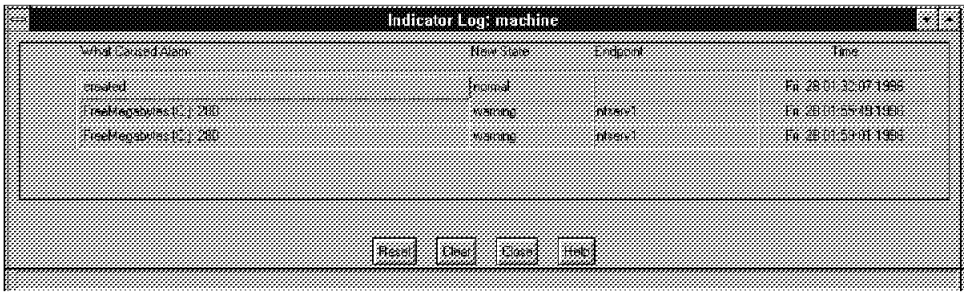


Figure 239. Indicator Log Files

7.1.7.2 Distribution Defaults

The distribution of the profiles is a function that needs to be done before they can start their monitoring function. You need to either be aware of the distribution defaults, or make some change to them to conform to your local policy. This action can only performed from the desktop. To set the Distribution Defaults open the profile you want to set. From the menu bar choose **Profile** and then **Distribution Defaults**. A window with the following options appears:

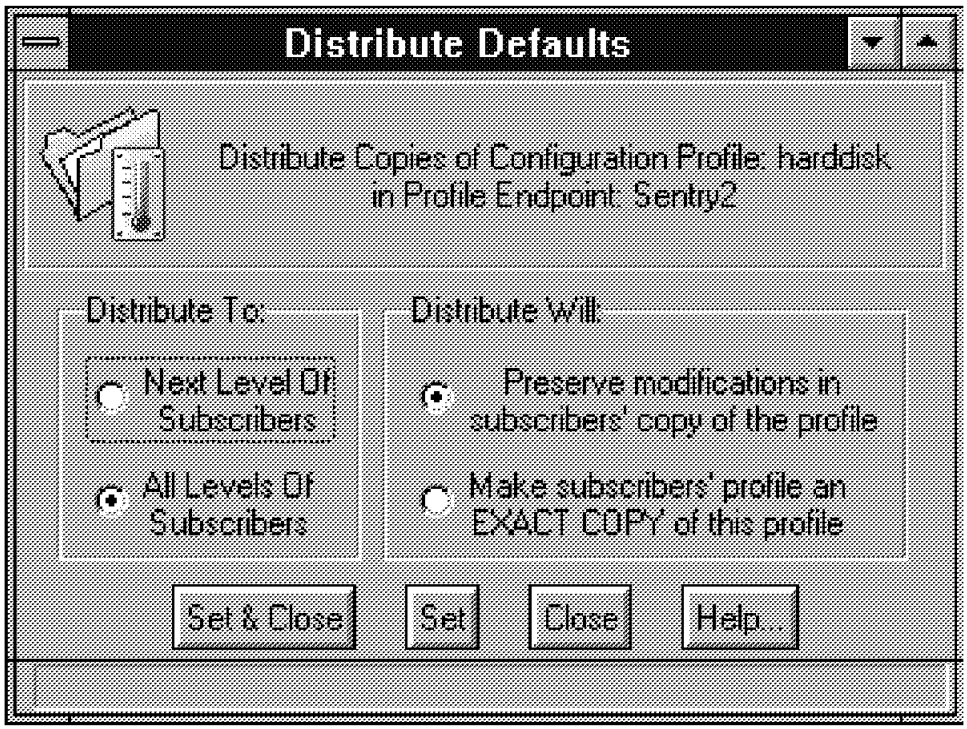


Figure 240. Distribution Defaults Window

The description we made for the setup of Sentry shows how to handle a new application. If you install another application for example, Courier, new profiles will be added to the TME. Sentry is a good example because it effects the configuration management. We next worked with the profiles monitors.

7.1.8 Monitors

Each profile contains one or more monitors. These monitors are predefined. The monitors are installed separately from the Sentry engine and are often called records. A record represents a monitor with all its settings and its response actions. To get a better understanding of this part, we show you a profile with several different monitors.

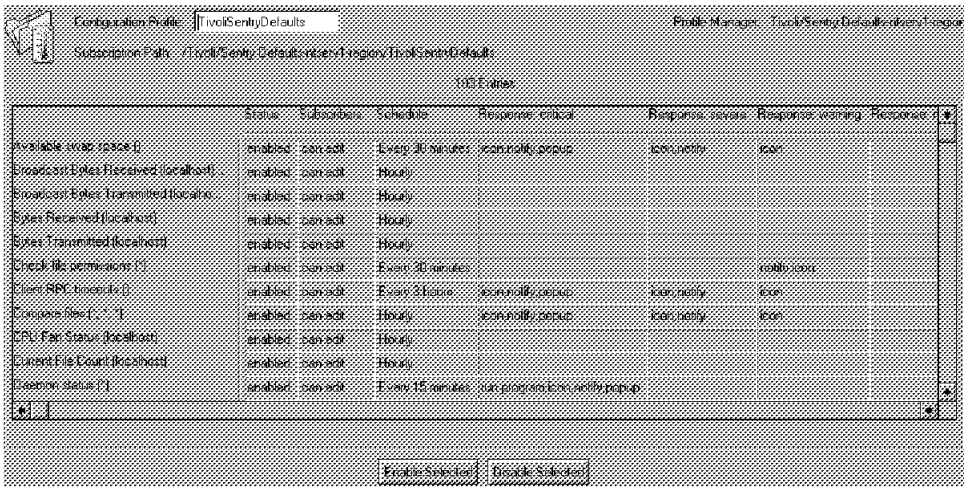


Figure 241. Profile with Monitors

The first step is adding a monitor to a newly defined profile.

7.1.8.1 Adding a Monitor

A monitor, or record, contains information that is needed to monitor a system or network resource. The record contains information about:

- What resource should be monitored
- How often it should be monitored
- Actions to taken when the conditions are met

Some of the monitors delivered with the monitor collections have predefined values. This minimizes the workload. To work with profiles and monitors you need the admin role.

To add a monitor to a profile you must open the profile you want to work with. In the profile you can select the **Add Monitor** button or the **Edit** menu and select **Add monitor** from the menu. The result is the same.

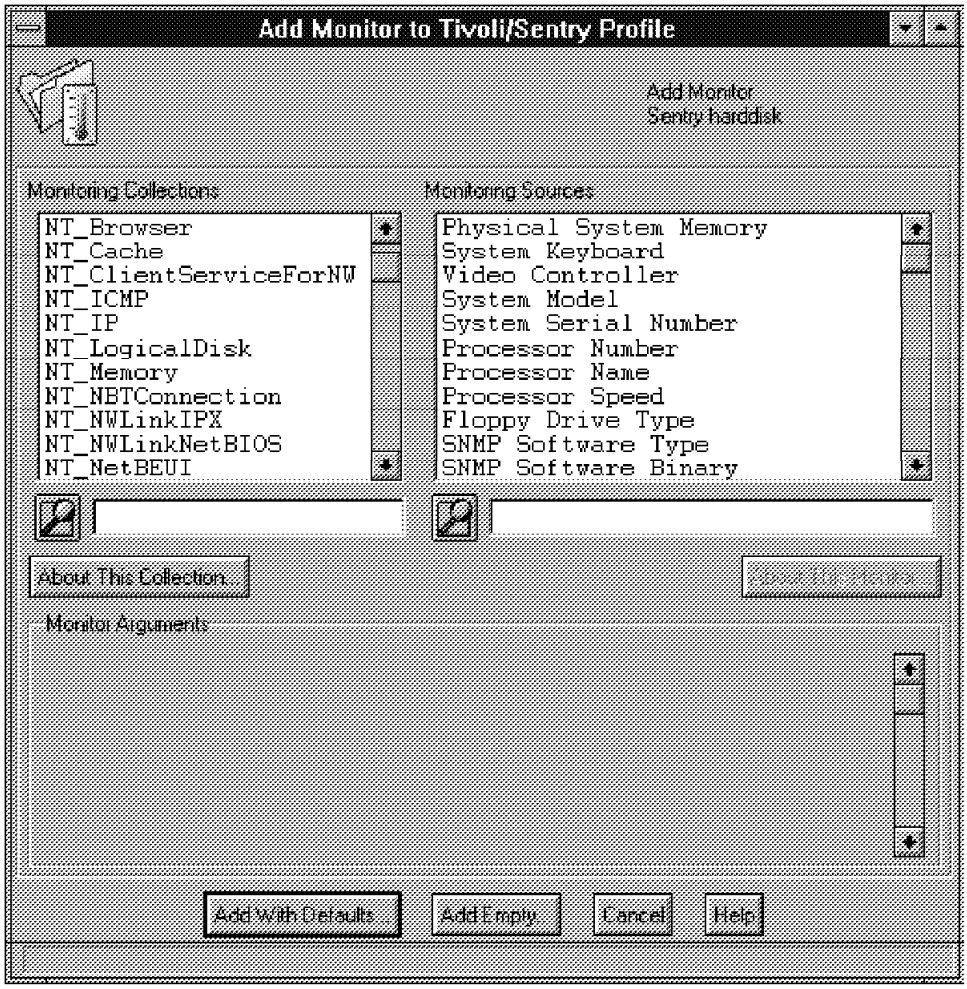


Figure 242. Add Monitor Window

In Figure 242 you can see monitoring collections in the left list box and monitoring sources in the right list box. Each monitoring collection contains different monitoring sources. If you need an explanation, there is a button under each list box called About This Collection or About This Monitor. The first thing you have to do is select a monitor from the left. The sources belonging to this monitor are shown in the right list box. You then need to select a monitoring source.

After you made your choice you can click on the **Add with defaults** button or the **Add Empty** button. If you select the Add with Empty button, the Edit Sentry Monitor dialog will be displayed. The Add with Default button uses the default settings. If there is no default an alert is displayed.

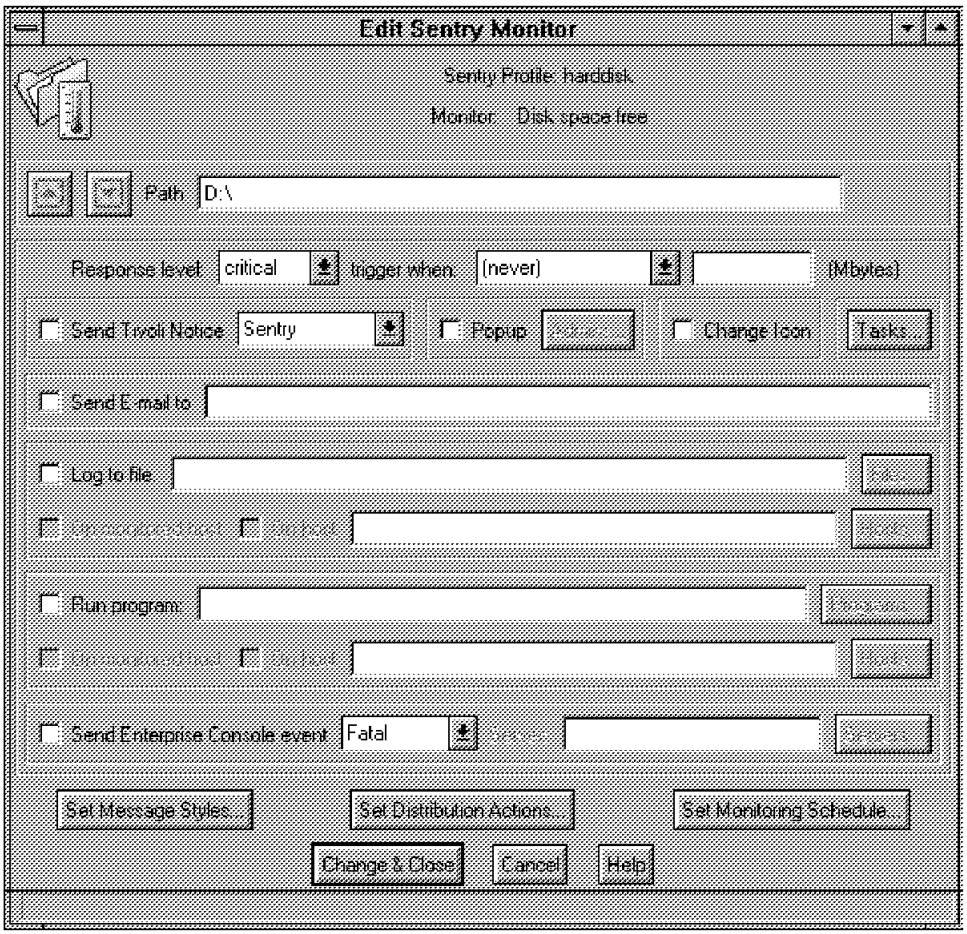


Figure 243. Path Input Field in Monitor Window

A few resources might have arguments that appear in the window. You have to fill in the arguments. In our example the path for the disk space monitoring to begin appeared. You can only add one monitor at a time. To add more you have to repeat the dialog.

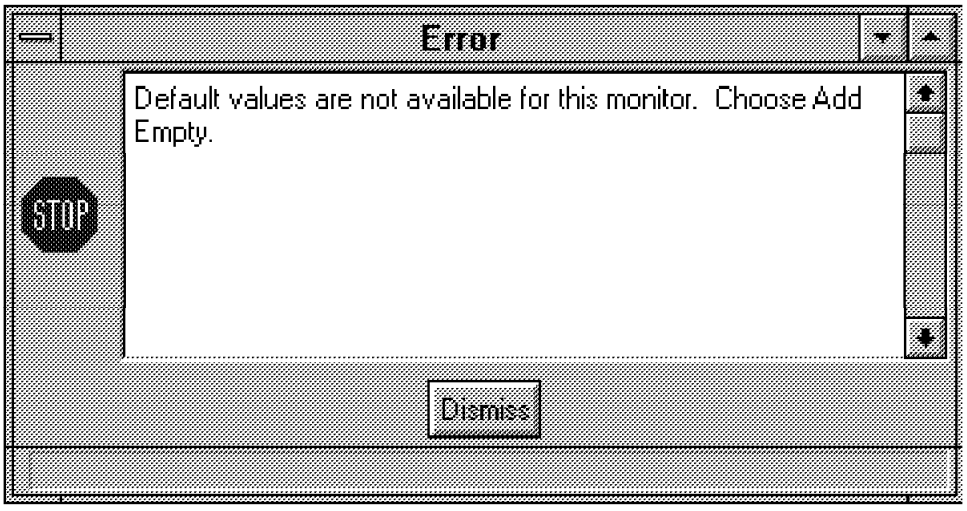


Figure 244. Alert Display

The creation of a monitor is even possible through the command line interface. The command to use is waddmon.

```
waddmon monitor-collection monitor (-a argument) (-t timing)
(-d/+d) response specifier profile-name
```

The parameters are:

- monitor-collection: Specifies the monitor-collection to add.
- monitor: Specifies the name of the monitor.
- -a argument: An argument must be added to the monitor.
- -t timing: Specifies the sampling frequency.
- -d/+d: Enables or disables the monitor.
- response-specifier: Several arguments for the response.
- profile-name: Name of the Sentry profile.

For example:

```
waddmon Universal diskused -a C: -t "60 minutes"  -c severe -R ">" 500
```

This adds a monitor to check the disk space used, to the profile called *hard disk*. It will check the metric every 60 minutes and it will send a message indicating its severity if the disk space used is greater than 500 MB. The monitored disk is C:.

The available monitoring sources for the Universal collection are the following:

- appStatus - Application status
- appInstances - Application instances
- swapavail - Swap space available
- pageouts - Page-outs
- loadavg - Load average
- host - Host availability
- oserv - Remote oserv status
- diskavail - Disk space free
- diskused - Disk space used
- diskusedpct - Disk space used in percentage
- filesize - File size
- filechk - File checksum
- filediff - File compare
- fileperm - File permission
- countstr - File pattern matches
- nasync - Asynchronous numeric
- sasync - Asynchronous string
- ncustom - Numeric script
- scustom - String script

Each collection has its own monitors described by a keyword. Therefore, refer to the monitor collection documentation of each of of the collections to more detailed information.

Note

You can only add a monitor from the command line if the monitor window from the desktop is closed, otherwise it is locked.

7.1.8.2 Edit Monitor Dialog

This dialog sets the configuration for how often a resource will be monitored and which response will be taken. The Edit dialog is only used for existing monitors or records. You can get to the dialog by selecting the **Edit Monitor** button on the Sentry Profile Properties window. The following dialog is displayed.

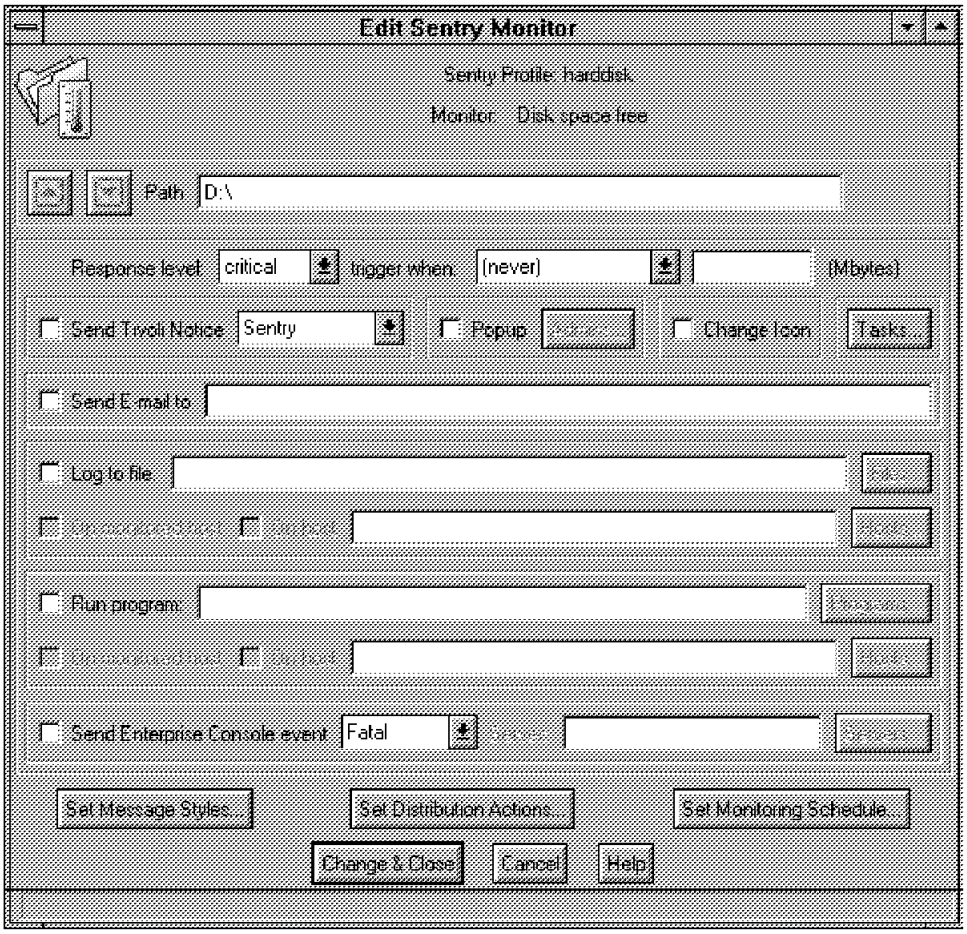


Figure 245. Edit Monitor Dialog

Each edit dialog is divided into several sections. We only set some of the fields here. The specific dialog window is dependent upon what you are monitoring. The windows are basically the same, but they vary slightly depending upon what you are going to monitor. In the case of disk monitoring:

- The first row expects resource information. It is the information you filled in in the Add Monitor dialog.
- The second row is for threshold settings, response levels and trigger definitions.

- The rest of the input fields are for response actions.

Beneath the input fields you can find several buttons to use for more options for the monitor. To understand the definitions we show an example that refers to the hard disk profile above.

- The path statement is D:, because we want to know the free disk space for this drive.
- To set the response level you can open a pull-down menu. There are five levels available for this Sentry monitor:

  1. Critical
  2. Severe
  3. Warning
  4. Normal
  5. Always

  For other applications there can be more levels. For our example, we chose the severe level. If you want you can define thresholds and actions for every response level. We recommend that you do that. Every new response level asks for new input for all fields below the second row. The result from your tailing the monitor will look like Figure 246.

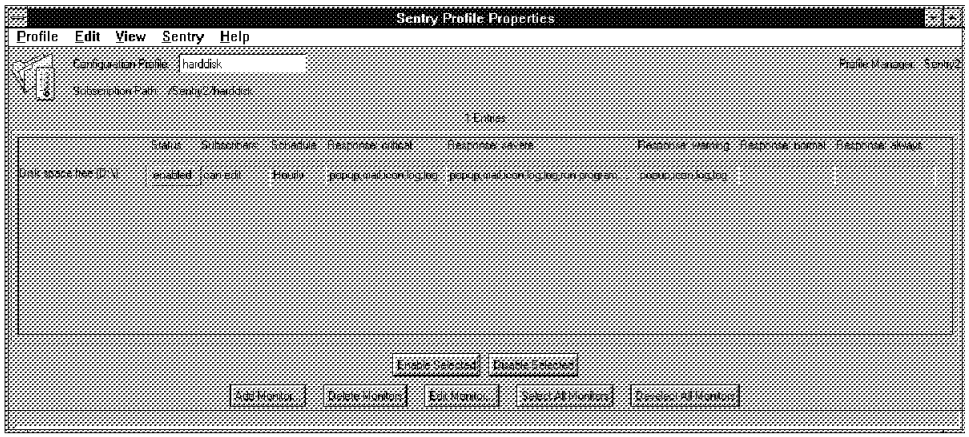


Figure 246. Monitor with Three Levels

This monitor has three levels associated with actions and two without.

- The next field is the field for the trigger options. You can specify when a message or action should be performed. For monitoring disk space the important value is Megabyte. So you can set a threshold for less than or greater than any amount of MB. The options that are available depend on the monitor, the source and the application. For our example we chose to take an action if the amount was greater then 250 MB.
- In the next row you can choose the response type for when the threshold is reached.
- Send Tivoli Notice sends a message to a notification group. To receive a message you must have a subscription for the notice group.
- Popup sends a pop-up to the administrators of the TMR and the connected TMRs. To define the receiving administrators select the **Admins** button.

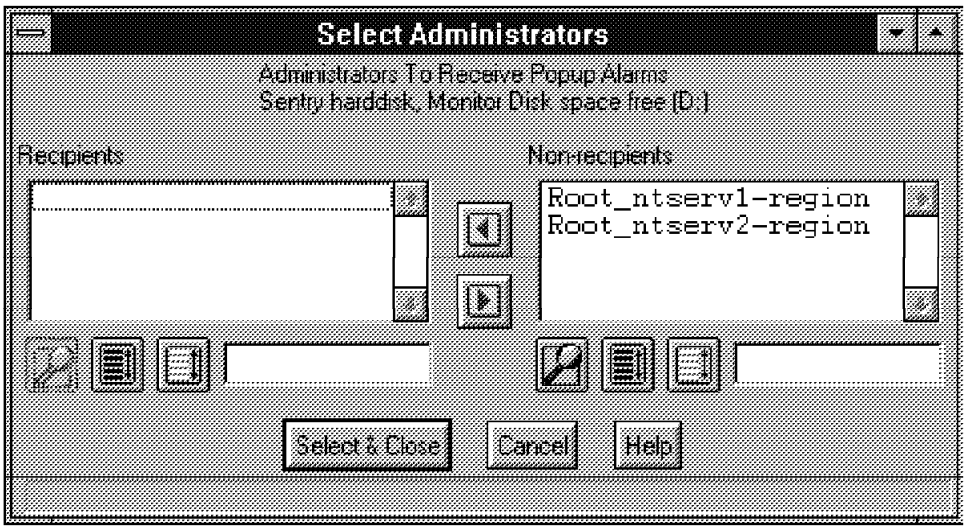


Figure 247. Edit Monitor Dialog Admin Pop-Up

In this window you can add available administrators to the recipients. We added all available administrators for this example.

- Change Icons allows icons in an Indicator Collection to change status. That means the arrow in the thermometer starts moving when a threshold is reached. We enable this function for our example.
- Tasks runs specific tasks when a threshold is reached. You can select task libraries in the left list box and specific tasks from the right list box.

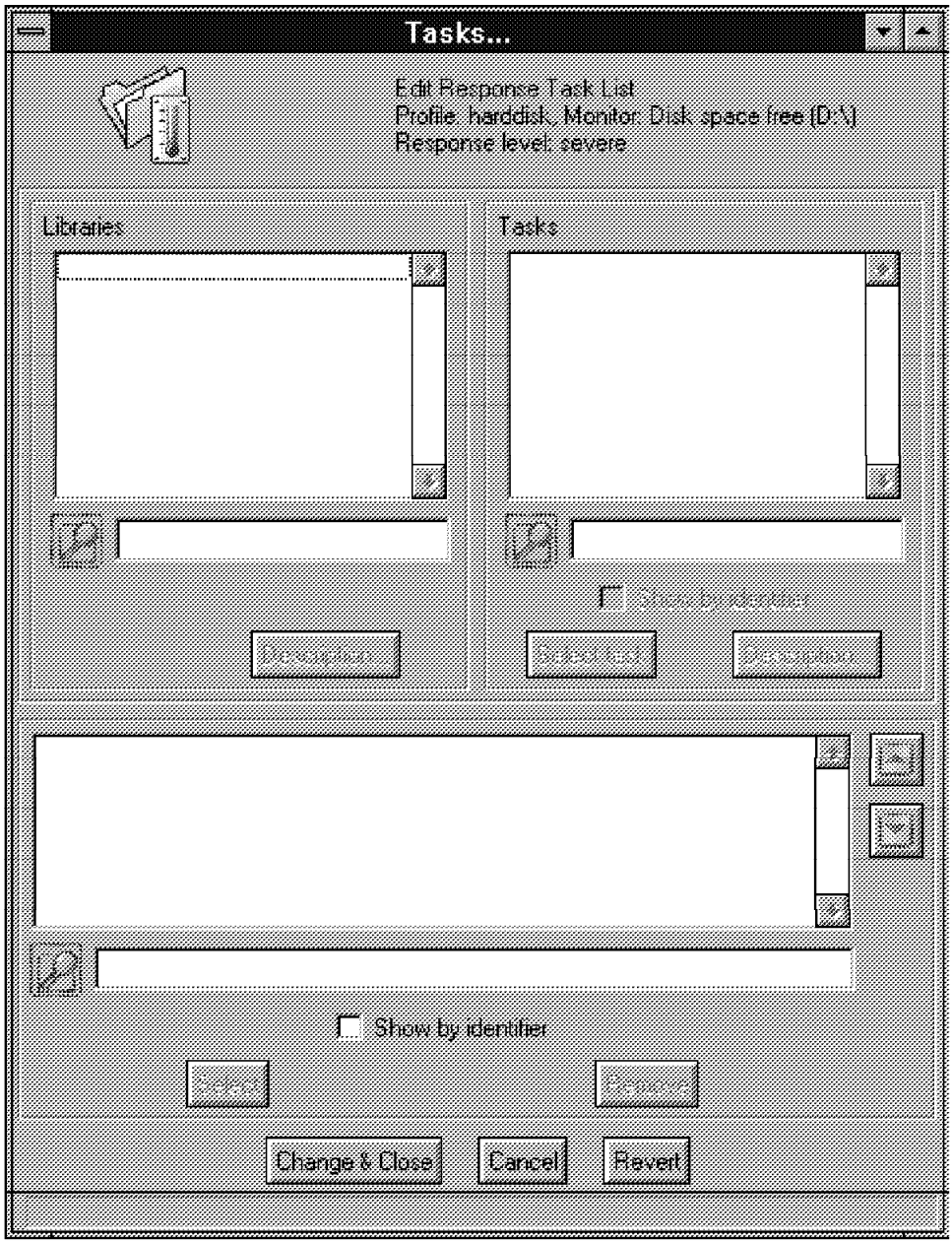


Figure 248. Edit Monitor Dialog..Task Window

- Send E-Mail to enables you to send an E-mail to a specified address or host. This is a very important interface when you are on the WWW interface. In our example we sent a file to deibmav4@ibmmail.com. Of course, you must have an E-mail application installed on that system. You also have the option of clicking on the **Files** option to fill in the field.
- Log to file adds an entry to a specified log file. You have to enter the file name in the blank field.

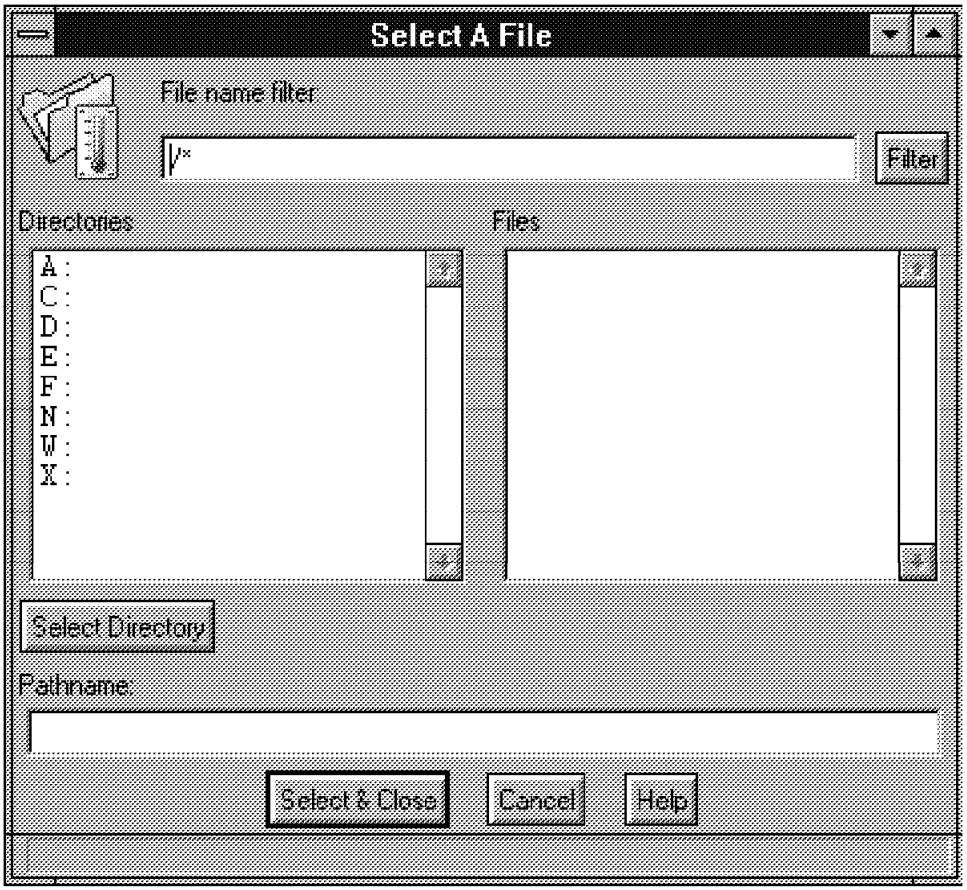


Figure 249. Edit Monitor Dialog Files Window

Additionally, you can choose if the log file should be placed on the host it occurs on, the monitored host, or if it should be routed to a specific host. If you enable the **On host** option, you can select a host from a dialog by selecting the **Hosts** button.

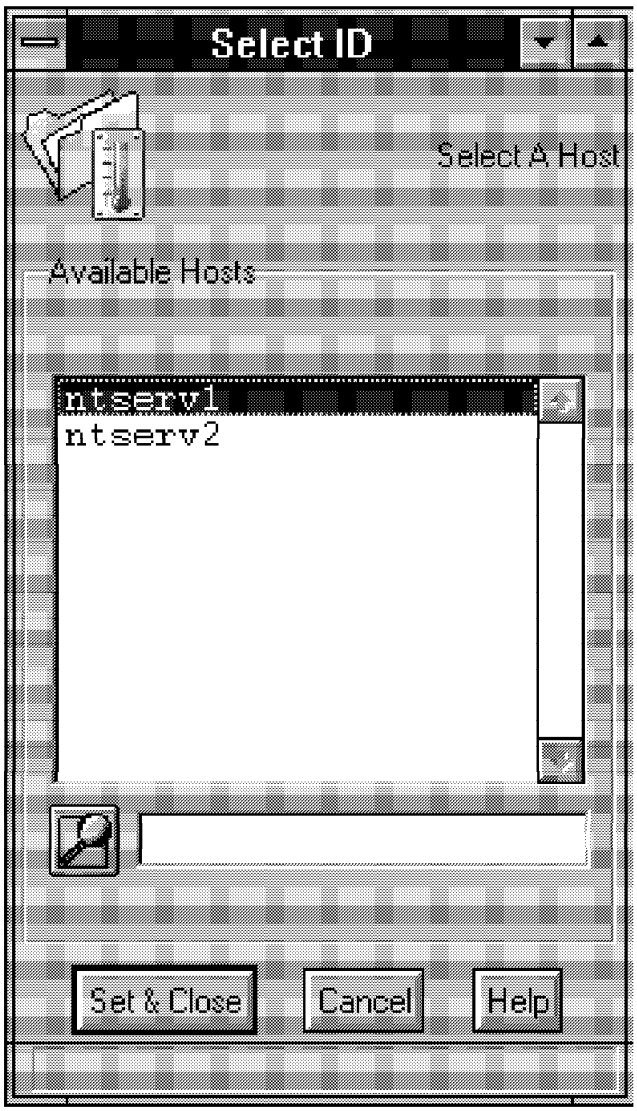


Figure 250. Edit Monitor Dialog Hosts Window

For our example we chose ntserv1 as the host. For the path we selected drive D: and the path /log.

- Run Program starts a specific program or executable file (script) when the threshold is reached. You have the choice to run the program on the monitored host or on another host. If you want to run a program you can set the path by selecting the Program button or you can enter it manually. If you click on the **Programs** button, you will get an input dialog similar to the log files window. In our case we want to start the notepad.exe from the ntserv1 host.
- Send Enterprise Console event sends Sentry data to the Tivoli/Enterprise Console. If you enable this option you must add a host, where the console resides. In our example, we didn′t send any data to the console.
- Set Message Style opens a window where you can set the form of the messages sent by Sentry.

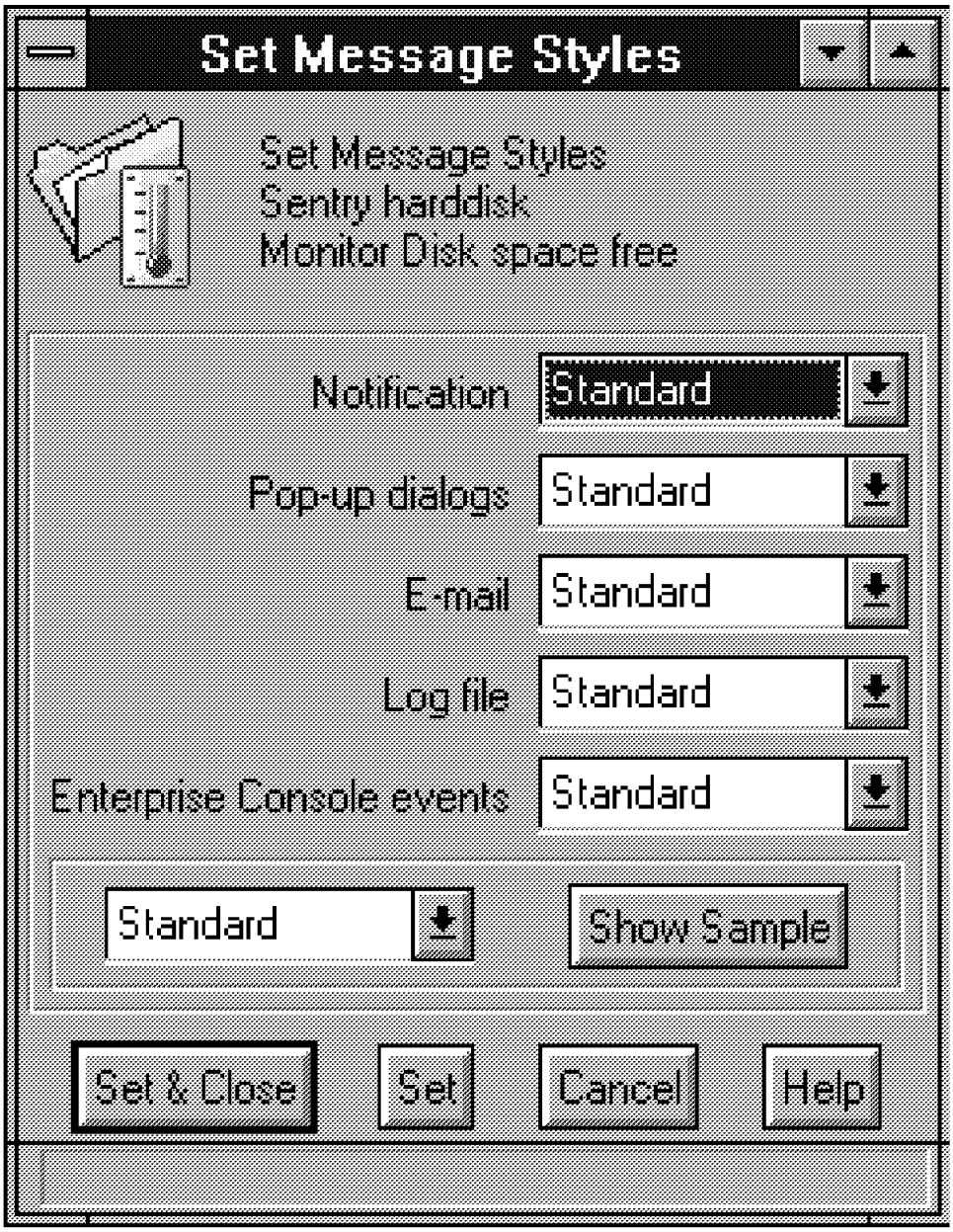


Figure 251. Edit Monitor Dialog - Set Message Style Window

The default is the Standard format. The following forms are available:

- Standard
- Brief
- Long
- Local Format 1
- Local Format 2

To get an understanding for the forms, there is a Show Sample button that can be selected. If you select one form from the pull-down and then select the button, the program will show you how the message text will look. We used the defaults for our example.

- Setting Monitoring Schedule allows you to set parameters for how often a monitor will check the monitored resources. You can set the schedule for the profile by selecting **Edit** and **Set Default Schedule** for each monitor in the Edit dialog.

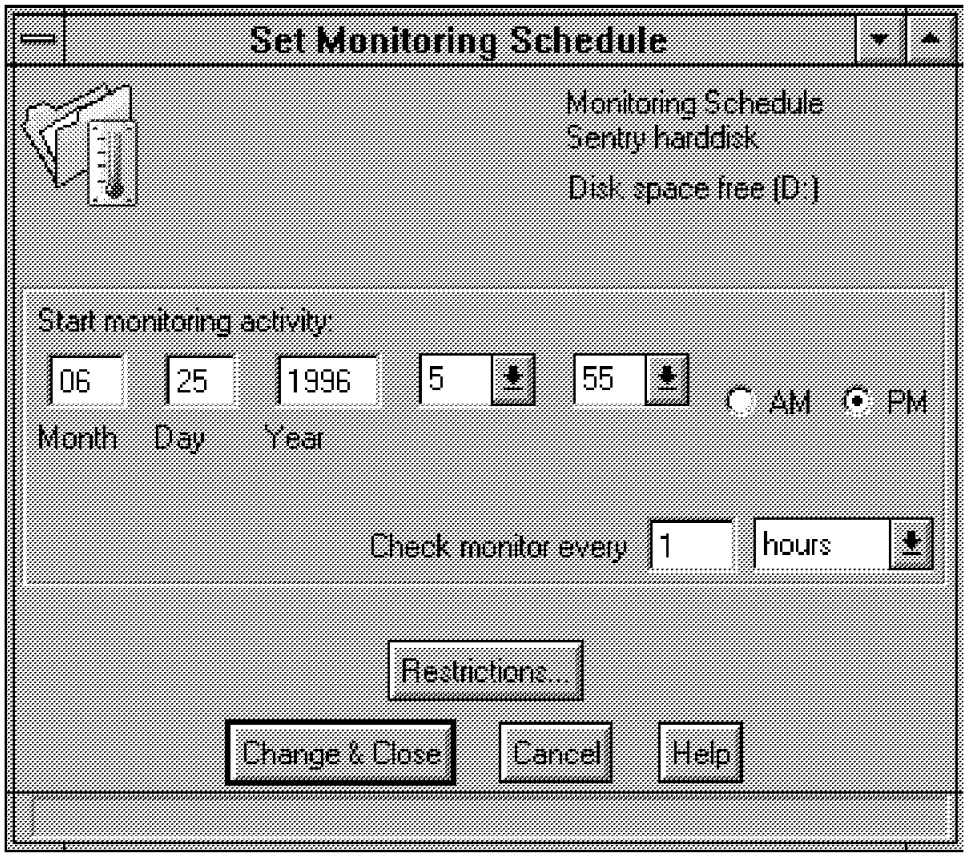


Figure 252. Edit Monitor Dialog - Set Monitoring Schedule

You can monitor things from one minute to 9999 months. The first thing you have to set is the month, day and the year. This is the start time for the schedule. Then select the hour and the minute from the pull-downs followed by AM or PM. To enable some time restrictions for the monitor click on the **Restrictions** button. A new window opens.

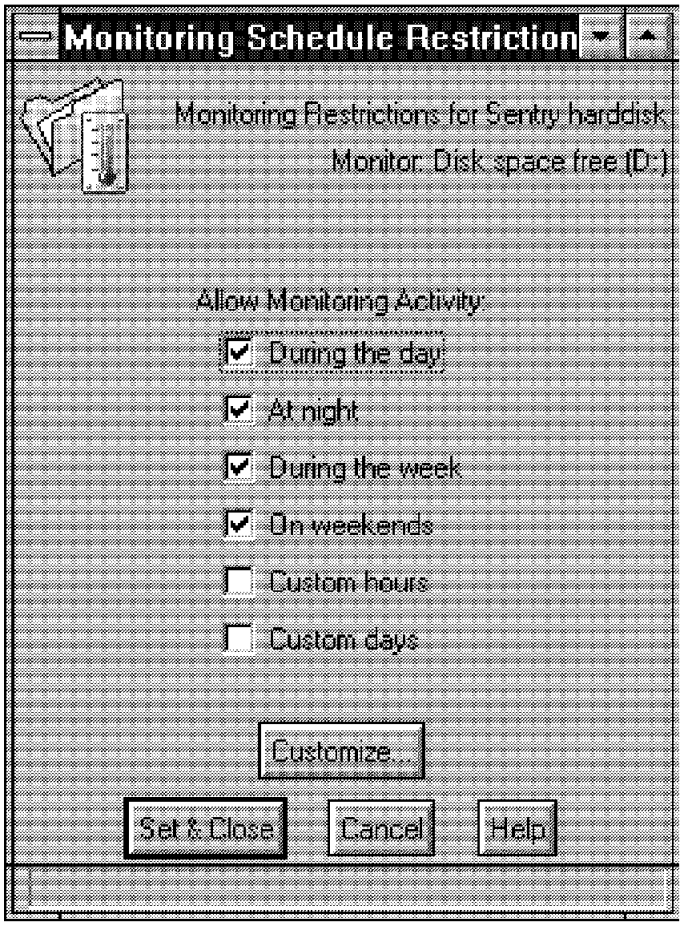


Figure 253. Edit Monitor Dialog - Set Monitoring Schedule Restrictions

In this window you can set the restrictions for when the monitor activity should occur:

- During the day
- At night
- During the week
- On weekends
- Custom hours
- Custom days

If you want to define your own times and days when the monitor activities should start, you have to choose **Custom hours** or **Custom days**. To define the time periods, click on the **Customize** button.

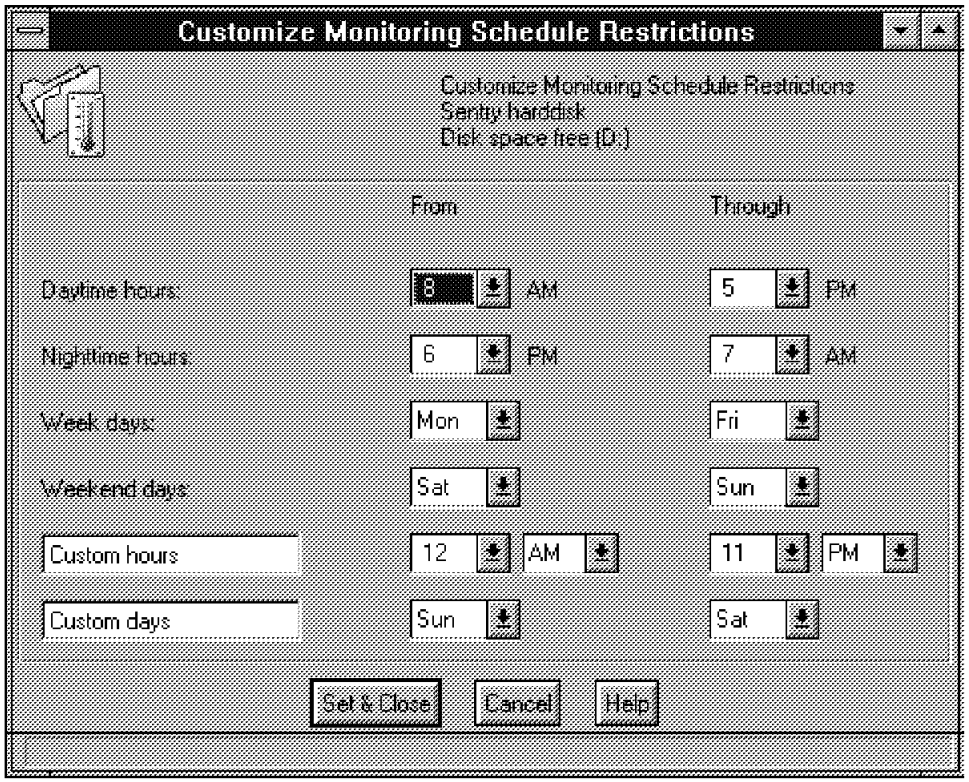


Figure 254. Edit Monitor Dialog - Customize Monitoring Schedule Restrictions

In this window you can define the time and the day when the activity should occur. In our example we used the default options. Click on the **Set & Close** button in each window to confirm your input and finish defining a monitor.

After you finish defining the monitor, a new monitor entry occurs in the profile.

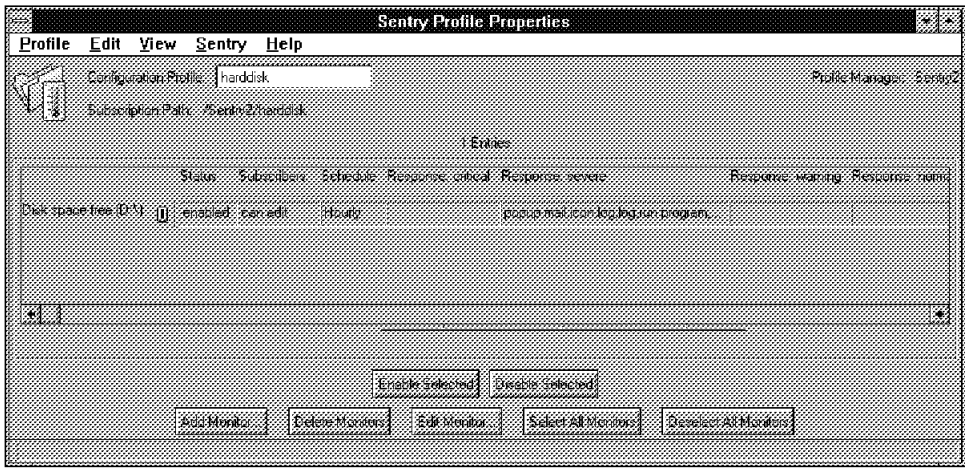


Figure 255. Profile with New Monitor

Note

The small rectangular box just to the left of the new entry indicates that you created a new monitor record but have not saved it. It will go away if you click on the **Save** option from the Profile pull-down menu.

Since it takes a while to set up a monitor, you might want to consider the following:

- Before you start defining the monitors make a plan of what is needed to be monitored.
- Define the monitors you will need.
- Think about your profile managers and your profiles. Define which system you will need to monitor. Perhaps you only need the monitor in one profile manager for a connected TMR.

7.1.9 Working with Monitors

There are several ways to work with monitors:

- Copying monitors between profiles
- Moving monitors between profiles
- Enable or disable monitors
- Adding and deleting monitors
- Finding/sorting monitors

7.1.9.1 Copying Monitors between Profiles

When you copy a monitor from one profile to another, Sentry makes an exact copy of the monitor, including its resources. To copy a monitor, go to the profile that includes the monitor you want to copy. Select the monitors you want to copy by using the mouse from the desktop of the profile. Then select **Edit** from the menu bar and the option **Copy monitors**.

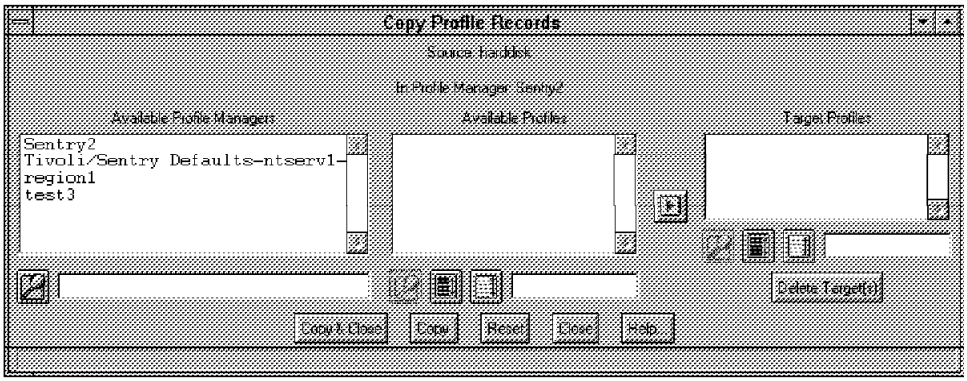


Figure 256. Copy Profiles

In this window you can select a profile manager from the left list box. After you have chosen one, the available profiles occur in the list box in the middle.

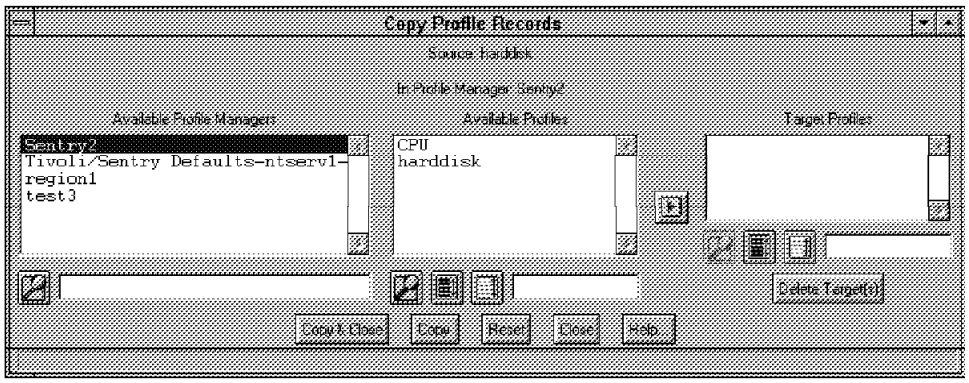


Figure 257. Copy Profiles

Select one of the available profiles and add it to the right list box with the button between the two list boxes. Then select the **Copy & Close** button to perform the copy.

Note: Be aware of the following:

- The copying process will not work if the target profile is still open and working. It will notify you of the error with a window like the following:

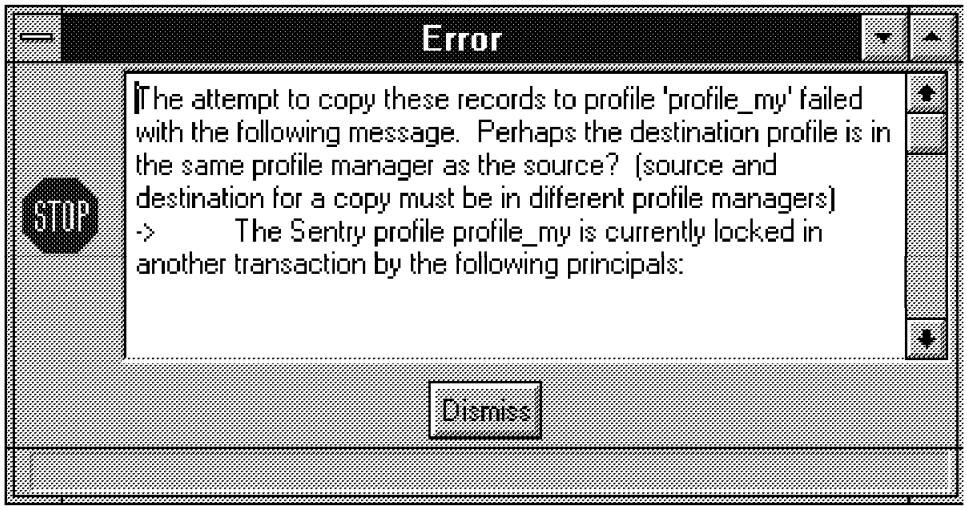


Figure 258. Error Message

- Close your profile to confirm the action. Then you can open the target profile. We detected some problems when you try to open the target profile at the same time that the source profile was open.

7.1.9.2 Moving Monitors between Profiles

The difference between moving and copying a monitor is that when you move a monitor the monitor will be deleted from the source profile and added to the target profile. To move a monitor you must first select one from the profile desktop. Then select **Edit** from the menu bar and choose **Move monitor**.

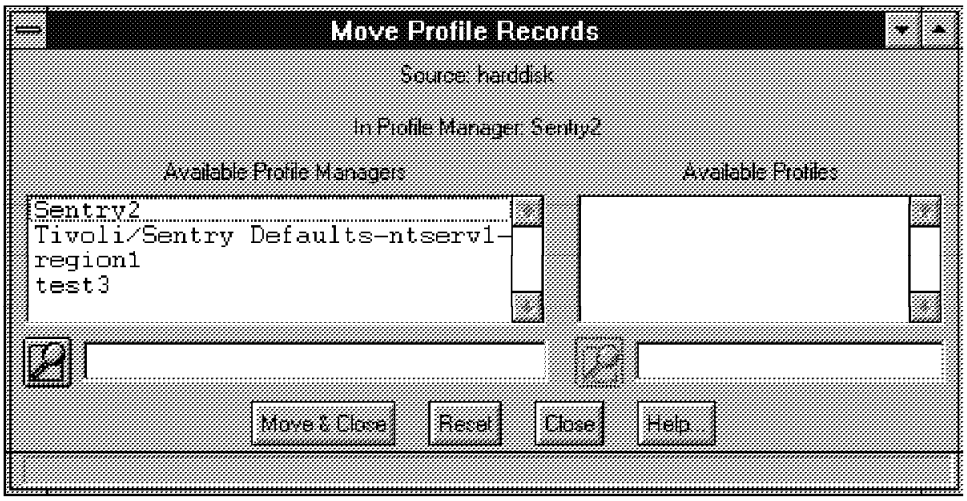


Figure 259. Moving Profiles

In this window you have to select a profile manager from the left list box as a target. Then click on the **Move & Close** button.

7.1.9.3 Enable/Disable Monitors

By default, all newly created monitors are enabled. You can disable a monitor for special situations. Disabling also means that no responses and no actions will be received and performed for the node or nodes the monitor is set for. To enable or disable a monitor you can use the desktop or the command line.

From the desktop click on the monitor you want to enable or disable. It is now highlighted. Then click on the **Enable Selected** or **Disable Selected** button to perform the action. To confirm select **Save** from the Profile menu in the Profile window.

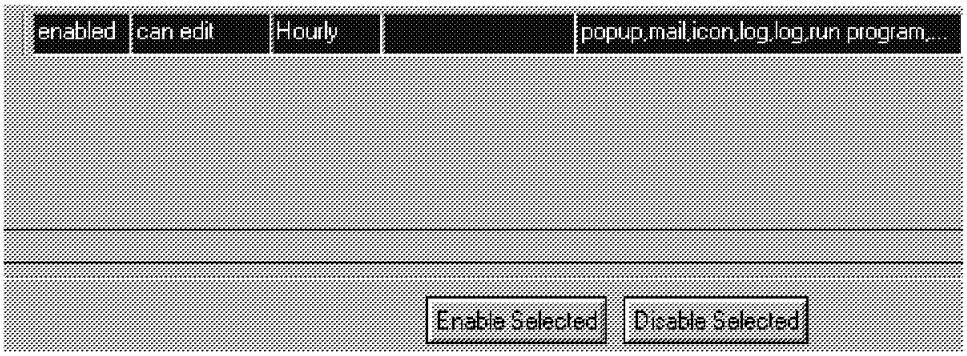


Figure 260. Enable/Disable

From the command line use the wsetmon command. The syntax is:

```
wsetmon + options key profile_name
```

The parameters are:

- + turns off the functioning of the options.
- key specifies a numeric key value.
- profile_name

The options are:

- -c level sets the response level. The level is warning, critical or severe.
- -t time specifies the frequency of the monitor sampling.
- -d disables a monitor.
- +d enables a monitor.

  When you choose -c there are some level specific options:

  - -e program enables a user-specified program to run. A fully qualified path is required: host: path name).
  - -f file enables file logging. The fully qualified path name is required.
  - -i enables icon switching.
  - -m address enables notification to a specified E-Mail address.
  - -n notice-group enables notification.
  - -T severity server enables Sentry to send events to an enterprise console. The severity can be fatal, critical, minor, warning, harmless or unknown.
  - -p list enables pop-up dialogs on administrator desktops.

For further information refer to the Sentry reference guide. For our purpose we only used the -d/+d parameter:

```
wsetmon +d 0 harddisk
```

This command enables the monitor with the number 0 in the hard disk profile. To get the number for the monitors use the wlsmon command:

```
C:\winnt35.0>wlsmon harddisk
0  Monitor: Disk space free(D:\)
  Timing:Hourly
  Responses:
    [critical]
    [severe]
    when probe result  > 250
      popup(Root_ntserv2-region,Root_ntserv1-region),mail(deibmav4@ibmmail.com),icon
(Sentry1),log(/D:/log),log(ntserv1./D:/log),run program(D:\winnt35.0\notepad.exe),ru
n program(ntserv1.D:\winnt35.0\notepad.exe)
    [warning]
    [normal]
    [always]
```

Figure 261. wlsmon Output

In the first line you can see the number of the specified monitor.

7.1.9.4 Delete a Monitor

If you delete a monitor it is removed from the Sentry profile. But it is not fully deleted. Only when you use the **Save** option from the Profile menu is the monitor fully removed from the database. From the desktop select a monitor and then click on the **Delete Monitors** button. Or you can use the Delete Monitors option from the Edit menu.

Note

If you detect that you deleted the wrong monitor there is a way to get the old monitor back. Select the **Reset** option from the Profile menu. If you saved your last state of the profile it won't be a problem.

7.1.10 Indicator Collections

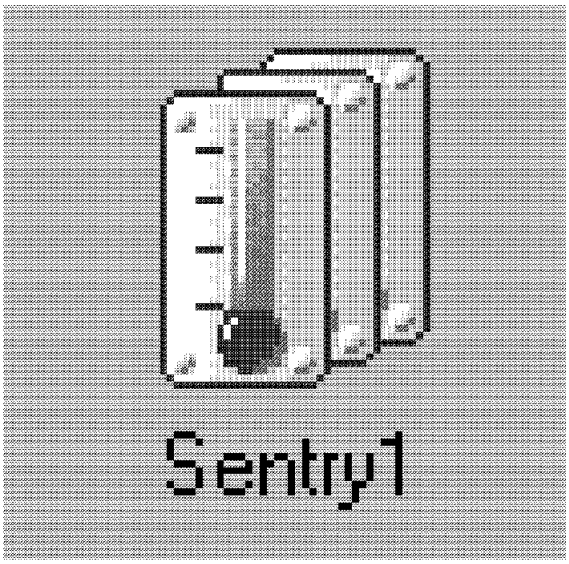


Figure 262. Indicator Collection Icon

Indicator Collections are a gage for different profiles. They will show the status of the associated profiles. Before you can work with Indicator Collections you have to:

- Create a Sentry profile
- Create an Indicator Collection
- Associate a Sentry profile with the Indicator Collection

In the beginning of this chapter we discussed Indicator Collections. When you open the Indicator Collection the window with the associated profiles appear.

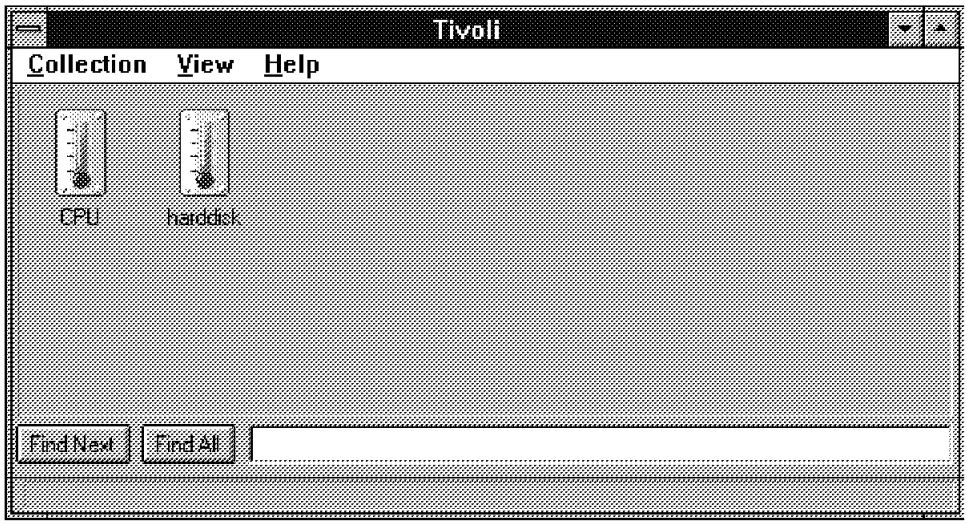


Figure 263. Indicator Collection Window

For each Indicator Collection based on a profile there is a context menu available. In this context menu you can:

- Open the indicator
- Save the indicator to a file
- Mail the indicator to a special person or system
- Reset the indicator
- Clear the indicator

7.1.11 Proxy Endpoints

Normally, the TME Managed Node is the endpoint of a distribution. But you can also have a *proxy endpoint*. A proxy endpoint is a network device, non-TME host, that subscribes to a profile. A proxy endpoint is useful when the value of the environmental variables vary. Each proxy endpoint object contains a list of environment variables that are passed to the Sentry engine. Proxy endpoints can also be used as filters for monitors. To create a proxy endpoint go to the Policy Region window and select **Sentry Proxy** from the Create menu.

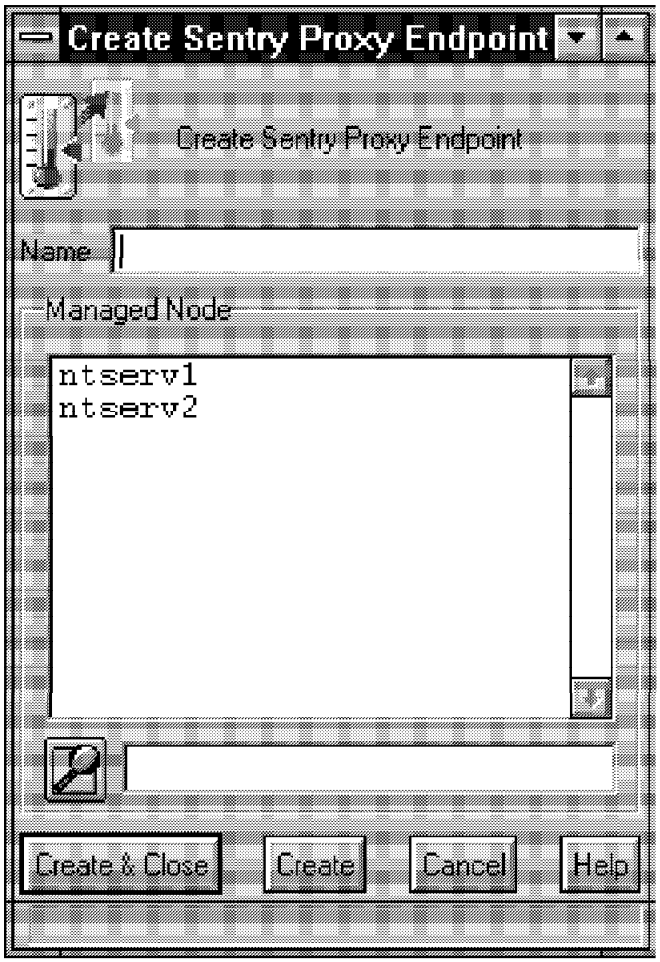


Figure 264. Sentry Proxy Window

In the window that appears you can select a Managed Node on which you want to install the proxy endpoint. Enter a label for the proxy endpoint. After the definition select the **Create & Close** button. From the command line use the command wcrtprx with the following syntax:

```
wcrtprx policy-region managed-node proxy-name
```

The parameters are:

- policy-region - Name of the Policy Region in which you want create the proxy endpoint.
- managed-node - Name of the node the proxy will reside on.
- proxy-name

For example:

```
wcrtprx ntserv1-region ntserv1 Sentry1
```

To work with a proxy endpoint you have to do the following things:

- Set the properties - Open the endpoint by double-clicking on the icon. Then select **Set environment** from the Configure menu.

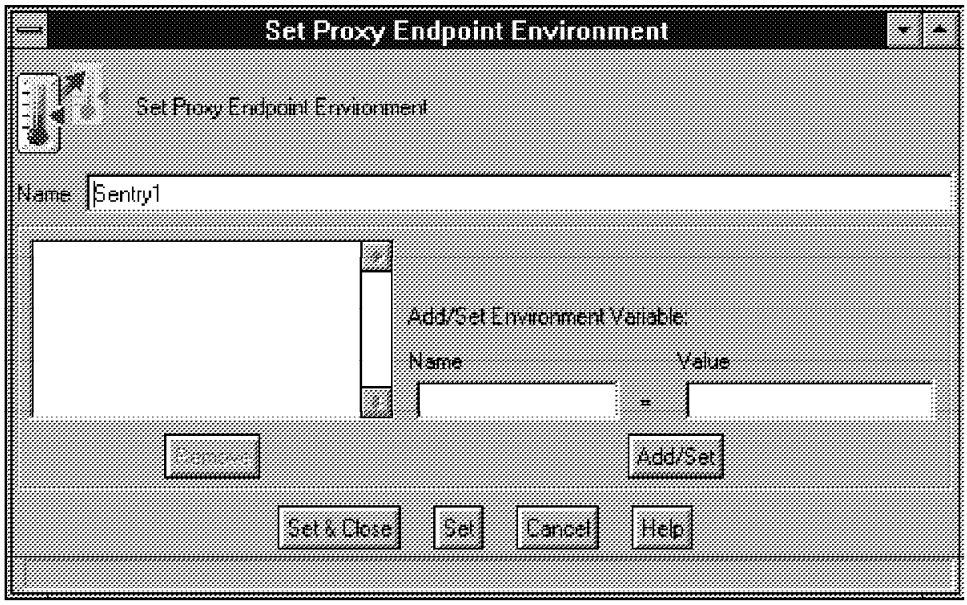


Figure 265. Set Environment

In this window you can add and set environment variables. There are several variables available. Please refer to the Tivoli User Guide for further information. We only discuss one or two of them.

One variable is the ADMIN variable. If you set this variable, the name of the administrator is responsible for each distribution through the endpoint. Another variable, HOST, gives back the label of the Managed Node where the proxy resides. For each of these variables you have to fill in a value. These variables are sent to the engine if it is in monitoring state.

- Set monitor filters - You can set monitor filters by selecting **Set Monitor Filter** from the Configure pull-down menu.

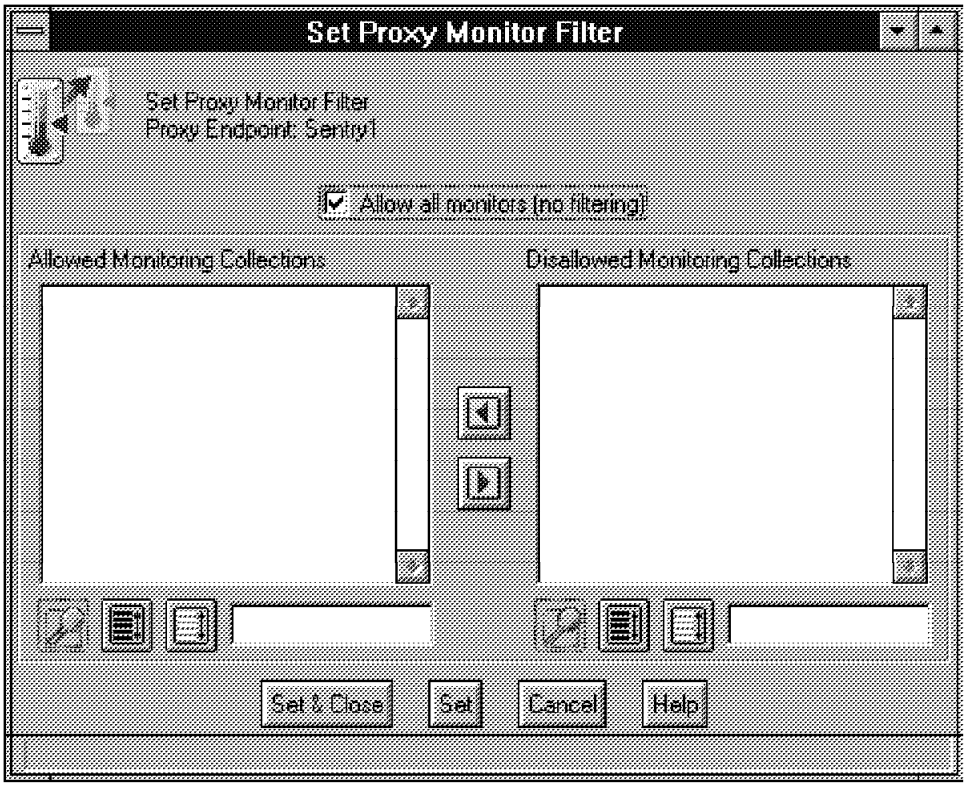


Figure 266. Set Monitor Filter

By default, the Allow all monitors check box is set on. That means that no monitor is filtered out. If you deselect the check box, the monitors will be displayed in the left list box. Now you can select several that will not be monitored.

Note

If you have made any changes to a monitor or profile, you must distribute the profile again. Changes are not automatically distributed through Sentry.

7.1.12 NT Sentry Examples

The following are some examples of using Sentry to monitor various platforms within our NT environment.

We used the NT-specific Sentry monitors in conjunction with the universal monitors for monitoring an NT Server. With the NT monitors we showed how to check for failed logins, and also for the number of connections from other NT machines. Using the universal monitors, we examined the percentage of disk space available and assigned different error levels for each condition.

The first thing is to create a Profile Manager. We labeled this profile manager Profiles. Here we placed all our profiles for these examples. The following window shows the creation of a profile manager and can be achieved by selecting **Create** and **ProfileManager** from within the Policy Region that the profile manager will reside.

Figure 267. Sentry Examples - Creating the Profile Manager

We now have defined an empty profile manager. By selecting this Profile Manager and opening it, we can define profiles that will reside within this profile manager and also define the subscribers to these profiles. Opening the profile manager is achieved by either double-clicking on the icon representing the profile manager or using the right mouse button and selecting **Open**. Within the Profile Manager a profile is created by selecting **Create** and **Profile**. The following window is presented. Here we can see profiles of type SentryProfile. When other Tivoli applications are added then other profile types are available, for example Tivoli/Courier has a different type of profile.

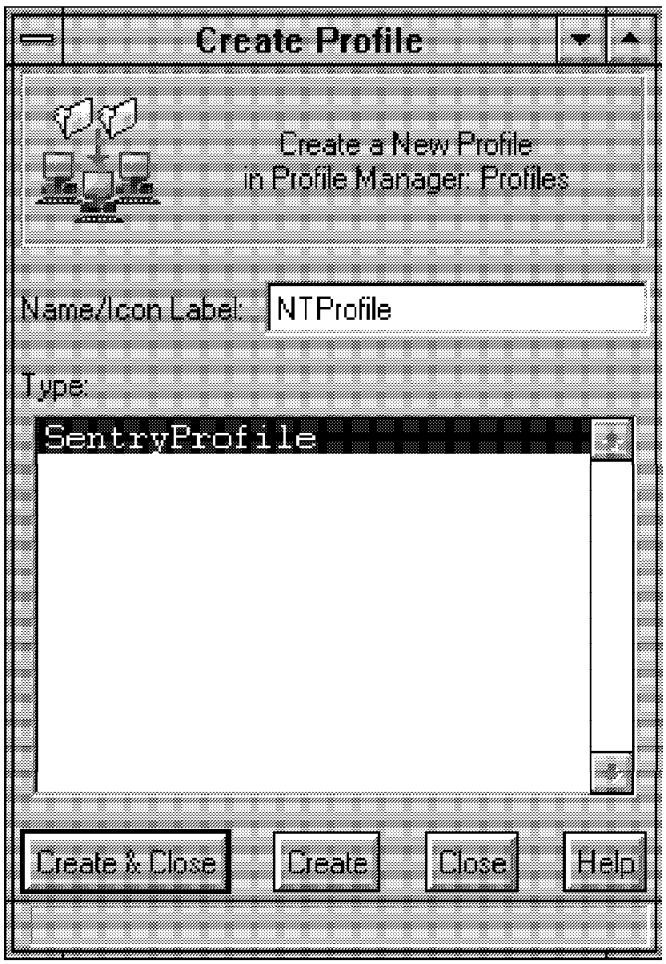


Figure 268. Sentry Example - Creating a Sentry Profile

After creating the profile manager and profile, the Profile Manager is populated with the new profile. Initially it is empty with no subscribers.

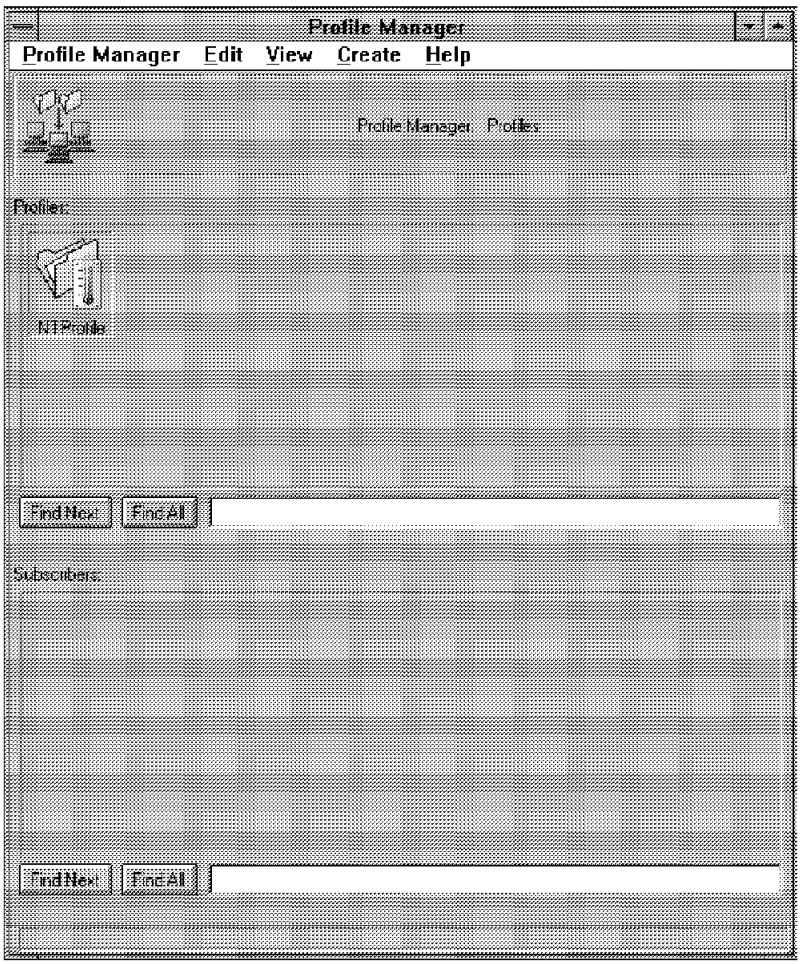


Figure 269. Sentry Example - Sentry Profile Manager and Profile

The next stage is to define the contents of the profile we have just created by double-clicking on the profile icon. We are then presented with the Sentry Profile Properties window where we can define the contents of our profile.

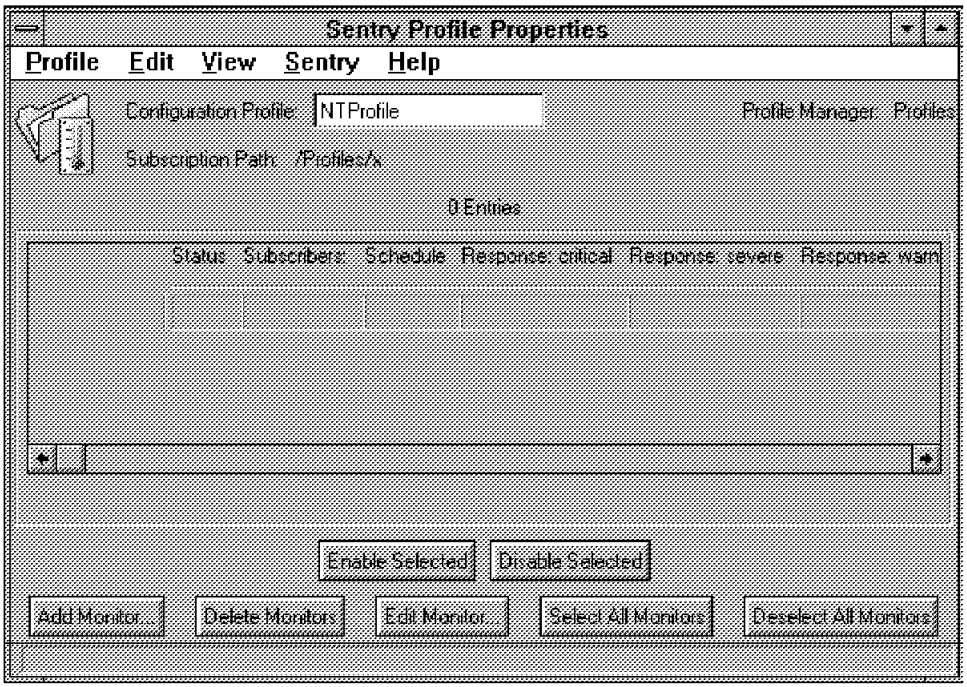


Figure 270. Sentry Example - Defining the Properties of the Sentry Profile

As can be seen from the previous window, the Sentry Profile is empty. We can now proceed and add monitors that we require for this profile. This is achieved by selecting the **Add Monitors** button.This enables us to add empty monitors to our profile for configuration. Here we can see in the next window, the different monitoring collections in the left-hand pane, alongside, in the right-hand pane, the monitoring sources.

Here we defined a monitor to check for logon errors.

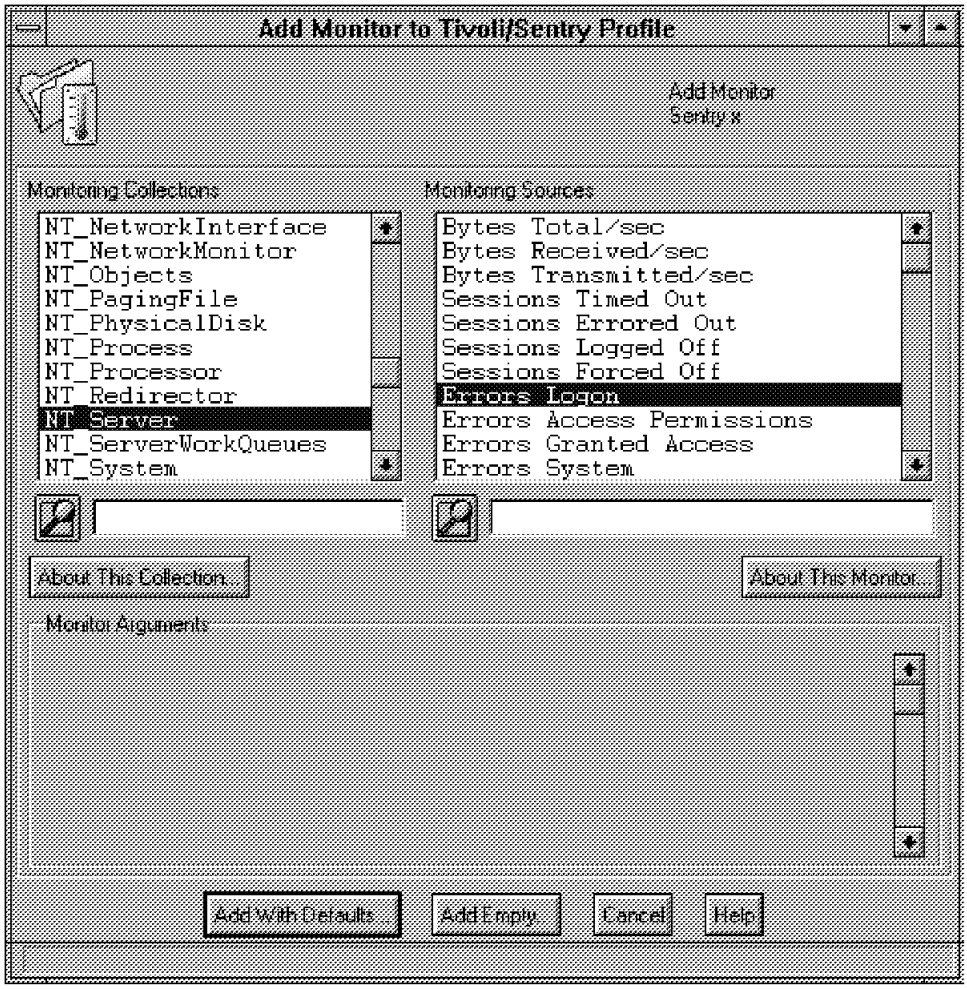


Figure 271. Sentry Example - Adding Sentry Monitor

The logon errors monitor has now been added, but is empty. From the Sentry Profile Properties window we can now select **Add Empty** and define the different settings for this monitor. The next window shows editing the Sentry Monitor. Here we can define the different response levels and associated actions such as sending Tivoli notices, sending E-mail, sending pop-ups, logging to files or running programs.

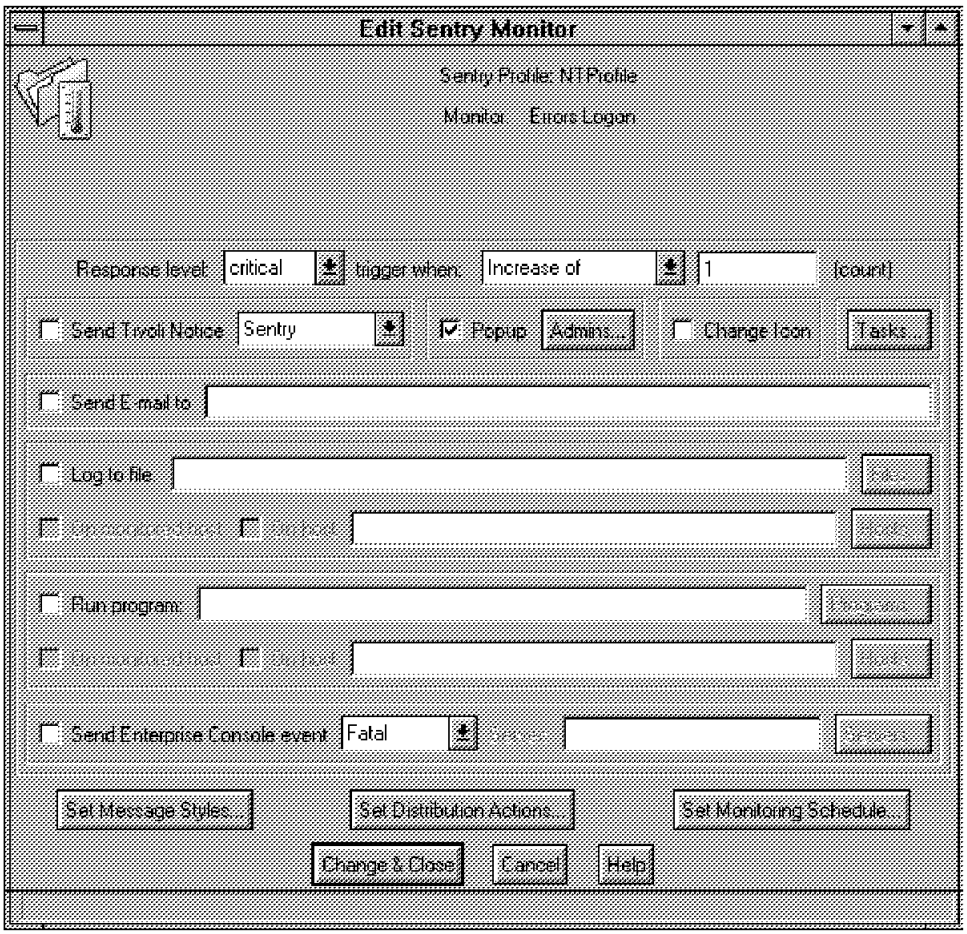


Figure 272. Sentry Example - Editing Sentry Monitors

We can also define the monitoring schedule for these monitors by choosing **Set Monitoring Schedule**. This presents us with the following window, which can be changed to the desired monitoring parameters.

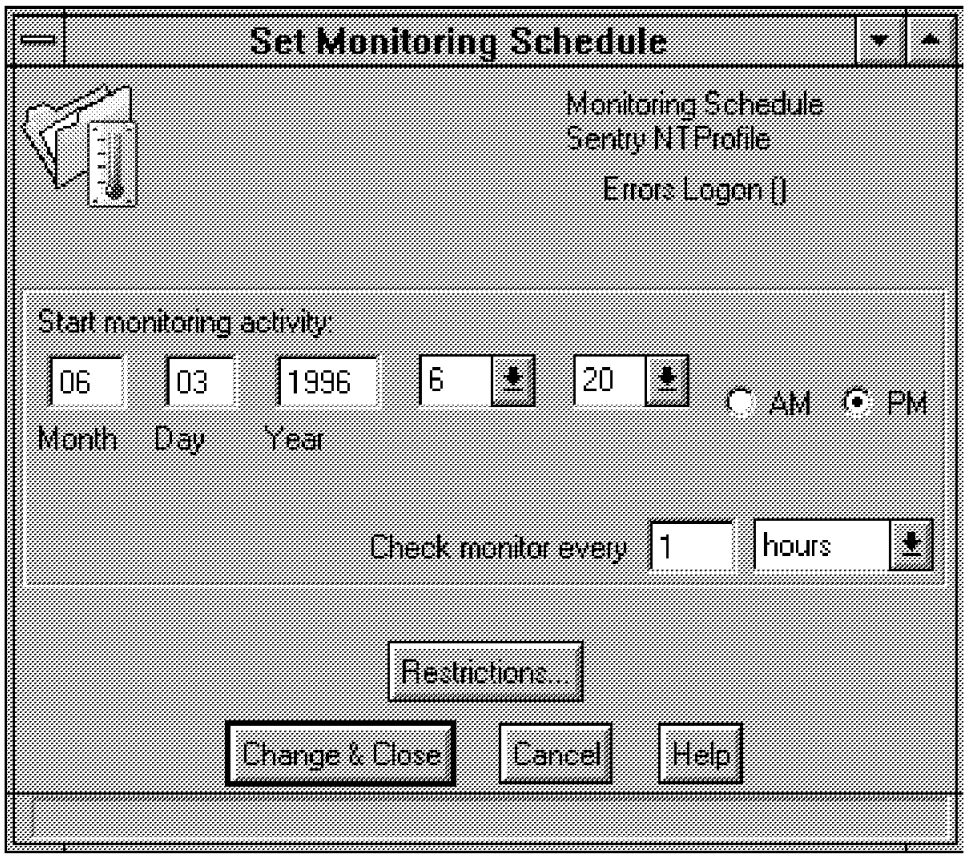


Figure 273. Sentry Example - Setting the Monitoring Time Parameters

Following this procedure, it is possible to create as many monitors as is desired for the subscribers, which will be added later. In the following window, we can see some monitors defined for the profile, NTProfile. These monitors include checking disk space thresholds, failed login attempts and windows NT connections.

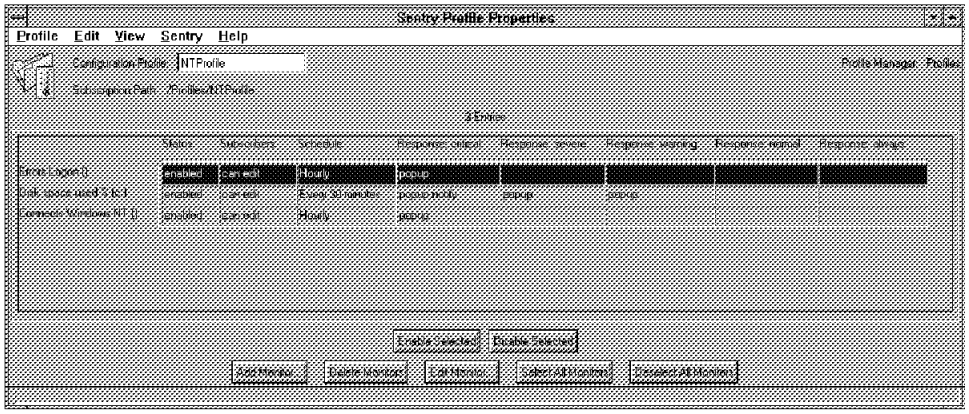


Figure 274. Sentry Example - NTProfile Properties

These are the properties of the NTProfile Sentry profile shown with the graphical user interface. These can also be viewed from the command line by issuing the command wlsmon NTProfile. The following output is obtained using this command:

```
Monitor: Connects Windows NT()
Timing:Hourly
Responses:
  critical
  when probe result  > 0
    popup(Root_ntserv2-region)
  severe
  warning
  normal
  always
Monitor: Errors Logon()
Timing:Hourly
Responses:
  critical
  when probe result absolute change >= 1
    popup(Root_ntserv2-region)
  severe
  warning
  normal
  always
Monitor: Disk space used %(c:)
Timing:Every 30 minutes
Responses:
  critical
  when probe result  > 80
    popup(Root_ntserv2-region),notify(Sentry)
  severe
  when probe result  > 60
    popup(Root_ntserv2-region)
  warning
  when probe result  > 40
    popup(Root_ntserv2-region)
  normal
  always
```

Figure 275. Sentry Profile

The previous output shows the following.

- First Monitor
  - Monitoring Collection - NT_Redirector
  - Monitoring Source - Connects Windows NT
  - Timing - Every hour
  - Critical Response - Send a pop-up to the Administrator Root_ntserv2-region when there are greater than 0 NT connections
- Second Monitor
  - Monitoring Collection - Universal
  - Monitoring Source - Disk Space used %
  - Timing - Every 30 minutes
  - Critical Response - Send a pop-up to the Administrator Root_ntserv2-region and a Sentry notice when there is greater than 80% usage of the C drive

  - Severe Response - Send a pop-up to the Administrator Root_ntserv2-region when there is greater than 60% usage of the C drive
  - Warning Response - Send a pop-up to the Administrator Root_ntserv2-region when there is greater than 40% usage of the C drive

- Third Monitor
  - Monitoring Collection - NT_Server
  - Monitoring Source - Errors Logon
  - Timing - Every Hour
  - Critical Response - Send a pop-up to the Administrator Root_ntserv2-region, when there is a change in the number of failed logins.

As of yet we still have no subscribers to this profile. To add a subscriber for this profile, we can either drag and drop the desired Managed Node into the profile manager, or select **Profile Manager** and **Subscribers** from the Profile Manager window to select the subscribers for this profile.

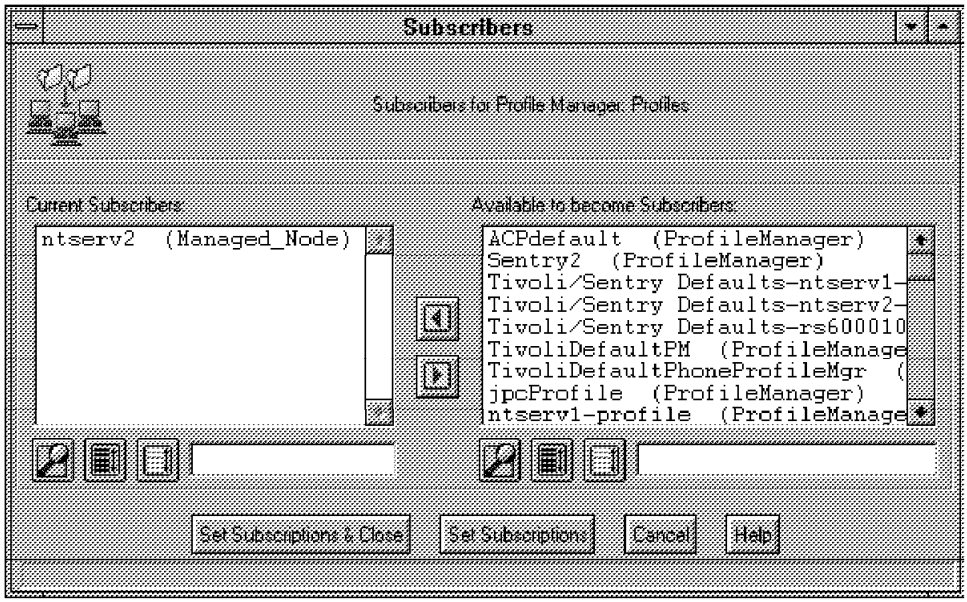


Figure 276. Sentry Example - Selecting Subscribers

By selecting **Set Subscriptions & Close**, this Managed Node will subscribe to the previous sentry profile that has been defined. All that remains to be done is to distribute the profile to the subscribers. This can be done by selecting **Profile Manager** and **Distribute**. The next two windows show the completed profile manager with a defined profile and subscriber. This is followed by an example of the profile distribution window.

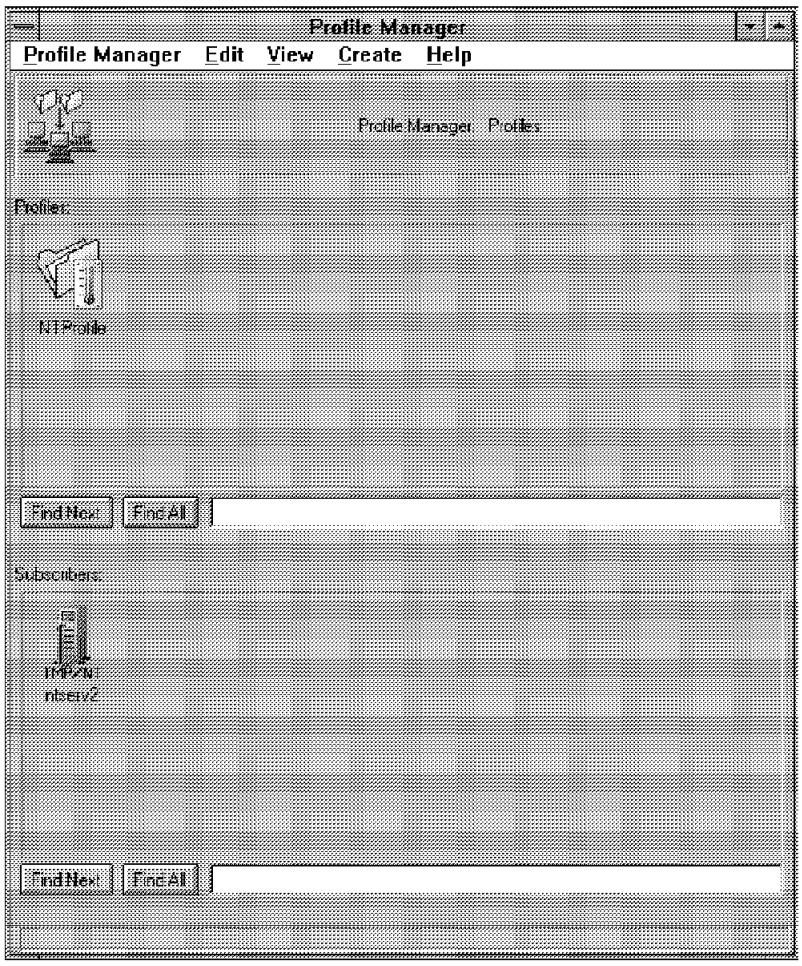


Figure 277. Sentry Example - Profile Manager with Subscribers

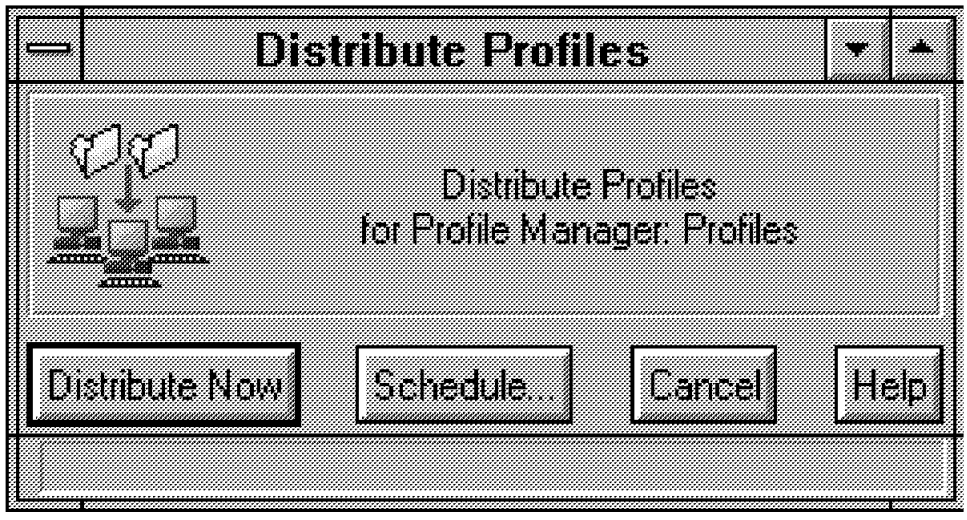


Figure 278. Sentry Example - Profile Distribution

Next we show the three windows that pop up when any of the previous monitors are triggered.

The first alert shows that there are greater than 0 connections to our monitored NT Server. You probably would not have set a threshold for this metric so low, but we just used it as an easy example of how to monitor for excessive connections to a server. The error would show up as follows:

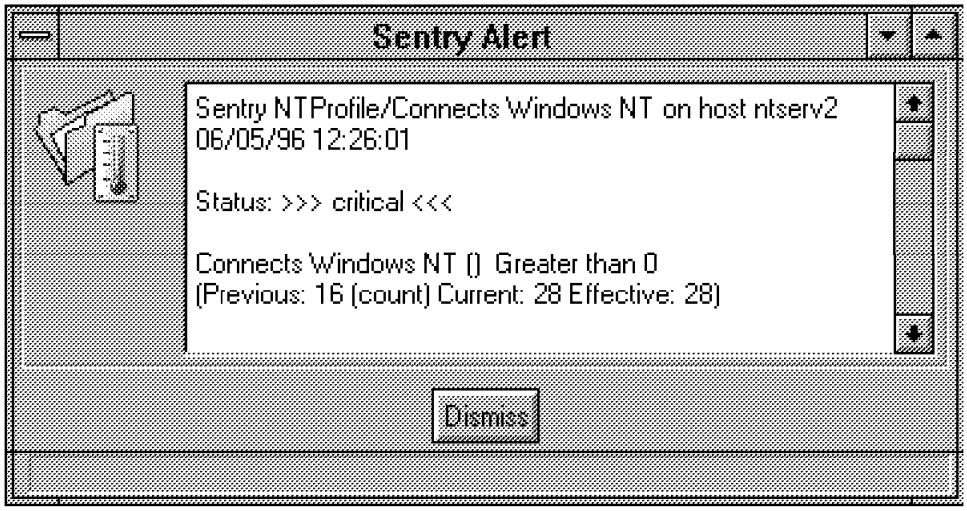


Figure 279. Sentry Example - Sentry Alert for Windows NT Connections

The second alert is the disk space threshold monitor. This has alerted the administrator at the ntserv2-region TMR that there is currently greater than 40% disk space being used and what the previous value was.

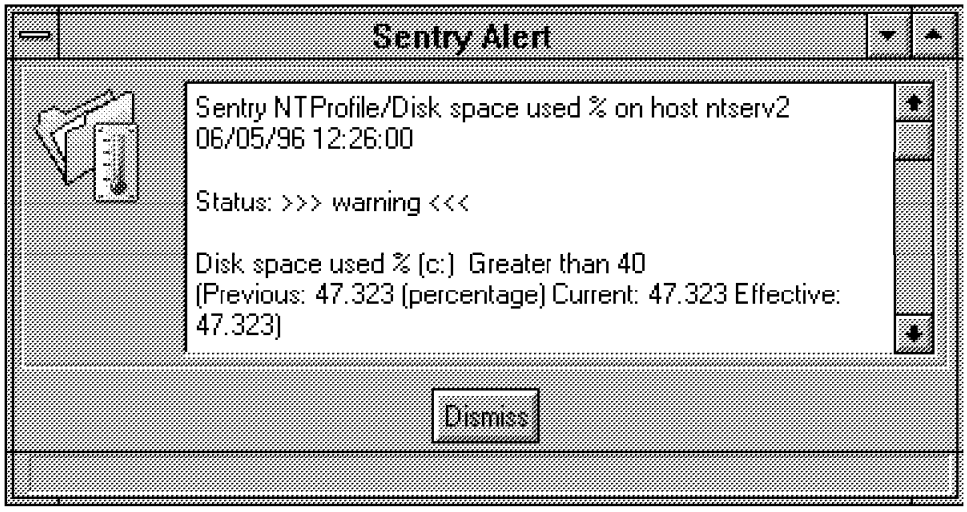


Figure 280. Sentry Example - Sentry Alert for Disk Space Thresholds

The third alert in this example displays the message stating that a failed login to the NT server ntserv2, the domain controller for the TMENTDOM domain, has occurred. This is monitored using a counting mechanism, so that if the count increases by one over the last time the monitor was run, another alert will get sent.

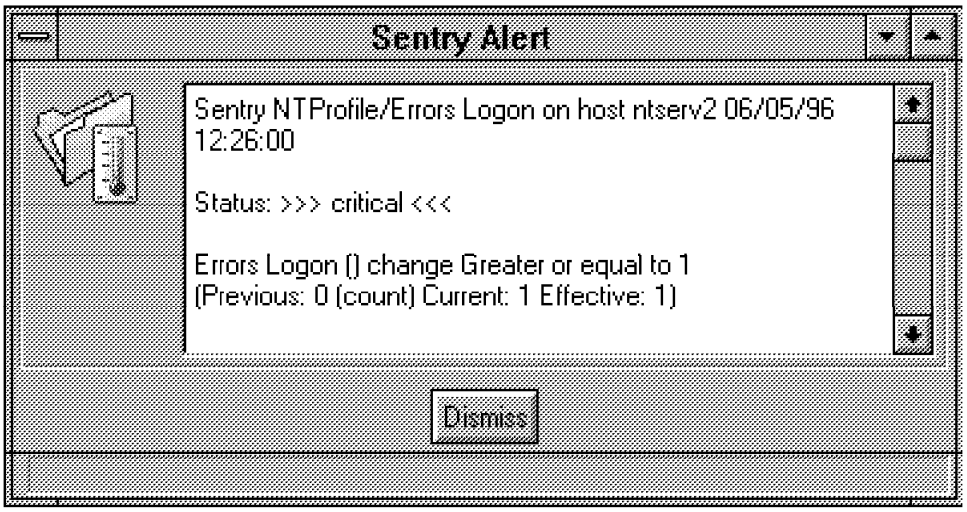


Figure 281. Sentry Example - Sentry Alert for Failed NT Logons

7.1.13 Sentry Proxy Configuration for Unmanaged Nodes

Within our configuration, we have a number of PcManagedNodes defined. These are clients with the TME agent installed. To manage these nodes we can use the Sentry monitors defined on a Managed Node to get management data. The steps for creating the profiles and subscribing to them are as described previously. Here we shall define proxy endpoints to enable management of OS/2, Windows 95 and an SNMP managed network hub.

7.1.14 OS/2 Warp Connect Example

The first stage in managing an OS/2 machine is to create a Sentry proxy in the Policy Region where it will reside by selecting **Create** and **SentryProxy**. A Sentry proxy endpoint requires a unique name and a Managed Node through which it can be monitored. In this case we call the Sentry proxy OS2_tmecli4_proxy, and it will be monitored from ntserv2.

Figure 282. Sentry Example - Creating Sentry Proxy Endpoint

By opening the Sentry proxy endpoint, it can be configured.

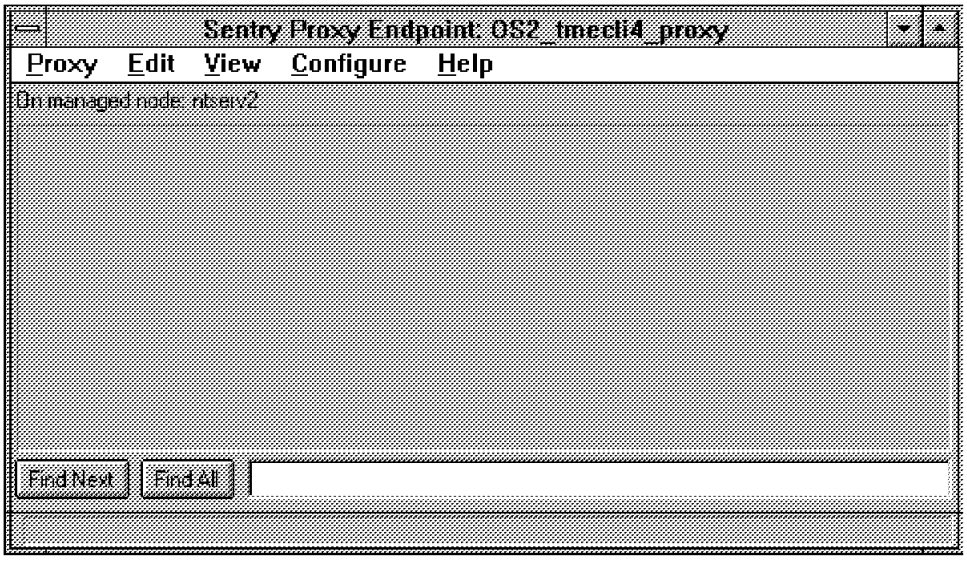


Figure 283. Sentry Example - Configuring Sentry Proxy Endpoint

There are two parts to configuring the proxy. The first part being the proxy environment configuration part. Here a number of environment variables can be configured to be passed to the proxy endpoint. In our example we used and defined the ENDPOINT variable to be the same as the IP hostname of the device we managed, tmecli4.

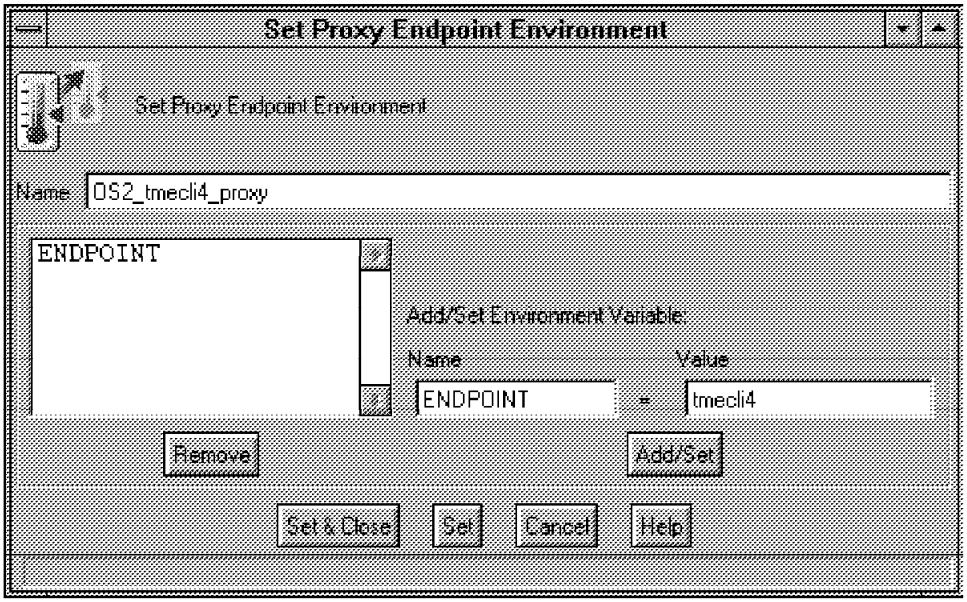


Figure 284. Sentry Example - Configuring Sentry Proxy Endpoint Environment

We then set up the monitor filter mechanism to allow access to the available monitors on the Managed Node we are using for our proxy. By default these are all disabled as in the following diagram.

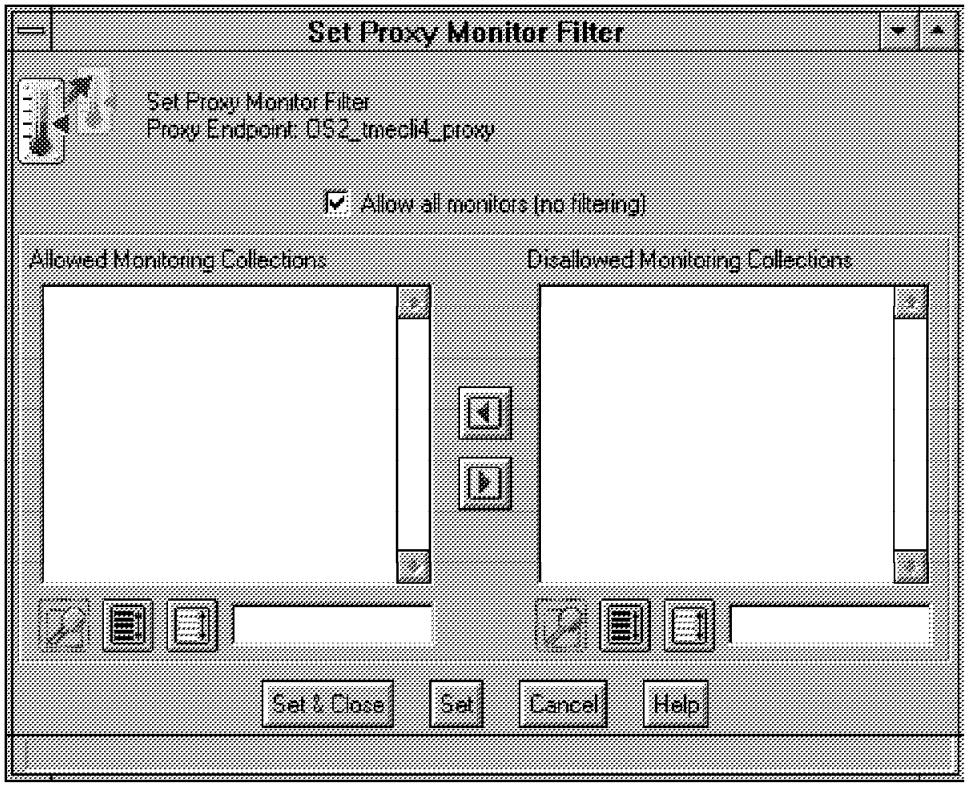


Figure 285. Sentry Example - Configuring Sentry Proxy Monitor Filters (1)

By deselecting this check box we can enable all the monitors.

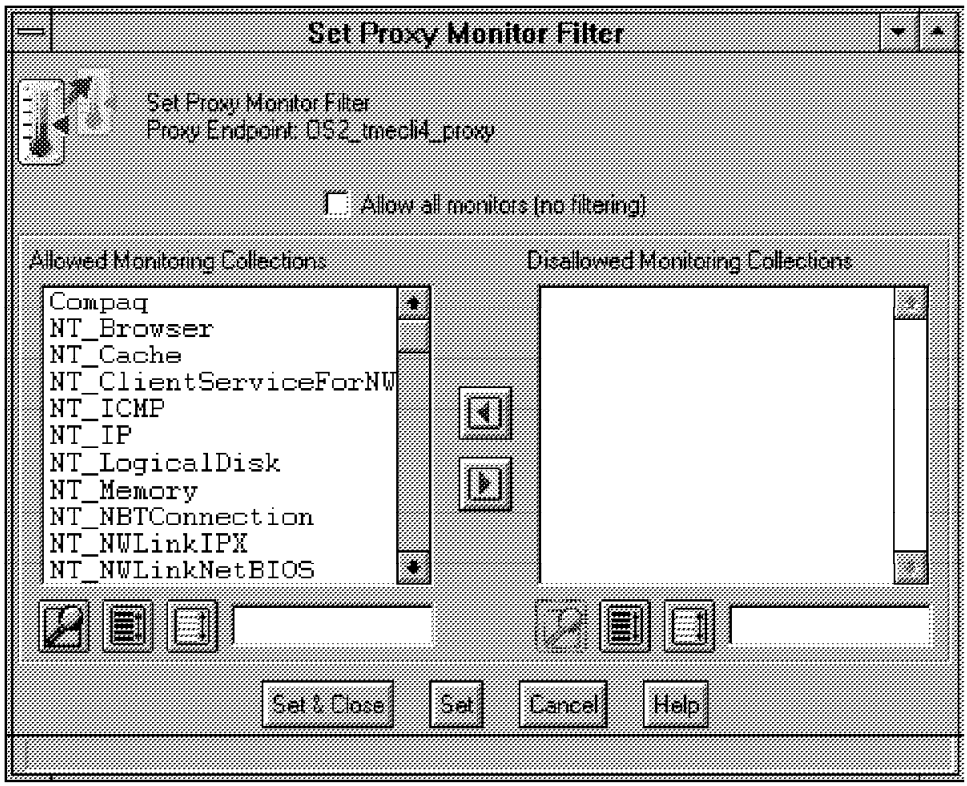


Figure 286. Sentry Example - Configuring Sentry Proxy Monitor Filters (2)

We now have defined and configured the proxy endpoint to be monitored through Managed Node ntserv2. All that has to be done now is to add and define valid Sentry monitors, for this OS/2 PcManagedNode.

As this is a PcManagedNode, we can use the Tivoli RFC1213 Monitoring Collection to get SNMP management information from OS/2. This requires SNMP to be enabled on the OS/2 client and on the Managed Node used for the proxy ntserv2.

7.1.14.1 SNMP Configuration for OS/2

For the OS/2 client we used the IBM TCP/IP V2.0 SNMP Agent. This was configured to autostart with the following parameters.

```
snmpd -d 4 -transport udp -dpi tcp

SNMP BASE - V0.14i, Compiled Feb 24 1995 at 16:41:29
enterprise oid: 1.3.6.1.4.1.2.6.46
agentAddress:   9.24.104.80 ('09186850'h)
generic-trap:   coldStart ('00000000'h)
specific-trap:  0 ('00000000'h)
time-stamp:     100 - 1.0 seconds

T:\TCPIP20\BASEKIT\SRC\SNMP\NVOTCP\AGENT\MP\SRC\
SNMP_MP_AGENT.C - Version 0.14mp - Feb 24 1995 at 16:50:01
Tracing set to 4
     4 Traps
using default IP port 161
SNMP logging data follows ============================
Log_type:       snmpLOGtrap_out
Destinations:   1
   send rc:     0
   destination: UDP 9.24.104.124 port 162
trap PDU data:
community:      public ('7075626c6963'h)
End of SNMP logging data: ============================
SNMP AGENT: we are ready to receive requests...
```

Figure 287. SNMPD Startup on OS/2

The purpose of the *-d* option was so that we could view outbound traps. We also could have redirected the output to a file for viewing at a later time.

7.1.14.2 SNMP Configuration for NT Server

To configure SNMP for NT the service has to be installed and configured. The configuration can be done from the control panel by selecting **Network** and then **Configure** on the SNMP Service installed network software option.

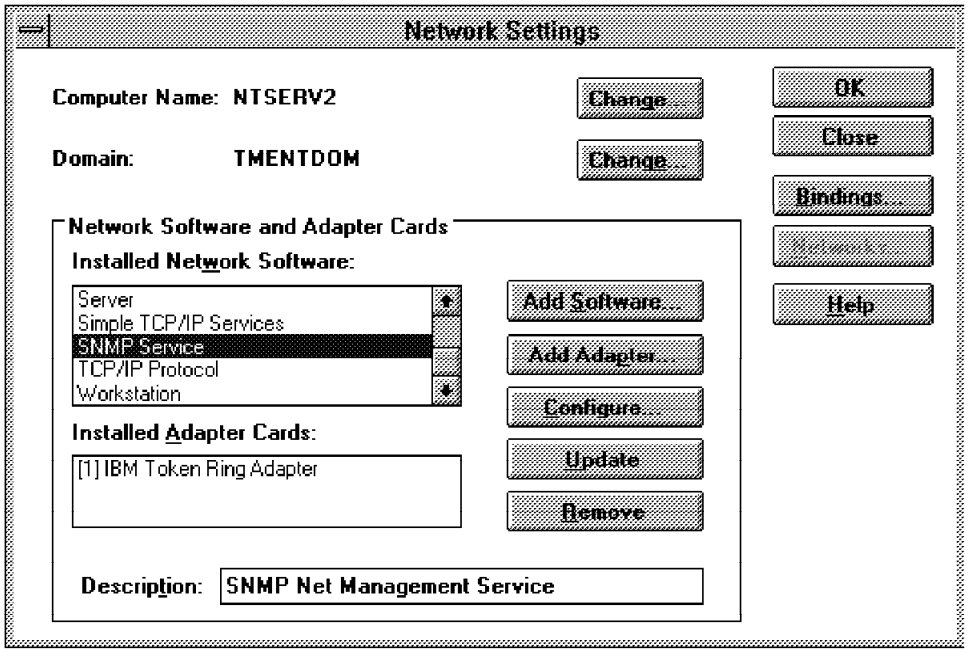


Figure 288. Sentry Example - Configuring NT SNMP (1)

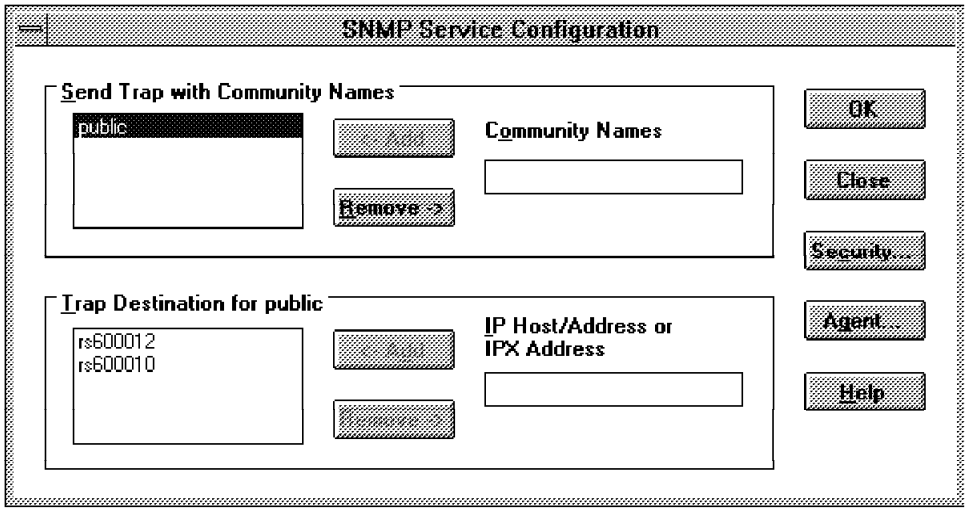


Figure 289. Sentry Example - Configuring NT SNMP (2)

In the previous window, community names can be configured, as can trap destinations. By selecting the **Agent** button, the SNMP agent can be configured as seen in the next window.

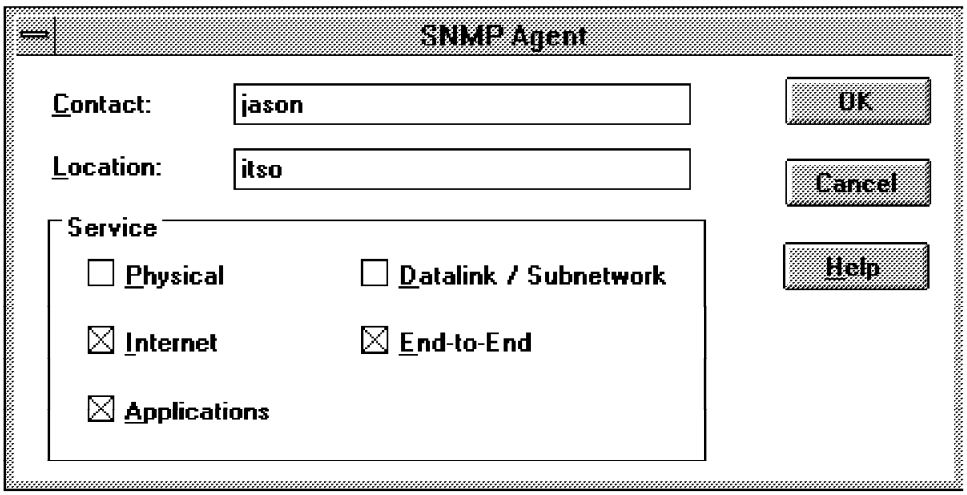


Figure 290. Sentry Example - Configuring NT SNMP (3)

Now that NT SNMP has been configured, it can be started and stopped like any other NT service with the following commands:

```
net start SNMP
    or
net stop SNMP
```

Figure 291. Controlling the SNMP Service

Defining and configuring the Sentry Monitor is the same in this example as the previous configurations. We just need to choose the correct Sentry monitor. In this case the monitor used will be the monitoring collection for RFC 1213. This monitor allows management data to be received conforming to RFC 1213 MIB-II definitions.

The first window shows selecting the Sentry monitor we wish to use with the Add Monitor button. Here we select to monitor the Host Contact monitor, for the SNMP host that satisfies the environment variable ${ENDPOINT}, which we have defined when creating the proxy endpoint.

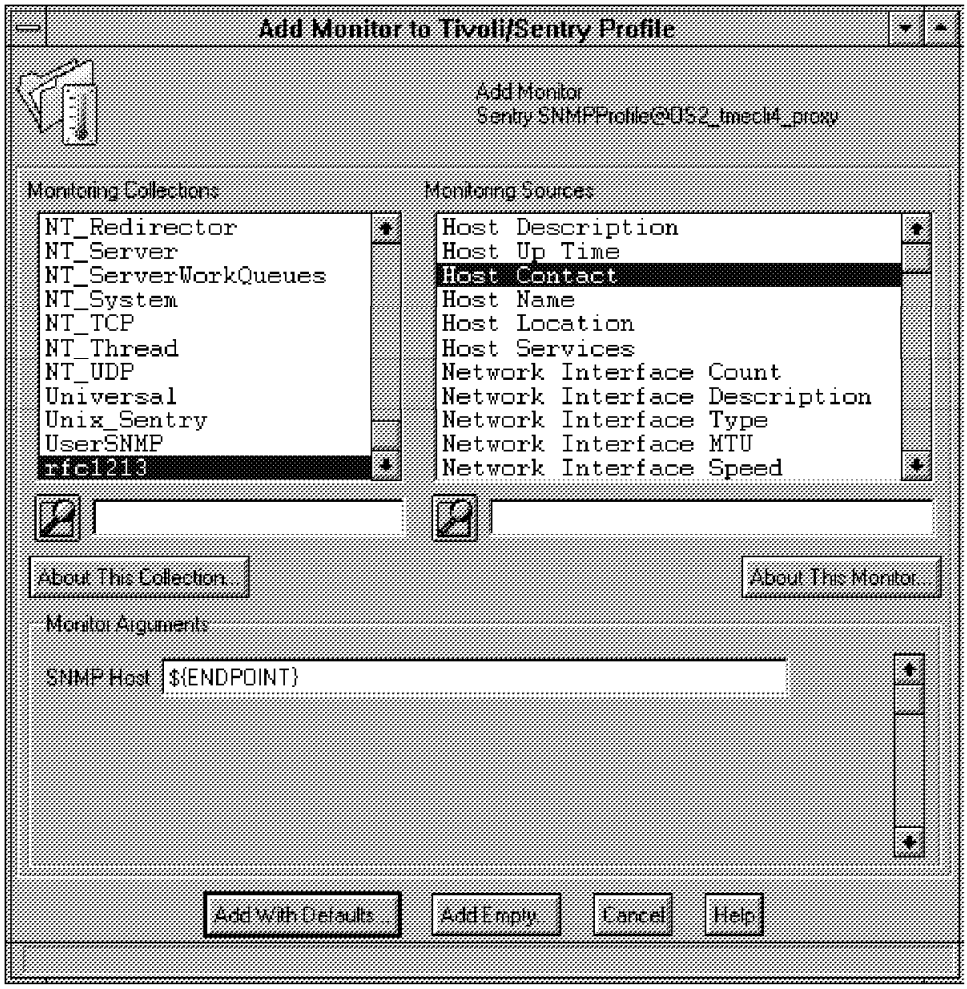


Figure 292. Sentry Example - Selecting RFC1213 Sentry Monitor

Now what we can do is add other SNMP monitors to this profile to retrieve SNMP MIB-II data from our SNMP host ${ENDPOINT}, tmecli4. We have the following monitors defined.

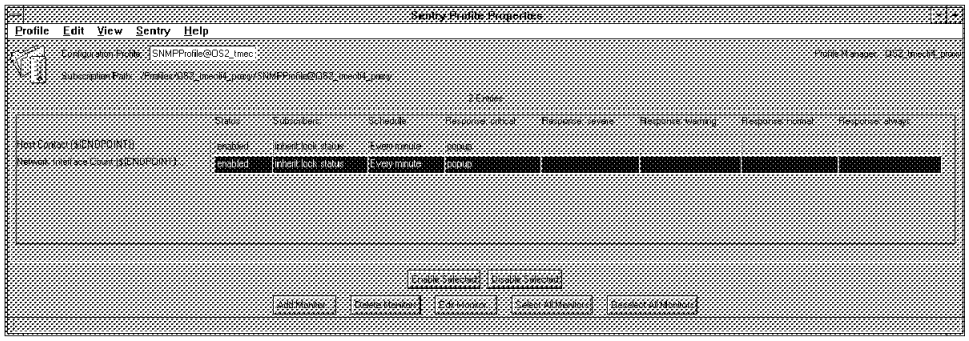


Figure 293. Sentry Example - Defined Monitors for RFC1213 Sentry Collection

From the command line this is represented as:

```
1  Monitor: Network Interface Count(${ENDPOINT})
  Timing:Every minute
  Responses:
    critical
    when probe result  > 0
      popup(Root_ntserv2-region)
    severe
    warning
    normal
    always
2  Monitor: Host Contact(${ENDPOINT})
  Timing:Every minute
  Responses:
    critical
    when probe result  != lopi
      popup(Root_ntserv2-region)
    severe
    warning
    normal
    always
```

Figure 294. wsnmpget Script

We have two monitors querying MIB-II information on the OS/2 client tmecli4. Only one of these will return an alert to the defined administrators desktop. The reason for this is that the SNMP agent on OS/2 does not support a full MIB-II standard. When we queried the object ID for the first monitor we had defined to obtain the network interface count, we got no response as this portion of the MIB is not available from OS/2 SNMPD.

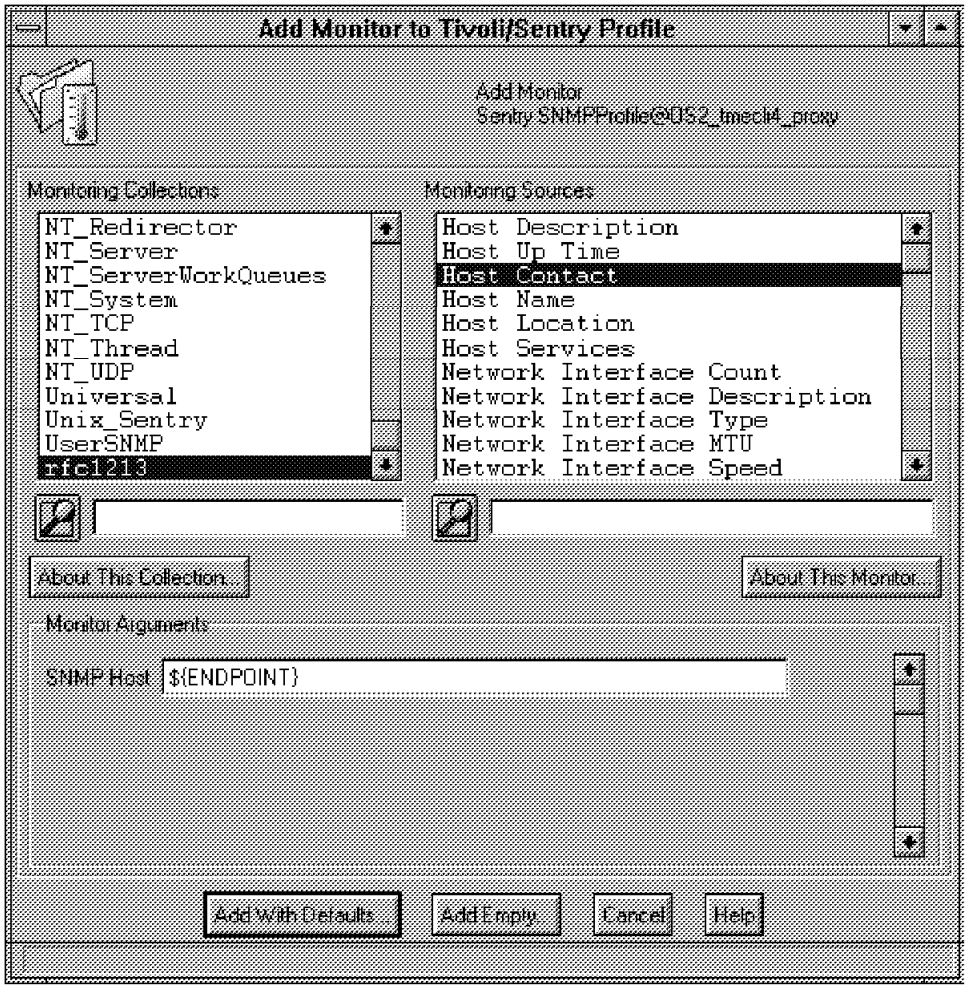


Figure 295. Sentry Example - Selecting RFC1213 Sentry Monitor

The way the SNMP RFC1213 monitoring collection works is with a script that comes with the TME platform. This script passes variables which you can define for each proxy that you create. The script calls the Tivoli command wsnmpget.exe which queries the MIB and object ID in question.

```
#!/bin/sh
################################################################
# Script used to access the SNMP value of a given MIB entry.
# Used for all generic SNMP-value based monitors
#
# $Id: wsnmpget.sh,v 1.1.2.7 1996/05/20 02:08:08 greg Exp $
################################################################

find_snmpdata_dir()
{
       BASEDIR=${INSTALLDIR:-"++NO-BINDIR++"}
       SNMPDATADIR=$BASEDIR/generic/SNMP

       if [ ! -d $SNMPDATADIR ] ; then
              echo "SNMP Data directory $SNMPDATADIR doesnt exist" >&2
              echo "(Environment variable SNMPDATADIR)" >&2
              echo "$SNMPDATADIR not found" > $LOGFIL.err 2>&1
              exit $BAD
       fi
}

load_collection_env()
{
```

```
        #
        # Load collection-specific environment
        #
        if [ -z "$SNMPDATADIR" ] ; then
              find_snmpdata_dir
        else
              if [ -f $SNMPDATADIR/$COLLECTION.ENV ] ; then
                    . $SNMPDATADIR/$COLLECTION.ENV
              fi
        fi
}

set_required_vars()
{
        #
        # Set variables if they are not already present from the
        # collection's environment data file (or proxy environment)
        #
        if [ -z "$DEBUG" ] ; then
              DEBUG=0                   # if non-0 then log to  $LOGFIL.tst
        fi
        if [ -z "$TIMEOUT" ] ; then
              TIMEOUT=5
        fi
        if [ -z "$COMM" ] ; then
              COMM="public"
        fi
        if [ -z "$ERRORS" ] ; then
              ERRORS="fail"             # if !ignore then wsnmpget errors cause
        fi                        # total failure
        if [ -z "$COMP" ] ; then
              COMP="exact"        # can be exact|min|max.  if min|max then
        fi                        # returned value will be lowest|highest
        if [ -z "$LOGFIL" ] ; then
              LOGFIL=/tmp/SnmpMonitor
        fi
        if [ -z "$INTERP" ] ; then
              INTERP=unknown
        fi
        if [ -z "$HOST" ] ; then
              HOST=unknown
        fi
}

expand_target()
{
        TGT="$1"

        case $TGT in
              @* )   # target was specified as a list
                     if [ ! -f $SNMPDATADIR/$COLLECTION.$TGT ] ; then
                           echo "List data file not present" >&2
                           echo "$SNMPDATADIR/$COLLECTION.$TGT not found" >&2
                           exit $BAD
                     fi
                     COMP=head -n 1 $SNMPDATADIR/$COLLECTION.$TGT
                     LIST=tail -1 $SNMPDATADIR/$COLLECTION.$TGT
                     ;;
              * )
                     COMP="exact"
                     eval LIST=\""$TGT"\"
                     ;;
        esac
}

seed_init_value()
{
        case $COMP in
              "min" ) CUM_VAL=99999999
                     ;;
              "max" ) CUM_VAL=0
                     ;;
              *) CUM_VAL=0
                     ;;
```

```
        esac
}


request_data()
{
        for i in $LIST
        do
                VAL=wsnmpget -h $i -t $TIMEOUT -c $COMM $ENTRY 2>&1
                RET=$?

                if [ $DEBUG ] ; then
                        echo "Data ($i)  : $VAL" >> $LOGFIL.tst
                fi

                case $COMP in
                        "exact" )
                                CUM_VAL=$VAL
                                ;;
                        "min" )
                                if [ $VAL -lt $CUM_VAL ] ; then
                                        CUM_VAL=$VAL
                                fi
                                ;;
                        "max" )
                                if [ $VAL -gt $CUM_VAL ] ; then
                                        CUM_VAL=$VAL
                                fi
                                ;;
                esac
                echo "$VAL  $CUM_VAL" >> $LOGFIL.tst

                if [ $RET ] ; then
                        if [ $ERRORS != "ignore" ] ; then
                                RETVAL=$RET
                        fi
                fi
        done
}

#########################################################################
#
# Main
#
#########################################################################
BAD=1

if [ $# -ne 3 ] ; then
        echo "Probe Version    : $Id: wsnmpget.sh,v 1.1.2.7 1996/05/20 02:08:08 greg Exp $" > $LOGFIL.err 2>&1
        date >> $LOGFIL.err
        echo "$# (args)        : $*" >> $LOGFIL.err
        echo "Usage: $0 collection probe-name device" >&2
        exit $BAD
else
        COLLECTION="$1"           # Monitor Collection name
        PROBE="$2"          # Collection probe name
        TARGET="$3"         # IP name, address or list (@list-name)
fi


load_collection_env

set_required_vars

#
# Load parameters
#

if [ -f $SNMPDATADIR/$COLLECTION.OID ] ; then
        ENTRY=sh $SNMPDATADIR/$COLLECTION.OID $INTERP $HOST $PROBE
else
        echo "Could not translate probe $PROBE to SNMP OID" >&2
        echo "Error in translation file $SNMPDATADIR/$COLLECTION.OID" >&2
        echo "Translation File Error:  $PROBE, $SNMPDATADIR, $COLLECTION" > $LOGFIL.err 2>&1
```

```
        exit $BAD
fi

if [ -z "$ENTRY" ] ; then
        echo "Could not translate probe $PROBE to SNMP OID" >&2
        echo "Probe not found in translation file $SNMPDATADIR/$COLLECTION.OID" >&2
        echo "Translation Probe Error:  $PROBE, $SNMPDATADIR, $COLLECTION" > $LOGFIL.err 2>&1
        exit $BAD
fi

expand_target "$TARGET"

#
# Run the command ( with or without debugging)
#

if [ $DEBUG -eq 1 ] ; then
        echo "Probe Version    : $Id: wsnmpget.sh,v 1.1.2.7 1996/05/20 02:08:08 greg Exp $" > $LOGFIL.tst 2>&1
        echo "Probe Version    : $Id: wsnmpget.sh,v 1.1.2.7 1996/05/20 02:08:08 greg Exp $" > $LOGFIL.env 2>&1
        echo "Environment During Probe"  >> $LOGFIL.env 2>&1
        env >> $LOGFIL.env 2>&1
        echo "command used     : wsnmpget -h $TARGET -t $TIMEOUT -c $COMM $ENTRY" >> $LOGFIL.tst
        echo "expanded targets : $TARGET -> $LIST" >> $LOGFIL.tst
        echo "comparison type  : $COMP" >> $LOGFIL.tst
fi

        seed_init_value
        request_data
        echo $CUM_VAL


if [ $RETVAL -ne 0 ] ; then
        echo "wsnmpget exited with non-zero return code"
        echo "command was: wsnmpget -h $TARGET -t $TIMEOUT -c $COMM $ENTRY"
fi

exit $RETVAL
#exit 0
```

Note: The line in this script LOGFIL=/tmp/ SnmpMonitor causes an error under NT. Make sure that a tmp directory exists on the drive your Tivoli Management Platform is installed. Otherwise an error condition occurs.

As stated earlier, not all MIB-II information is guaranteed. When we used SNMP Sentry Monitors, the above shell script created an SnmpMonitor.tst file under /tmp on the Tivoli installation drive. In that file there is a list of alerts that had happened showing the previous value and the gcurrent value. If there had been a failure of some kind it is usually that the wsnmpget.exe command had been unsuccessful in obtaining the OID from the desired host. In our case that was the network interface count node.

The following is an example of the SnmpMonitor.tst file:

```
Data (tmecli4)  : Error: noSuchName
Error: noSuchName  Error: noSuchName
Data (tmecli4)  : "Kathryn Kathryn"
"Kathryn Kathryn"  "Kathryn Kathryn"
```

Figure 296. /tmp/SnmpMonitor.tst Output

The Error: noSuchName is returned from the failed attempt to access the OID for network interface count using the Tivoli wsnmpget.exe.

In addition, Tivoli supplies a file for all the SNMP OIDs for MIB-II called, rfc1213.OID. If this file does not exist, the alerts will fail and a file called \tmp\SnmpMonitor.err is created.

```
Translation File Error:  NetIntfCnt, D:-Tivoli-bin/generic/SNMP, rfc1213"
```

```
#!/bin/sh
#
# Translation from Probe Name to MIB OID
#
# $0 $INTERP $HOST $PROBE
#
# Should echo OID to stdout
#
# $Id: rfc1213.OID,v 1.1.2.2 1996/04/19 21:46:19 spofford Exp $
#

NET=2
case $1 in
 hpux10 ) NET=4 ;;
 hpux9 )
  /bin/hp9000s700
  if [ $? -eq 0 ] ; then
   NET=4;
  else
   NET=1;
  fi
  ;;
 axp-osf ) NET=1 ;;
esac

case $3 in
 HostDescr )  echo "1.3.6.1.2.1.1.1.0" ;;
 HostUpTime )  echo "1.3.6.1.2.1.1.3.0" ;;
 HostContact )  echo "1.3.6.1.2.1.1.4.0" ;;
 HostName )  echo "1.3.6.1.2.1.1.5.0" ;;
 HostLocation )  echo "1.3.6.1.2.1.1.6.0" ;;
 HostServices )  echo "1.3.6.1.2.1.1.7.0" ;;
 NetIntfCnt )  echo "1.3.6.1.2.1.2.1.0" ;;
 NetIntfDescr )  echo "1.3.6.1.2.1.2.2.1.2.$NET" ;;
 NetIntfType )  echo "1.3.6.1.2.1.2.2.1.3.$NET" ;;
 NetIntfMTU )  echo "1.3.6.1.2.1.2.2.1.4.$NET" ;;
 NetIntfSpeed )  echo "1.3.6.1.2.1.2.2.1.5.$NET" ;;
 NetIntfAdmnStat )  echo "1.3.6.1.2.1.2.2.1.7.$NET" ;;
 NetIntfOperStat )  echo "1.3.6.1.2.1.2.2.1.8.$NET" ;;
 NetBytesRcvd )  echo "1.3.6.1.2.1.2.2.1.10.$NET" ;;
 NetBytesXmtd )  echo "1.3.6.1.2.1.2.2.1.16.$NET" ;;
 NetBcstBytesRcvd )  echo "1.3.6.1.2.1.2.2.1.12.$NET" ;;
 NetBcstBytesXmtd )  echo "1.3.6.1.2.1.2.2.1.18.$NET" ;;
 NetBytesRcvdErr )  echo "1.3.6.1.2.1.2.2.1.14.$NET" ;;
 NetBytesXmtdErr )  echo "1.3.6.1.2.1.2.2.1.20.$NET" ;;
 NetBytesRcvdDisc )  echo "1.3.6.1.2.1.2.2.1.13.$NET" ;;
 NetBytesXmtdDisc )  echo "1.3.6.1.2.1.2.2.1.19.$NET" ;;
 NetBytesXmtdQlen )  echo "1.3.6.1.2.1.2.2.1.21.$NET" ;;
 IPForwarding )  echo "1.3.6.1.2.1.4.1.0" ;;
 IPDefaultTTL )  echo "1.3.6.1.2.1.4.2.0" ;;
 IPInReceives )  echo "1.3.6.1.2.1.4.3.0" ;;
 IPInDiscards )  echo "1.3.6.1.2.1.4.8.0" ;;
 IPOutRequests )  echo "1.3.6.1.2.1.4.10.0" ;;
 IPOutDiscards )  echo "1.3.6.1.2.1.4.11.0" ;;
 IPOutNoRoutes )  echo "1.3.6.1.2.1.4.12.0" ;;
 TCPMaxConn )  echo "1.3.6.1.2.1.6.4.0" ;;
 TCPEstabReset )  echo "1.3.6.1.2.1.6.8.0" ;;
 TCPCurrEstab )  echo "1.3.6.1.2.1.6.9.0" ;;
 UDPNoPorts )  echo "1.3.6.1.2.1.7.2.0" ;;
 UDPInErrors )  echo "1.3.6.1.2.1.7.3.0" ;;
esac
```

Figure 297. RFC1213.oid

7.1.15 Windows 95 Example

For a Windows 95 client, the same procedure should be followed as for an OS/2 machine. Only the SNMP configuration is different, depending on which SNMP agent is used.

7.1.16 Nways 8238 Stackable Hub Example

To monitor any SNMP device with MIB-II, a similar procedure can be followed. In this example we monitor an IBM 8238 Hub. We created the proxy endpoint as before, calling it nways_itso8224_proxy and configured it in the same manner as the earlier proxy endpoints.

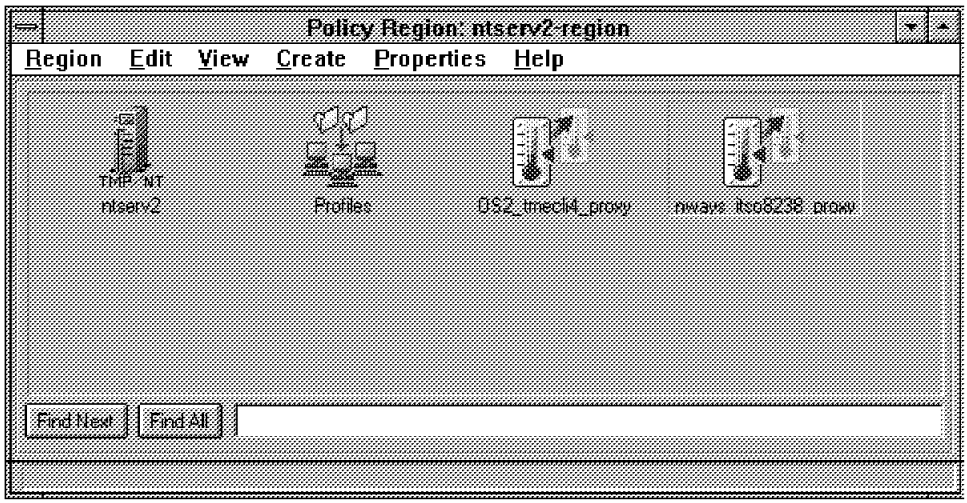


Figure 298. Sentry Example - Creating 8238 Hub Proxy

The proxy is configured as before, defining the endpoint name, and allowing all the monitors via the monitor filter function. We can now distribute the profile to the Nways endpoint in the same way as we distributed the OS/2 client.

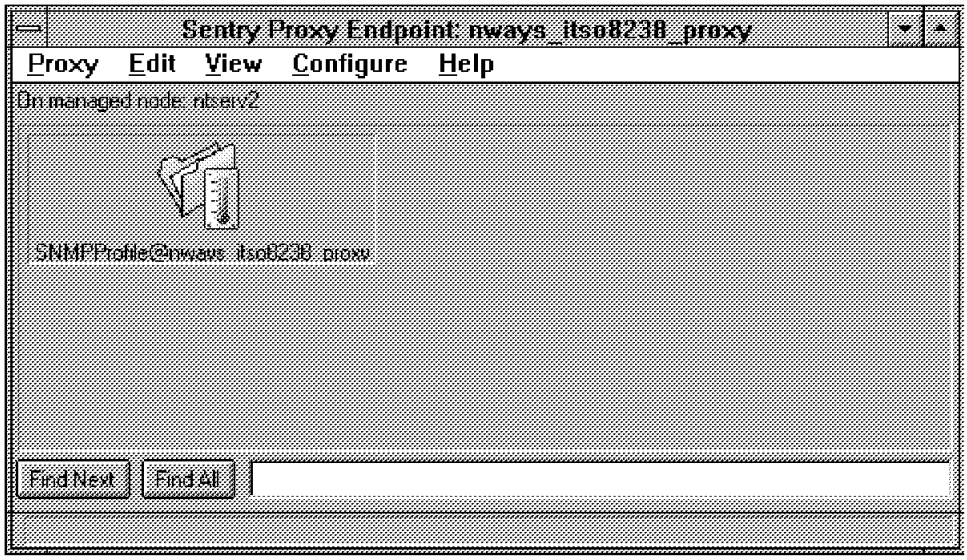


Figure 299. Sentry Example - 8238 Hub Proxy on ntserv2 Managed Node

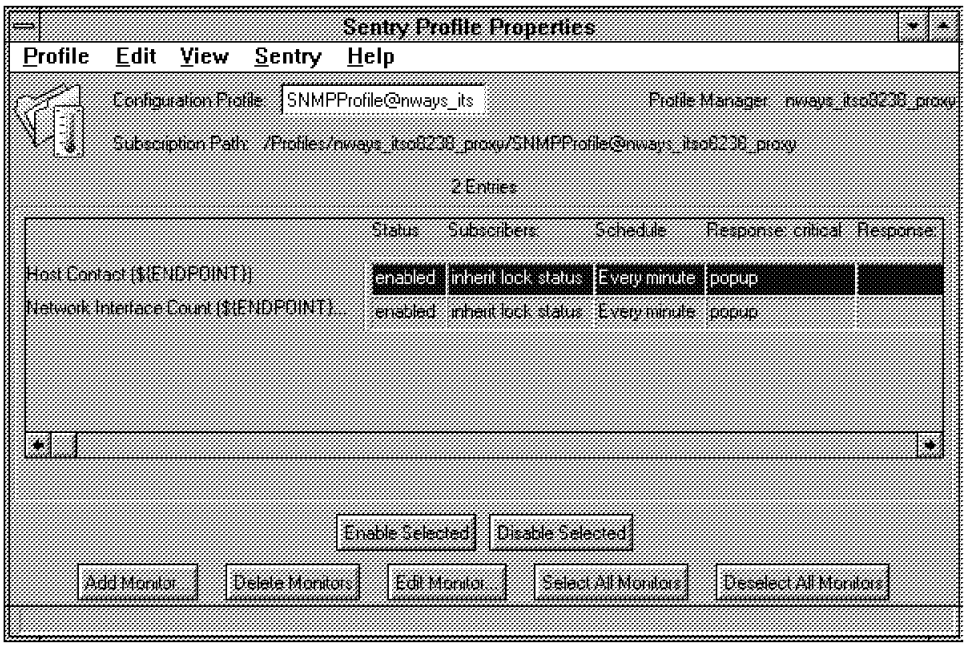


Figure 300. Sentry Example - 8238 Hub Proxy Sentry Properties

For this device we have used the same Sentry monitors as for the OS/2 example. This time we receive alerts from both monitors as the wsnmpget.exe from the wsnmpget.sh file is successful on both counts as the MIB for this device is more complete.

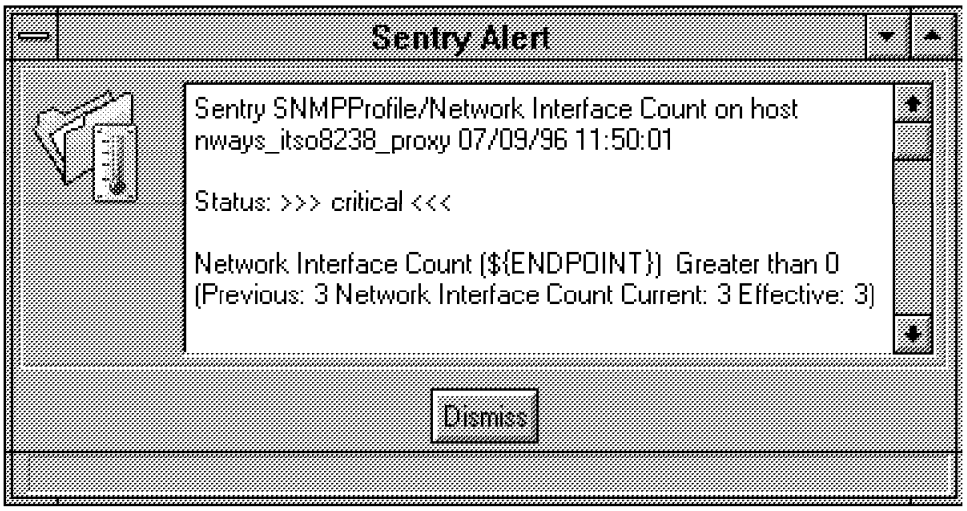


Figure 301. Sentry Example - 8238 Hub Proxy Sentry Monitor 1

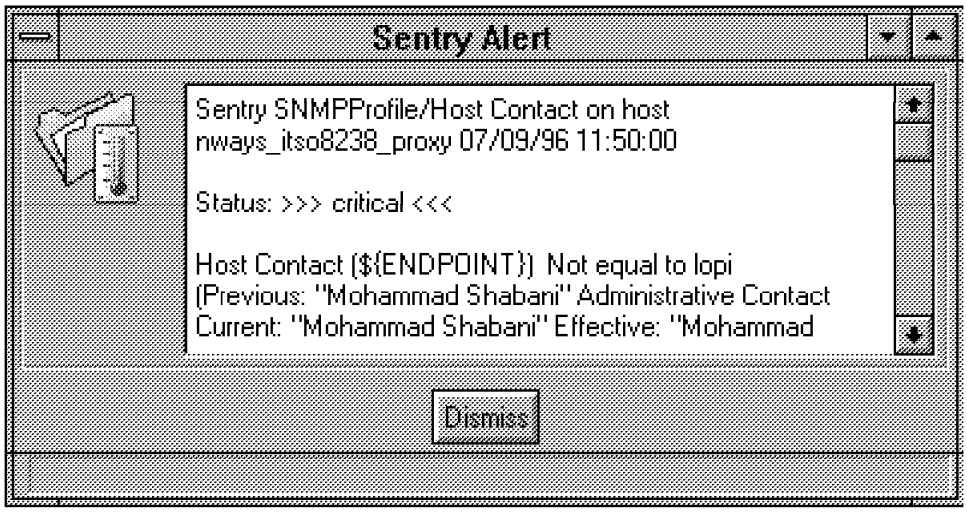


Figure 302. Sentry Example - 8238 Hub Proxy Sentry Monitor 2

Chapter 8. Working with Tasks, Jobs and the Scheduler

This chapter begins to take a look at the scheduler functions to investigate automation. We also take a look at the TME resource called the task library, and its associated components, tasks and jobs. We define their roles as well as show examples of their use.

8.1 The Task Library

The task library acts as a central repository for defining tasks and jobs, which constitute activities that occur daily in heterogeneous networks. These activities could be system backups, printer management or jobs that have to be run or scheduled at certain times. This schedule can be set to occur in units of minutes, hours, days, weeks or years.

The tasks and jobs that are created within the task library can be configured to run on one or more managed resources within the network, independent of the operating system platform in use. These task libraries can exist alone or as a group within one or many Policy Regions.

From within the task library, tasks and jobs can be created, edited and executed. Each task and job within the task library has parameters available for configuration. These tasks and jobs are what are used to populate the task library.

8.1.1 Task Library Policy

Task libraries use default and validation policies as a control mechanism for tasks and jobs residing in the library. These default and validation policies are similar to those used before in policy management for managed resources and profiles. A default policy for the task library is used when creating the task library. Validation policy is used for ensuring the task library meets certain constraints laid down by the policy.

Within task library policy management, the default policies define lists of possible task endpoints and Profile Managers on which the task can be run. A task endpoint being, for example, a ManagedNode, PcManagedNode or SentryProxy Endpoint. These lists are used to select the places that the task or job will run. The validation policies for the task library will control the creation and execution of the tasks and jobs.

The first default policy used is tl_def_dist_mode and defines where the task binaries are distributed upon task creation. The possible options available are:

- ALI - Copied to the TME server only (default)
- LOCAL - Copied to all file servers in the local TMR
- GLOBAL - Copied to all file servers in all connected TMRs

```
ALI
```

Figure 303. Task Library Policy - Default Policy tl_def_dist_mode

Next, we have a default/validation policy pair for Managed Nodes. The purpose of these are to provide a list of task endpoints (places where the task can be sent) where the task or job can be executed, and also to validate these endpoints when selected as execution targets.

```
#!/bin/sh
library=wlookup Library
supporters=idlcall $library select_instance_managers
 '\"TMF_CCMS::ProfileEndpoint\"'
FALSE FALSE FALSE

num_sup=idlarg 1 $supporters
limit=expr $num_sup + 1
i=2
while [ $i -le $limit ]; do
 arg=`idlarg $i $supporters`
 name=`idlarg 2 $arg`
 name=`echo $name | tr -d \"`
 wlookup -r $name -a 2> /dev/null |
  awk -F' ' '{
   for (i = 1; i < NF; ++i) {
    printf "%s (%s)", $i, "'"$name"'";
    if (i < NF-1)
     printf "\t";
   }
   printf "\n";
  }'
 i=`expr $i + 1`
done
```

Figure 304. Task Library Policy - Default Policy tl_def_man_nodes

```
#!/bin/sh
################################################
#
# $Id: tl_val_man_nodes.sh,v 1.2.6.1 1996/04/17 04:36:57 tulloh Exp $
#
# This script implements the "validate_execution_managed_nodes"
# policy method for the task library.  The script is provided with the
# name of task, the label of the Admin and all of the Managed Nodes
# selected for execution targets of the task.  Modify the code below
# if you want something different returned.
#
# To debug your changes you could add the lines:
#
#  set -xv
#  exec > `wtemp`/debug.output 2>&1
#
# These lines will allow you to see any errors that occur by looking
# in the `wtemp`/debug.output file.
#
#  NOTE:   This script can also be called when a check_policy
#  operation is performed.  In that case, the name of the
#  Admin will be "any".  Make sure that you handle that
#  case if you modify this script.
#
################################################

task_name=$1
administrator=$2
shift 2

## Example of how to validate the list of Managed Nodes. ##

# for i in $*; do
#  if [ $i = "the evil managed node" ]; then
#   echo FALSE
#  exit 0
# fi
# done

echo TRUE
exit 0
```

Figure 305. Task Library Policy - Validation Policy tl_val_man_nodes

Next we have a default/validation policy pair for profile managers. The purpose of these are to provide a list of default profile managers where the task or job can be sent for execution, and also to validate these endpoints when selected as execution targets.

```
#!/bin/sh
library=`wlookup Library`
supporters=`idlcall $library select_instance_managers '\"
TMF_CCMS::ProfileManager\"'

num_sup=`idlarg 1 $supporters`
limit=`expr $num_sup + 1`
i=2
while [ $i -le $limit ]; do
 arg=`idlarg $i $supporters`
 name=`idlarg 2 $arg`
 name=`echo $name | tr -d \"`
 wlookup -r $name -a 2> /dev/null |
  awk -F' ' '{
   for (i = 1; i < NF; ++i) {
    printf "%s (%s)", $i, "'"$name"'";
    if (i < NF-1)
     printf "\t";
   }
   printf "\n";
  }'
 i=`expr $i + 1`
done
```

Figure 306. Task Library Policy - Default Policy tl_def_prof_mgrs

```
#!/bin/sh
############################################################
#
# $Id: tl_val_prof_mgrs.sh,v 1.2.6.1 1996/04/17 04:36:58 tulloh Exp $
#
# This script implements the "validate_execution_profile_managers" policy
# method for the task library.  The script is provided with the name of t
# task, the label of the Admin and all of the profile managers selected f
# execution targets of the task.  Modify the code below
# if you want something different returned.
#
# To debug your changes you could add the lines:
#
#  set -xv
#  exec > `wtemp`/debug.output 2>&1
#
# These lines will allow you to see any errors that occur by looking
# in the `wtemp`/debug.output file.
#
#  NOTE:        This script can also be called when a check_policy
#               operation is performed.  In that case, the name of the
#               Admin will be "any".  Make sure that you handle that case
#               if you modify this script.
#
#
############################################################

task_name=$1
administrator=$2
shift 2

## Example of how to validate the list of profile managers. ##

# for i in $*; do
#  if [ $i = "the evil profile manager" ]; then
#   echo FALSE
#  exit 0
# fi
# done

echo TRUE
exit 0
```

Figure 307. Task Library Policy - Validation Policy tl_val_prof_mgrs

The previous four default and validation policies are used to manipulate lists of places when selecting to execute a TME *task* or a TME *job*. These are available as execution targets.

The next four default and validation policies are geared to user IDs and group IDs. These user IDs must be valid across the different TMRs where the task will be performed, otherwise the task cannot be created. There is also a restriction on creating jobs under the root user ID. This was treated as a security exposure by Tivoli.

The first pair relate to default and validation of the task. The user ID creating the task must be able to execute the task on all the platforms chosen. The default is *. This translates into the ID of the current TME administrator.

For example, if the user ID used to create the task is meldun and this is valid under the NT platform, but not valid under the AIX V4 platform where the task will be executed, then the validation policy returns an error, and the task creation will not continue. If, however, the task is only to be run on NT platforms, the task will pass the validation policy, as the user ID is valid.

```
#!/bin/sh
############################################################
#
# $Id: tl_val_set_uid.sh,v 1.1.6.3 1996/04/17 20:35:29 paul Exp $
#
# This script implements the "validate_set_uid" policy
# method for the task library.  The script is provided with
# the label of the Admin value the user chose for the user id of the new
# task.  Modify the code below
# if you want something different returned.
#
# To debug your changes you could add the lines:
#
#  set -xv
#  exec > `wtemp`/debug.output 2>&1
#
# These lines will allow you to see any errors that occur by looking
# in the `wtemp`/debug.output file.
#
############################################################

administrator=$1
user_id=$2
shift 2


## Example of how to validate a uid ##
#
# if [ $administrator != root ]; then
#  if [ $user_id = "root" ]; then
#   echo FALSE
#   exit 0
#  fi
# fi

#
# For the 2.5 release this policy is changed so that no one can create
# tasks that run as root. It was considered a security hole.
#
if [ "$user_id" = "root" ]; then
 echo TRUE
 exit 0
fi

if [ "$user_id" = "Administrator" ]; then
 echo TRUE
 exit 0
fi

if [ "$user_id" = '$root_user' ]; then
 echo TRUE
 exit 0
fi

#
# Temporary work around for problem where a task with nothing for UID
# runs as root.
#
if [ X"$user_id" = X ]; then
 echo TRUE
 exit 0
fi


echo TRUE
exit 0
```

Figure 308. Task Library Policy - Validation Policy tl_val_set_uid

The last two policies are used for group names when creating tasks. The default for the tl_def_set_gid policy is blank, so is omitted here. The tl_val_set_gid, validates the group assigned to a specific task.

```
#!/bin/sh
############################################################
#
# $Id: tl_val_set_gid.sh,v 1.1.6.1 1996/04/17 04:36:58 tulloh Exp $
#
# This script implements the "validate_set_gid" policy
# method for the task library.  The script is provided with
# the label of the Admin and the value the user choose for the group id
# of the new task.  Modify the code below
# if you want something different returned.
#
# To debug your changes you could add the lines:
#
#  set -xv
#  exec > `wtemp`/debug.output 2>&1
#
# These lines will allow you to see any errors that occur by looking
# in the `wtemp`/debug.output file.
#
############################################################

administrator=$1
group_id=$2
shift 2

## Example of how to validate a gid ##
#
# if [ $administrator != root ]; then
#  if [ $group_id = "bin" ]; then
#   echo FALSE
#   exit 0
#  fi
# fi

echo TRUE
exit 0
```

Figure 309. Task Library Policy - Validation Policy tl_val_set_gid

8.1.2 Creating a Task Library

The first stage in creating Task Libraries is to ensure the managed resource type task library is currently set for your Policy Region in which the task library will exist. This is achieved by making sure that the resource, task library, is in the list of current resources. If it's not, you can move it there from the Set Managed Resources window by selecting **Properties** and **Managed Resources** from the menu within the Policy Region and moving it from Available Resources to Current Resources.

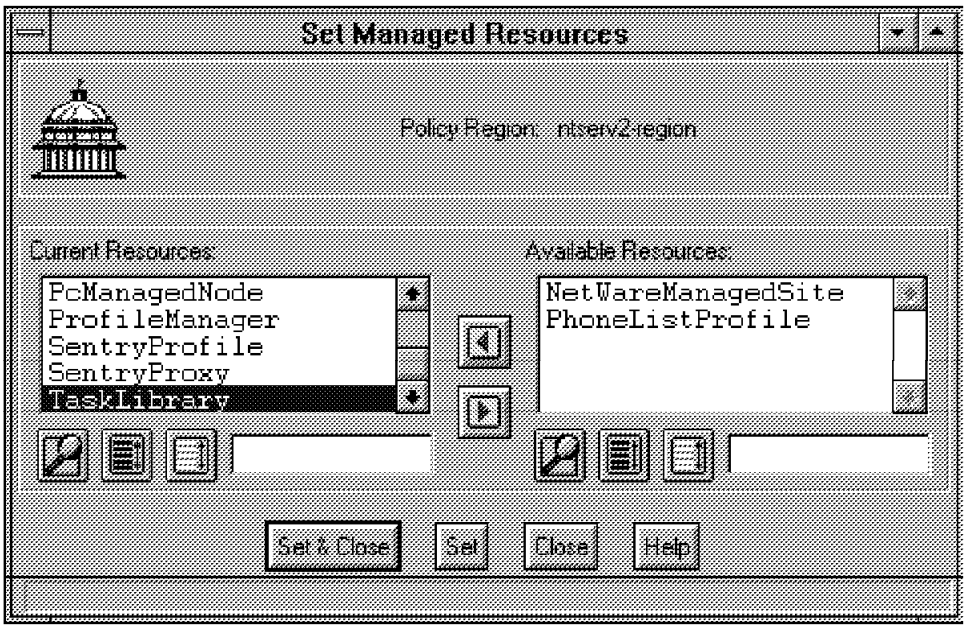


Figure 310. Task Library - Setting Task Library As a Managed Resource

When this is set correctly, we can create a task library, by selecting **Create** and **Task Library** for that Policy Region.

Figure 311. Task Library - Creating a Task Library

We now have a task library defined, although not configured, within our Policy Region.

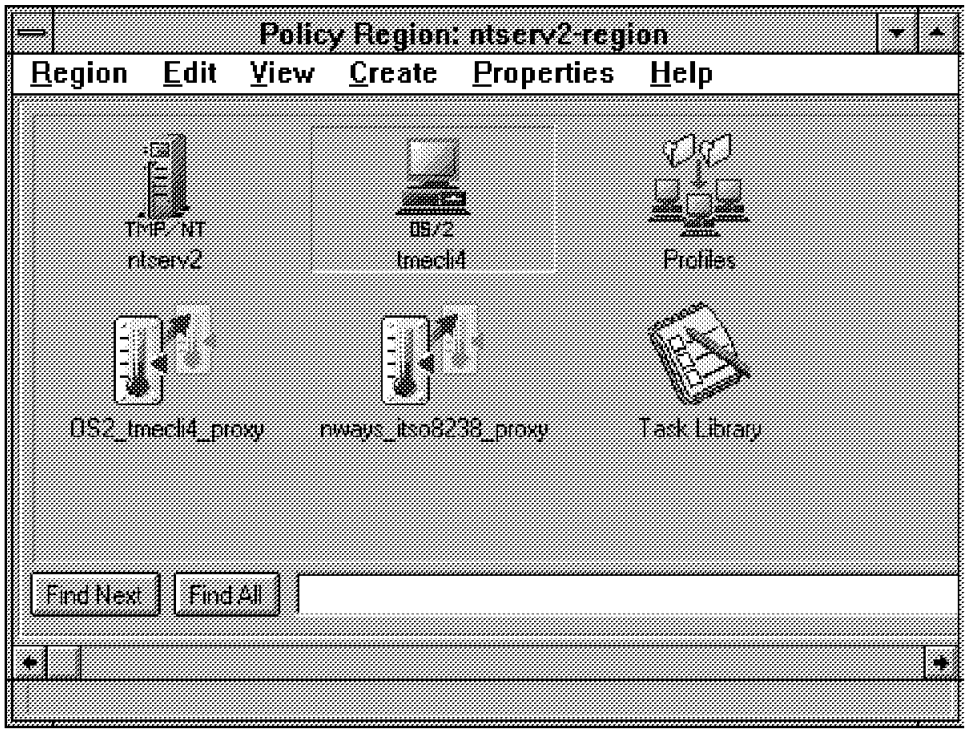


Figure 312. Task Library - Created Task Library

When we open this task library, either by double-clicking or using the right mouse button and selecting **Open**, we can see the empty task library.

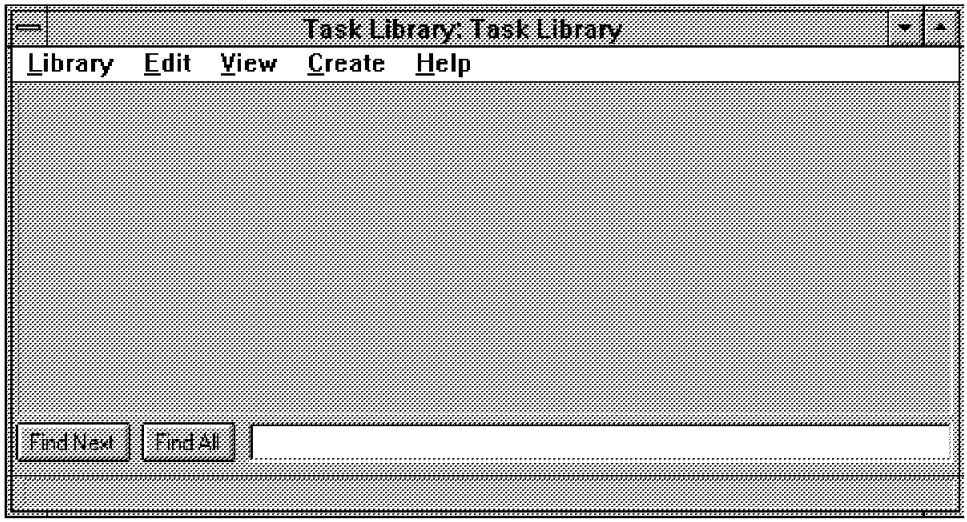


Figure 313. Task Library - Opened Unpopulated Task Library

The options available from the Task Library icon using the right mouse button are:

- Open
- Create Task
- Create Job
- Execute Task

- Execute Job

These options are covered in the following sections.

8.2 Tasks

A task is a typical routine that is performed throughout many network environments. It can also be a TME application used as a customized action within an organization to run on different platforms, but hiding the complexity of the job to users. For example, starting the backup of a NetWare Server, a HP-UX Server, an AIX mksysb and an NT Server backup could all be defined under a task called server backups. Each of these backups are initiated differently, and more often than not organizations have a NetWare specialist, an NT specialist and a UNIX specialist to accomplish each task.

By defining a task we can run different commands on different platforms and send the commands to whichever managed hosts we care to. When defining tasks, there are a number of fields to be tailored within the Create Task section.

8.2.1 Creating Tasks

Tasks are created from within the task library. Creating tasks can be achieved by selecting **Create** and **Task**. Each task that is created requires a name and has certain properties to be completed. These properties include the platforms on which the platform will run, the roles required to execute the task, the user and group IDs under which the task will run and any comments concerning the task being created.

We follow the creation of a task and a subsequent job, showing the execution of the odadmin Tivoli function and send information to a single file location and display on an administrators desktop. We show this from an NT TME Server and an AIX TME Server.

First we need to create the task within the task library.

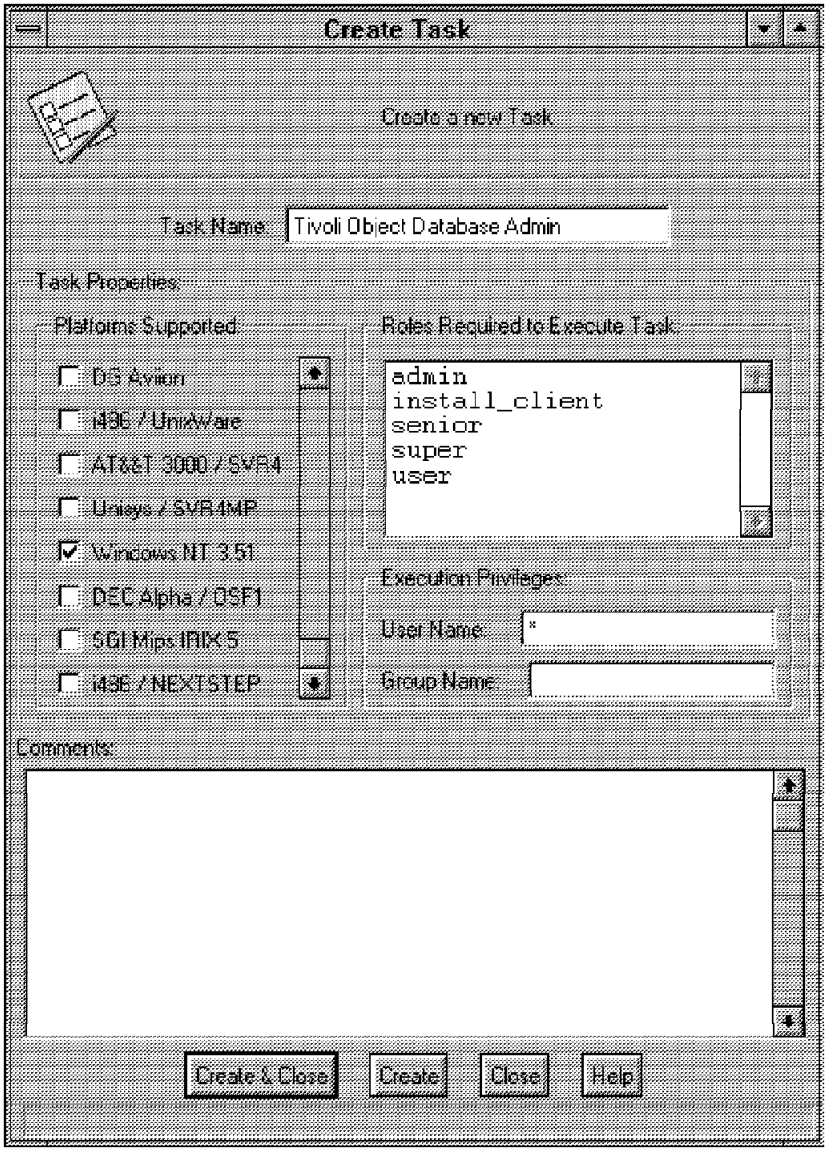


Figure 314. Tasks - Creating a Task (1)

After providing a name for the task, we then define a Windows NT executable for the task. This includes the host location and the absolute path to the executable.

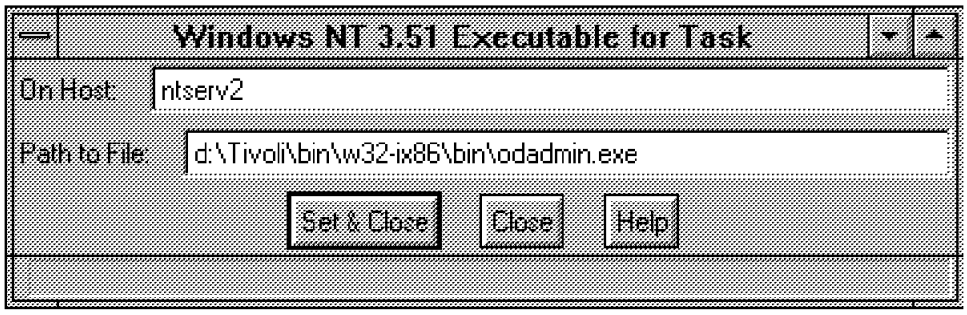


Figure 315. Tasks - Define an NT Executable for the Task

After the executable for this platform is created, we then check the box for the next desired platform, in our case IBM RS/6000 AIX 4.

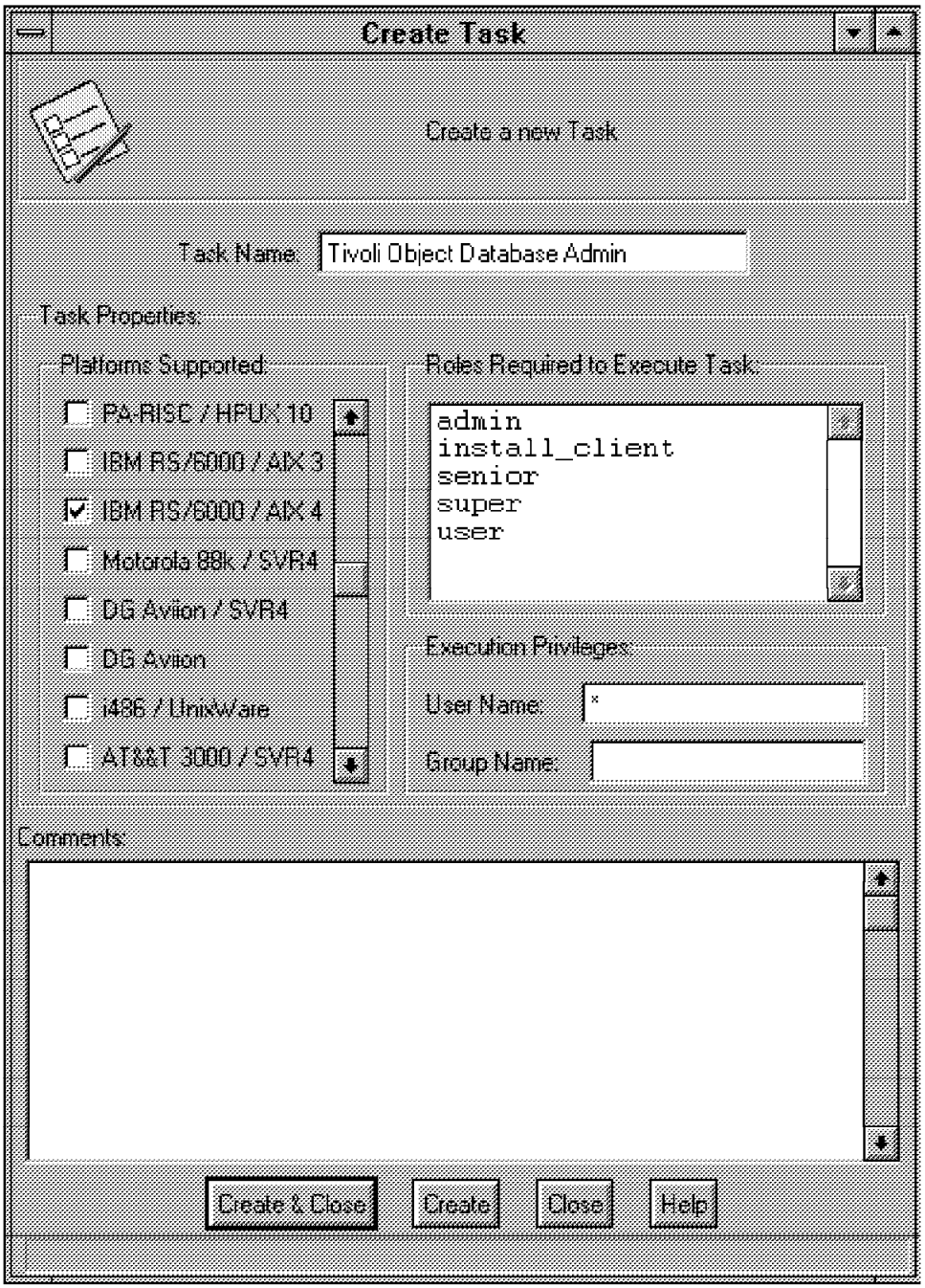


Figure 316. Tasks - Creating a Task (2)

We then point to another platform to run a command that will provide the same function. In this case, we used AIX V4.1.4 and we also chose the odadmin command. We could have picked any command or script to run but we chose to use the odadmin function for our first example.

Note the exact location of the task executable must be entered for that platform, not just the directory path. When a job is created, and other RS6000s are added, the task executable could reside anywhere on similar endpoints (or not exist at all). The task executable defined is run from the TME server. This can be changed by altering the tl_def_dist_mode policy.

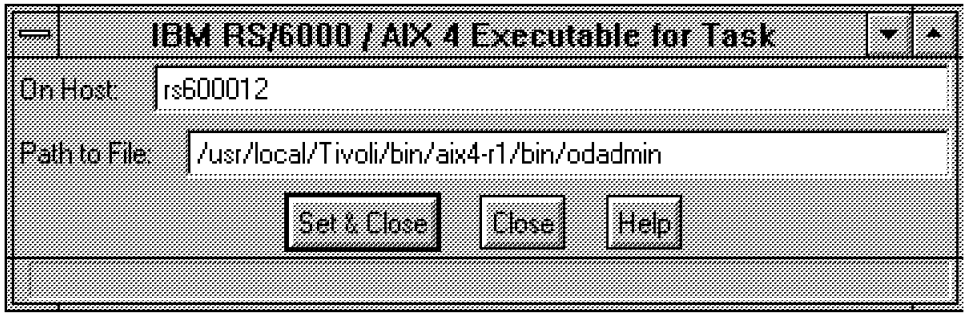


Figure 317. Tasks - Define an AIX Executable for the Task

We now can define the role required to run the task, add comments and alter the user and group ID. Once this has been done, we can create and close the Create Task window.

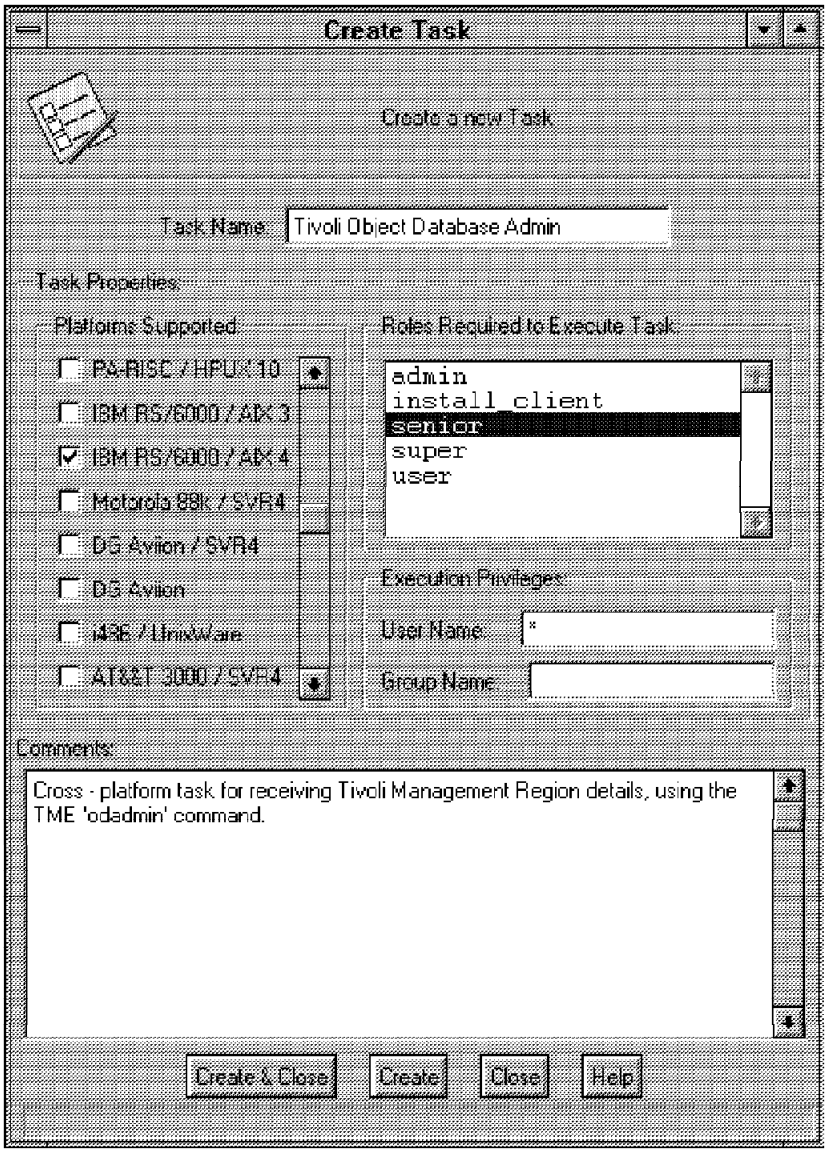


Figure 318. Tasks - Defining Roles, User ID, Group ID and Comments

We have now populated the task library with a task. It will automatically show up in the task library as shown in the following screen.

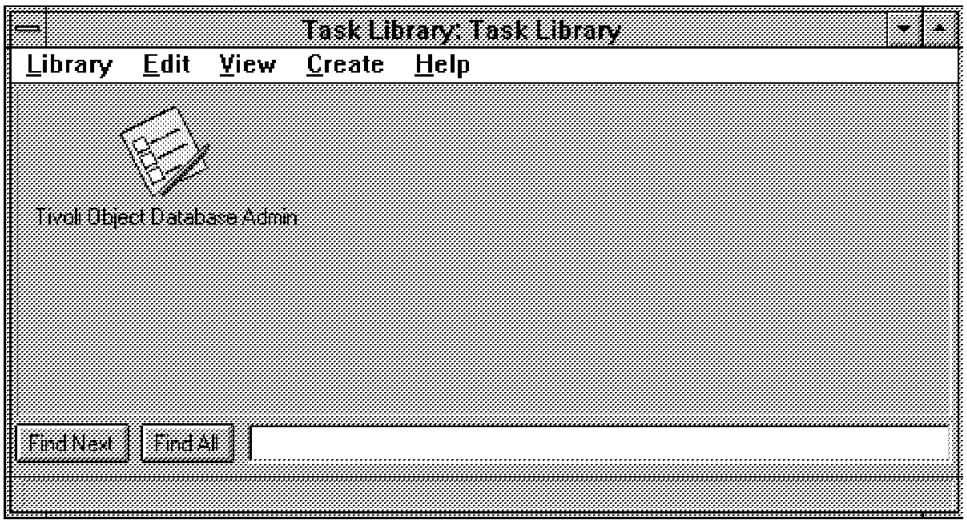


Figure 319. Tasks - Populated Task Library with a Single Task

With the newly created task, we can either edit the task or execute the task. To edit the task you can use the right mouse button. To execute the task, select the **Task** icon with the right mouse button or double-click on the **Task** icon.

The next window displays the options for executing a task. There are two main panels, one for options and one for targets. The options panel consists of:

- Execution Mode
  1. Parallel - All tasks executed in parallel.
  2. Serial - All tasks executed serially in endpoint alphabetical order.
  3. Staged - All tasks staged in groups (see Execution Parameters).
- Execution Parameters
  1. Timeout - Amount of time the task will wait before returning results
  2. Staging Count - Used when the mode is Staged, number of endpoints.
  3. Staging Interval - Used when the mode is Staged, group interval.
- Output Format
  1. Header
  2. Return Code
  3. Standard Error
  4. Standard Output
- Output Destination
  1. Display on Desktop - Displays a window on the desktop.
  2. Save to File - Saves the output to a file on a selected host.

The next window shows the customized version of the task we just described. We chose to:

- Execute the task in parallel on task endpoints ntserv2 and rs600012.

- Set a timeout value of 60 seconds.
- Display all possible fields in the output.
- Display the output in a file and on the desktop.

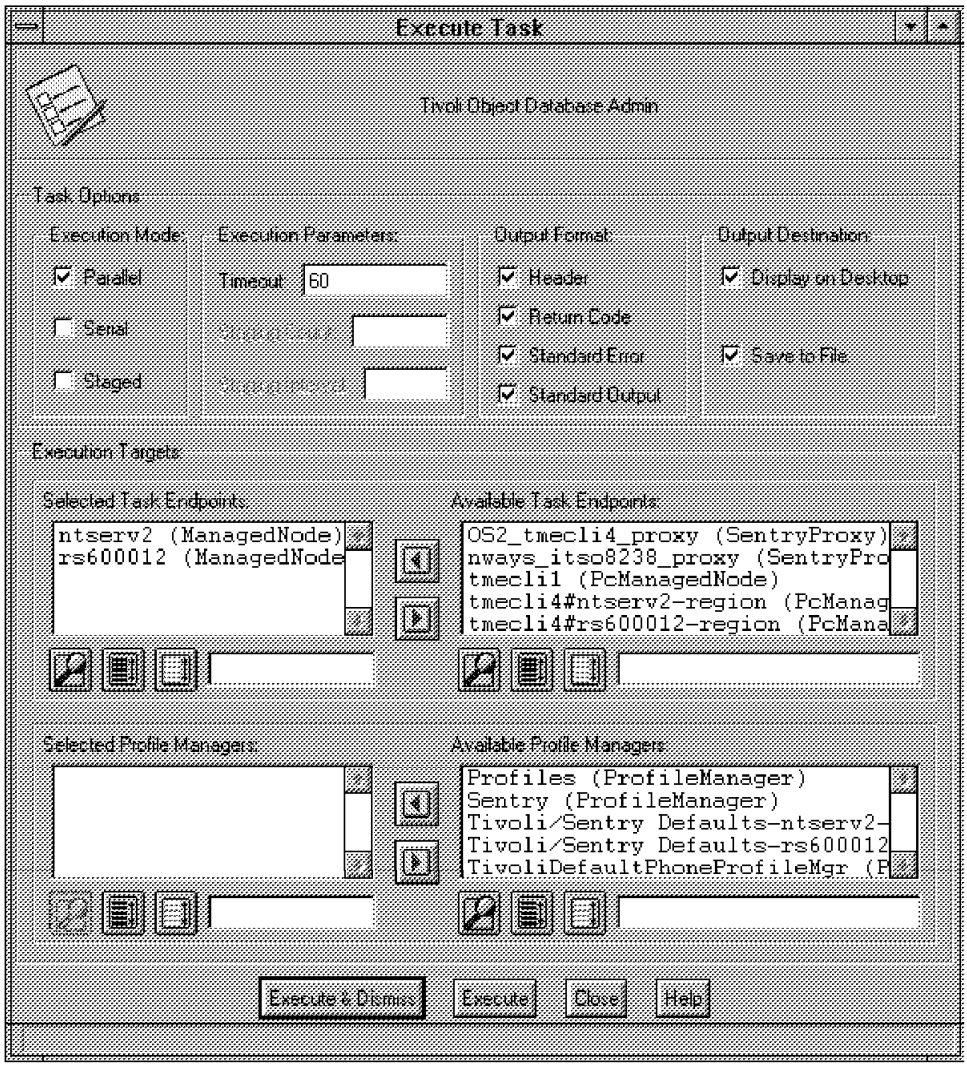


Figure 320. Tasks - Execute Defined Task

When the output destination box for saving to a file is checked, the following window is displayed asking for a host and file location to output the results to.

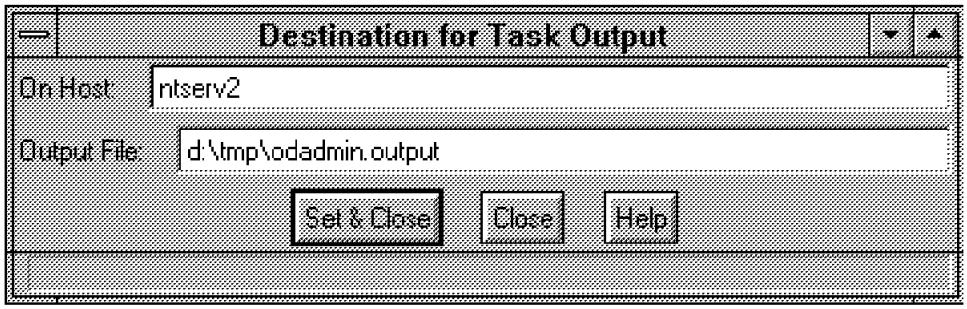


Figure 321. Tasks - File Destination for Output

As we have chosen to send the output to the desktop and a file, examples of both are now shown.


```
################################################################################
Task Name:      Tivoli Object Database Admin
Task Endpoint:  ntserv2 (ManagedNode)
Return Code:    0
------Standard Output------
Region = 1606608427
Dispatcher = 1
Interpreter type = w32-ix86
Database directory = D:\Tivoli\db\ntserv2.db
Install directory = D:\Tivoli\bin
Inter-dispatcher encryption level = simple
Kerberos in use = FALSE
Remote client login allowed = TRUE
Tivoli Management Framework Rev 2 () #1 Thu May  9 18:32:33  1996
Copyright (c) 1990-1995 by Tivoli Systems, Inc.

State flags in use = TRUE
State checking in use = TRUE
State checking every 180 seconds
------Standard Error Output------
################################################################################
Task Name:      Tivoli Object Database Admin
Task Endpoint:  rs600012 (ManagedNode)
Return Code:    0
------Standard Output------
Region = 1899640400
Dispatcher = 1
Interpreter type = aix4-r1
Database directory = /var/spool/Tivoli/rs600012.db
Install directory = /usr/local/Tivoli/bin
Inter-dispatcher encryption level = simple
Kerberos in use = FALSE
Remote client login allowed = TRUE
Install library path =
/usr/local/Tivoli/lib/aix4-r1:/usr/lib:/shadow/iblib/aix4-r1:/usr/lib:/usr/local/Tivoli/lib/aix4-r1:/usr/lib
Tivoli Management Framework Rev 2 (tmpbuild) #1 Fri Apr 19 11:20:36 CDT 1996
Copyright (c) 1990-1995 by Tivoli Systems, Inc.  All Rights Reserved.

State flags in use = TRUE
State checking in use = TRUE
State checking every 180 seconds
------Standard Error Output------
################################################################################
```

Figure 322. Tasks - Output Displayed on Desktop

```
############################################################
Task Name: Tivoli Object Database Admin
Task Endpoint: ntserv2 (ManagedNode)
Return Code: 0
------Standard Output------
Region = 1606608427
Dispatcher = 1
Interpreter type = w32-ix86
Database directory = D:\Tivoli\db\ntserv2.db
Install directory = D:\Tivoli\bin
Inter-dispatcher encryption level = simple
Kerberos in use = FALSE
Remote client login allowed = TRUE
Tivoli Management Framework Rev 2 () #1 Thu May  9 18:32:33  1996
Copyright (c) 1990-1995 by Tivoli Systems, Inc.

State flags in use = TRUE
State checking in use = TRUE
State checking every 180 seconds
------Standard Error Output------
Task Name: Tivoli Object Database Admin
Task Endpoint: rs600012 (ManagedNode)
Return Code: 0
------Standard Output------
Region = 1899640400
Dispatcher = 1
Interpreter type = aix4-r1
Database directory = /var/spool/Tivoli/rs600012.db
Install directory = /usr/local/Tivoli/bin
Inter-dispatcher encryption level = simple
Kerberos in use = FALSE
Remote client login allowed = TRUE
Install library path = /usr/local/Tivoli/lib/aix4-r1:/usr/lib:
/shadow/iblib/aix4-r1:/usr/lib:/usr/local/Tivoli/lib/aix4-r1:/usr/lib
Tivoli Management Framework Rev 2
(tmpbuild) #1 Fri Apr 19 11:20:36 CDT 1996
Copyright (c) 1990-1995 by Tivoli Systems, Inc.  All Rights Reserved.

State flags in use = TRUE
State checking in use = TRUE
State checking every 180 seconds
------Standard Error Output------
```

Figure 323. Tasks - Output to a File

You should notice in the output from either of the preceding figures that the Region number was different, thus showing that it did execute on the different regions. In addition, you can see that the install library paths are different for each platform.

8.3 Jobs

A TME job cannot exist before a task is created. A job is a task that is executed on a number of specific TME resources. Upon job creation, a task is selected to be executed and certain information is defined for that job. Once this job has been created, it can be scheduled to run without providing any further information.

The information required to define a job is the same information that you needed when you executed a task. The job, once defined, can be executed directly, whereas when executing the task these parameters must be passed each time. A list of all the possible options for TME jobs follows:

- Execution Mode
  1. Parallel - All tasks are executed in parallel.
  2. Serial - All tasks are executed serially in *endpoint* alphabetical order.
  3. Staged - All tasks are staged in groups (see Execution Parameters).
- Execution Parameters
  1. Timeout - Amount of time the task will wait before returning results.
  2. Staging Count - Used when the mode is Staged, number of endpoints.
  3. Staging Interval - Used when the mode is Staged, group interval.
- Output Format
  1. Header
  2. Return Code
  3. Standard Error
  4. Standard Output
- Output Destination
  1. Display on Desktop - Displays a window on the desktop.
  2. Save to File - Saves the output to a file on a selected host.

8.3.1 Creating Jobs

Jobs can be created after a task has been created. A job runs a pre-defined task. To create a job from within the task library select **Create** and **Job** from the Task Library menu bar, or using the right mouse button over the Task Library icon within the Policy Region, select **Create Job**.

The next window shows a job being created. It requires a name and an associated task. The previously discussed job information must then be configured, including destination endpoints for the task to be executed. Be sure to click on the **Task Name** even if it is the only task in the list.

When all this is done, the job can be created by selecting the **Create and Close** button.

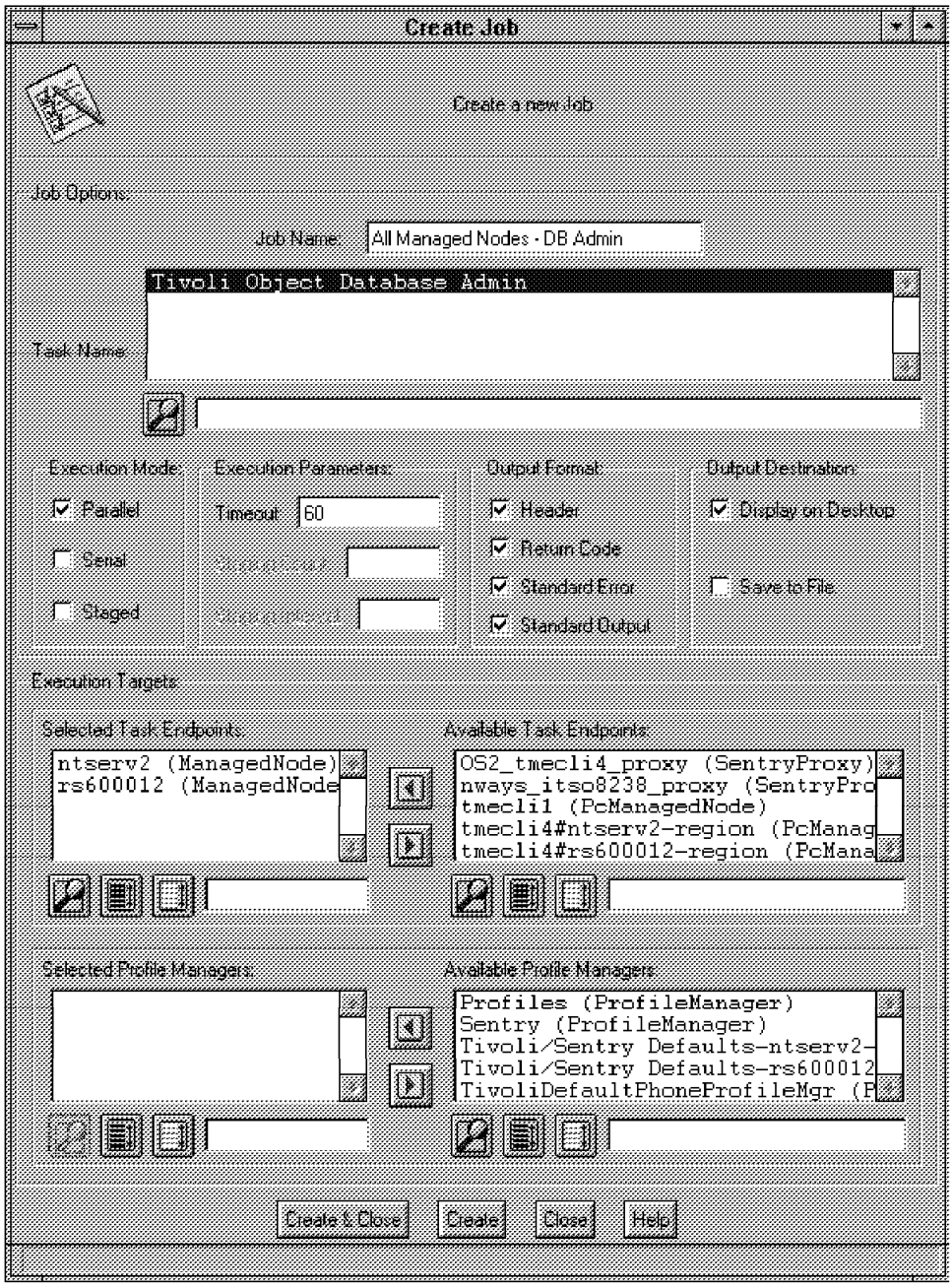


Figure 324. Jobs - Creating a TME Job

Now that the job has been created, the task library contains both a task and a job. As can be seen in the next window, these are represented by two different icons.

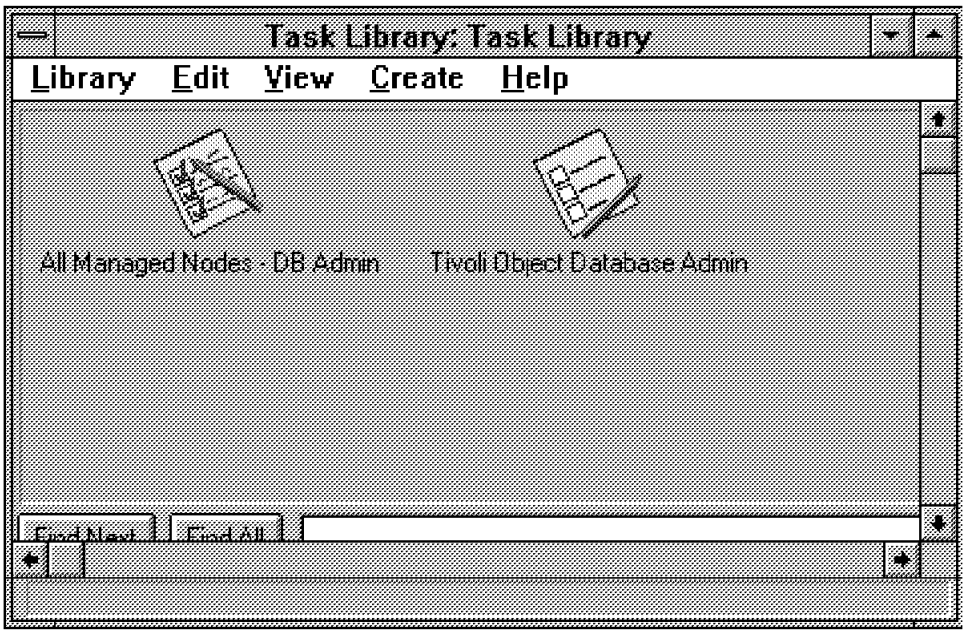


Figure 325. Jobs - Task Library with a Task and a Job

To execute this job, either double-click on it or with the right mouse button select **Execute Job**. Using the right mouse button the job can be altered in the future by selecting **Edit Job**. The following window represents the output from the job after execution. The output was to be sent to the desktop only.

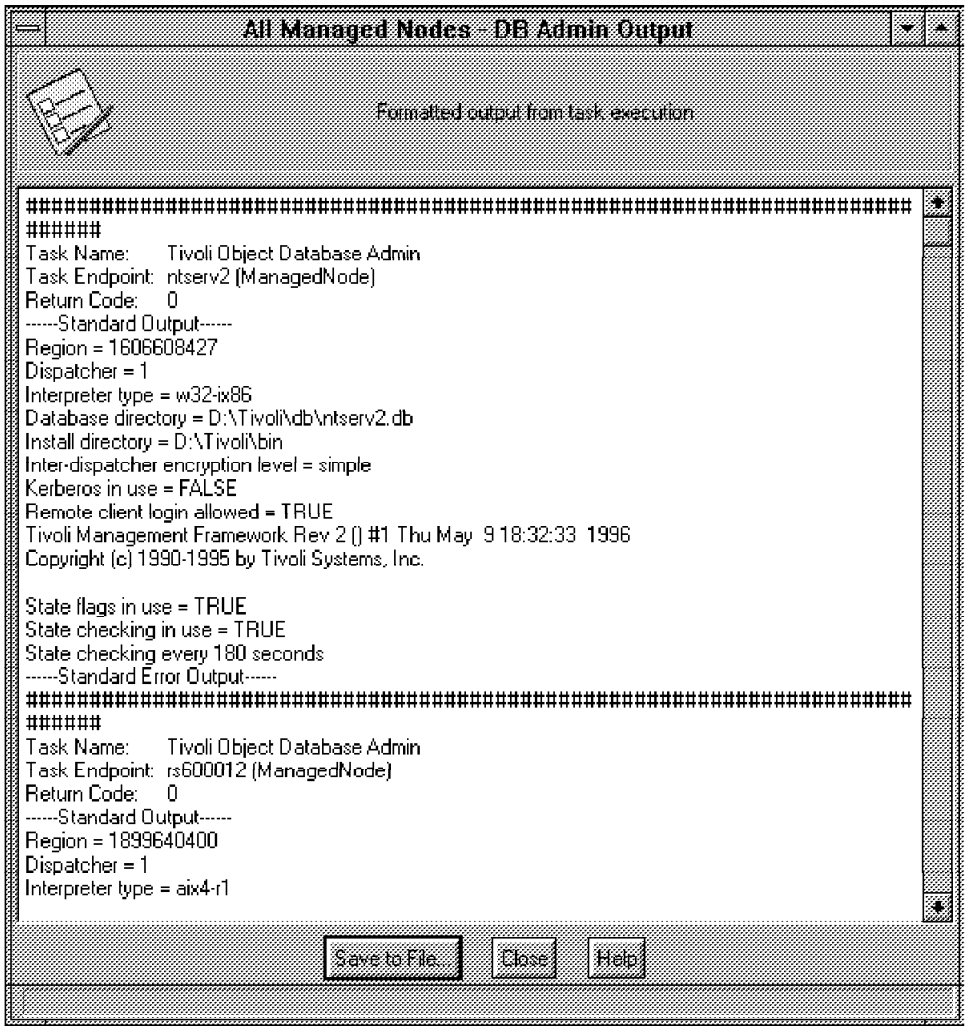


Figure 326. Jobs - Output from the Job after Execution

8.4 The Scheduler

TME provides a mechanism for scheduling jobs to be run. Usually jobs that tie up significant resources, or that need exclusive access to a resource, such as backups, are run off hours. As a job is a defined entity, the scheduler only allows you to create scheduled jobs. It does not allow you to create jobs. A job must be created before it can be added to the scheduler.

The scheduler within TME is a TME service that can be used for scheduling purposes. These purposes include scheduling jobs that will only be run once, recurring jobs or scheduling jobs to run indefinitely or until they succeed. Jobs can be restricted to run at certain times only or to run within a certain time frame. There are also a number of options to choose from to define what to do once the scheduled job has completed.

8.4.1 Scheduler Example

We use the previous job example to show a job being scheduled. As we indicated before, we have to create a job within the task library, before we can schedule it. Then after we have a created job we have to add it to the scheduler by dragging and dropping it on the Scheduler icon. If we double-click on the **Scheduler** icon now, we receive a message that there are no jobs currently scheduled.

The following two windows show the TME desktop and the contents of a task library called Task Library. To add the job All Managed Nodes - DB Admin to the TME Scheduler, select it with the left mouse button and drag and drop it onto the Scheduler on the TME Desktop.

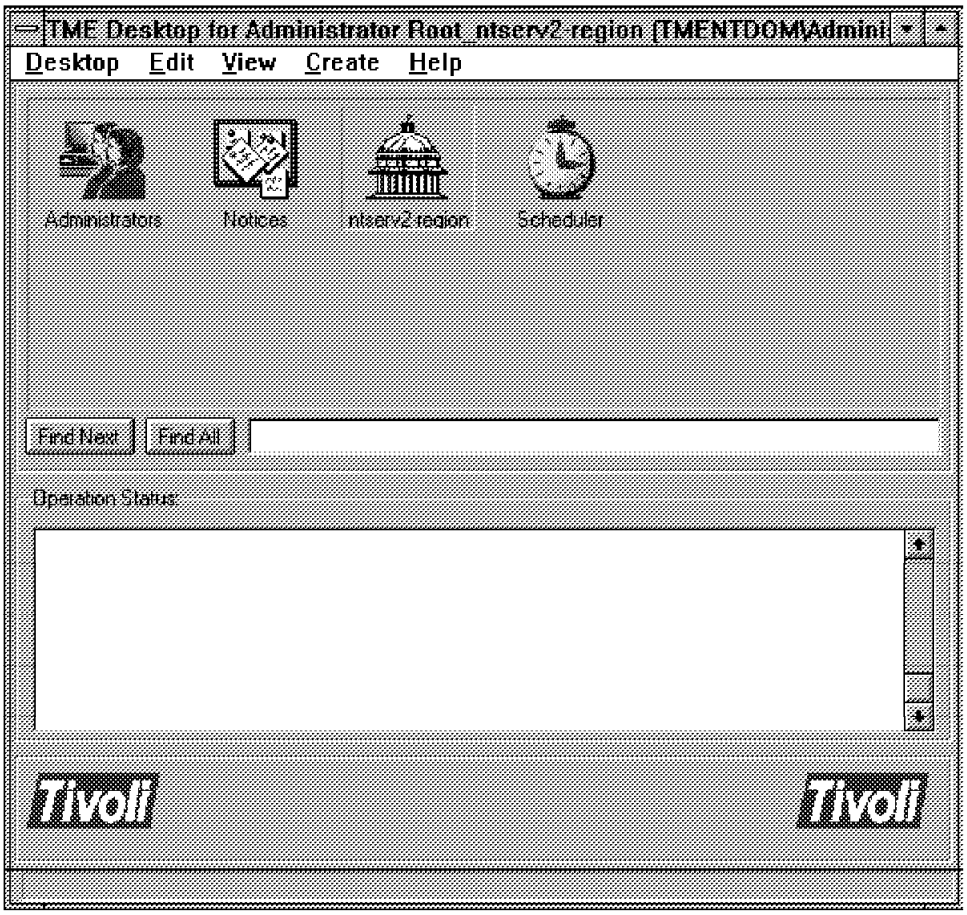


Figure 327. Scheduler - TME Desktop with Scheduler

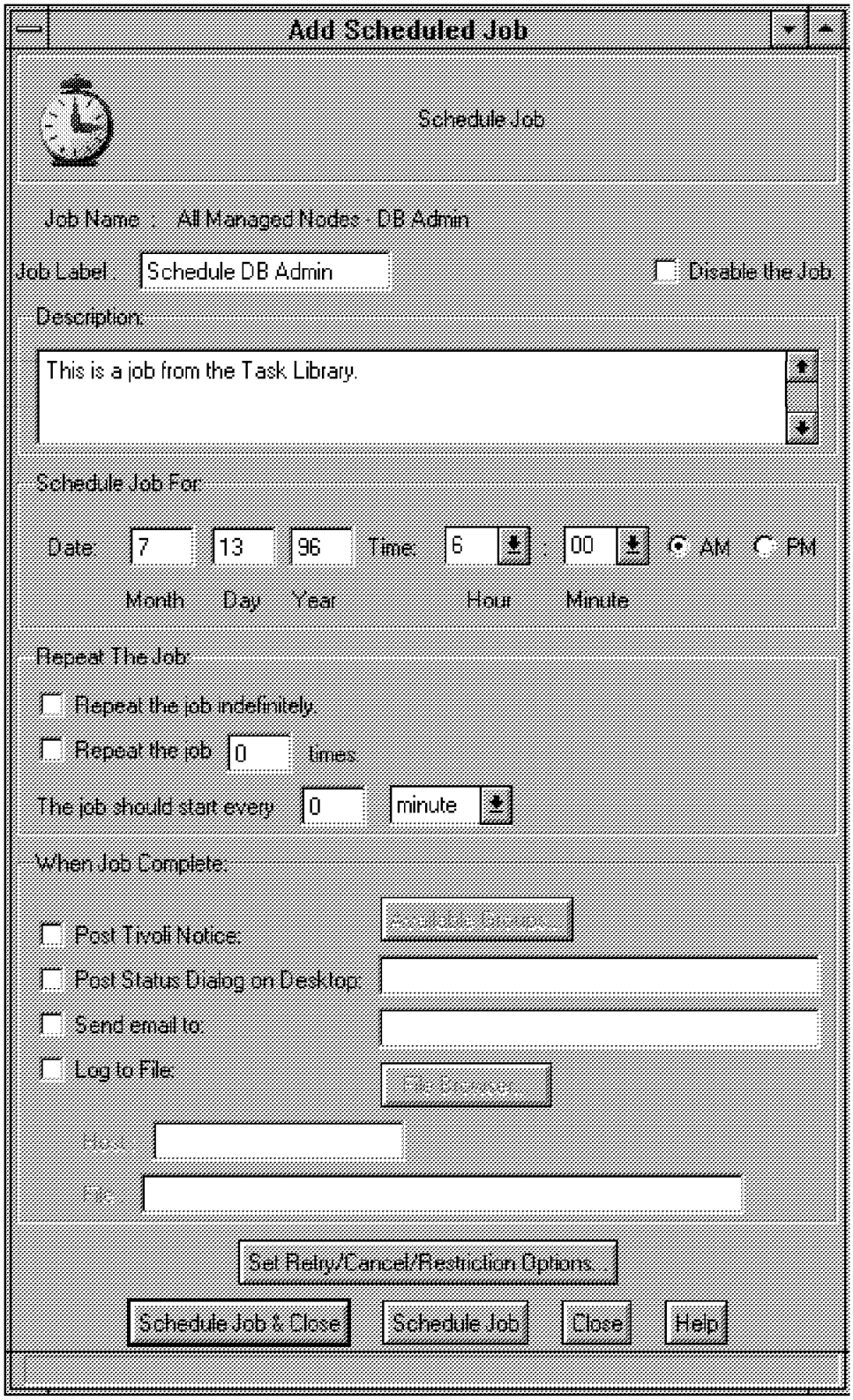


Figure 328. Scheduler - Task Library with Job All Managed Nodes - DB Admin

Once this job has been added to the scheduler, it has to be configured for scheduling. Each schedule can be configured with the following parameters:

- Job Label - The default is the job itself.

- Description - A description of the job.
- Disable - Allows for a scheduled job to be disabled.
- Schedule for:
  - Month, Day, Year
  - Hour, Minute, AM or PM
- Repetition
  - Indefinite
  - How many times the job should repeat
  - At what intervals the job should repeat
- On Completion
  - Post Tivoli Notice - Send a notice to a Tivoli notice group
  - Post Status Dialog on Desktop - Send message to the desktop
  - Send E-Mail
  - Log to a file - Log the output to a file

The following scheduled job is added with some of these options checked. This job will do the following:

- Runs at 12:15pm on the 12th July 1996.
- The job is repeated 3 times at two minute intervals.
- On completion each time it:
  - Posts a Tivoli Notice
  - Logs to a file on host ntserv2 called d:/tmp/sched.output

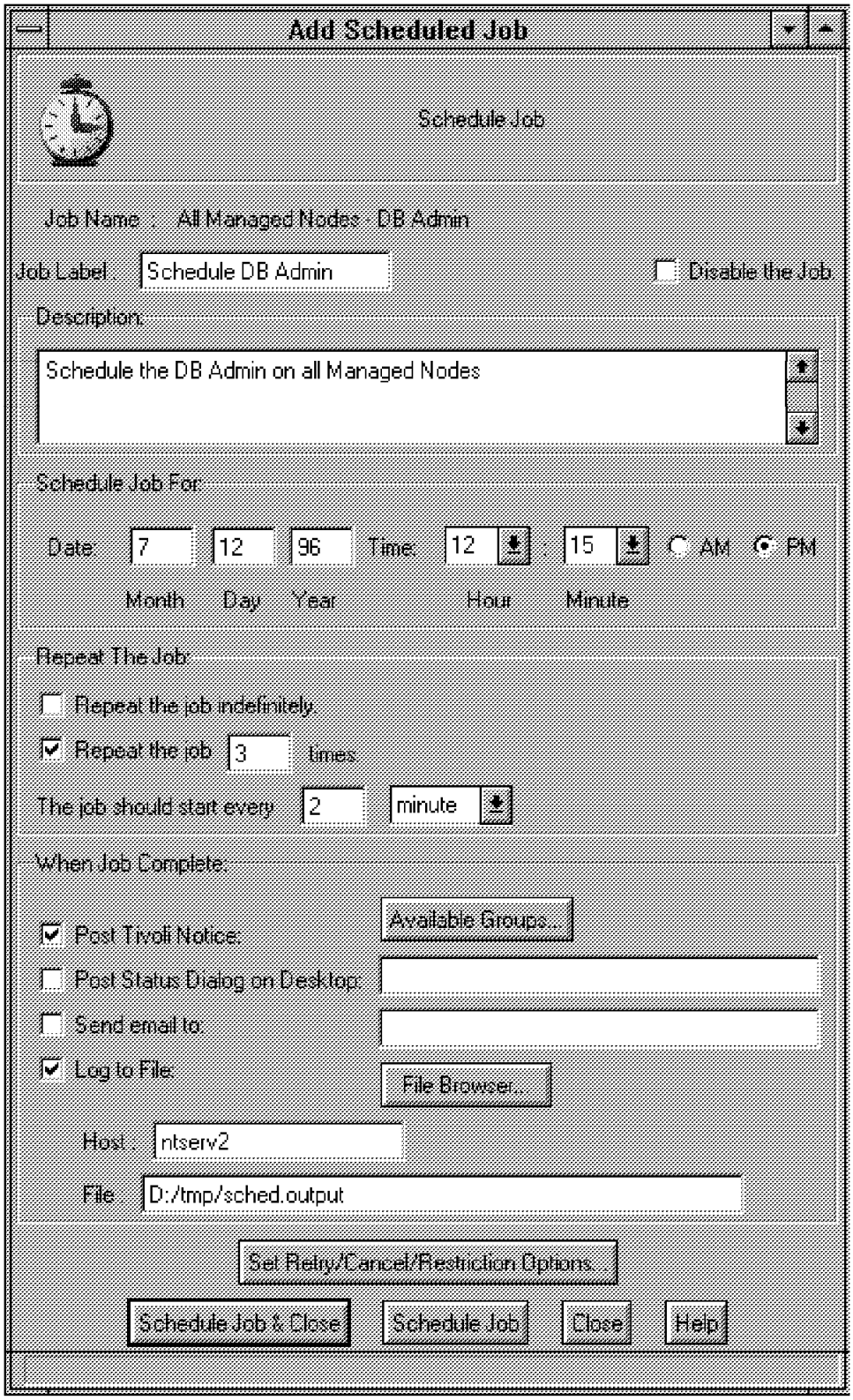


Figure 329. Scheduler - Scheduled Job

There are options that you can select for actions to be taken upon completion of the job. This includes posting a Tivoli notice (see Figure 329 for other options). When this is selected the Available Groups button is activated. Upon selecting this option you will be given a choice of which Tivoli notice group should receive

the message. The following window shows the selection of a valid Tivoli notice group:

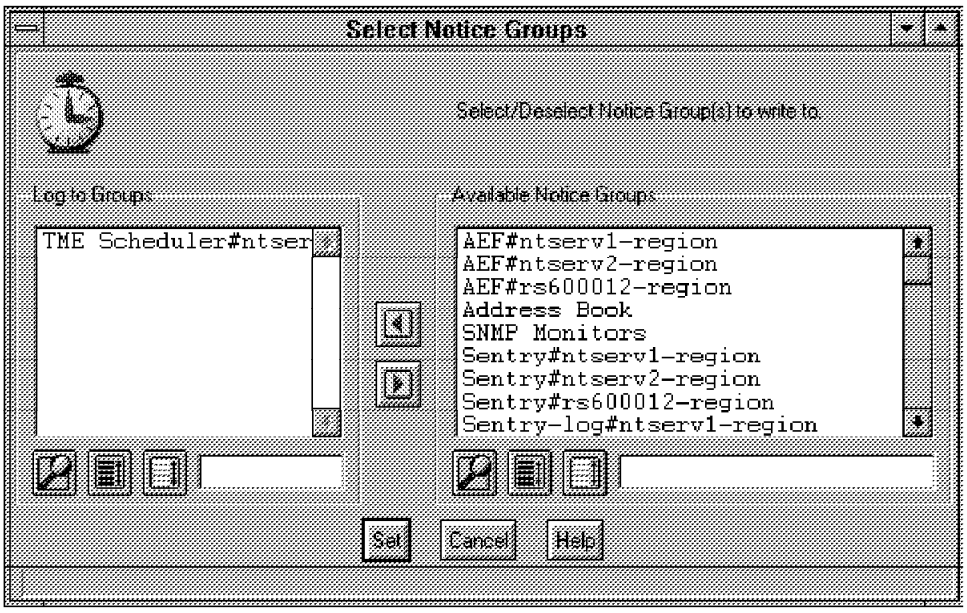


Figure 330. Scheduler - Posting Tivoli Notices on Completion

Also upon completion, the output from this scheduled job is sent to a file on a host. When this box is checked, the File Browser button is activated and when selected the following window is created. From here a file can be selected as the destination for logging purposes. Alternatively a new file can be chosen.

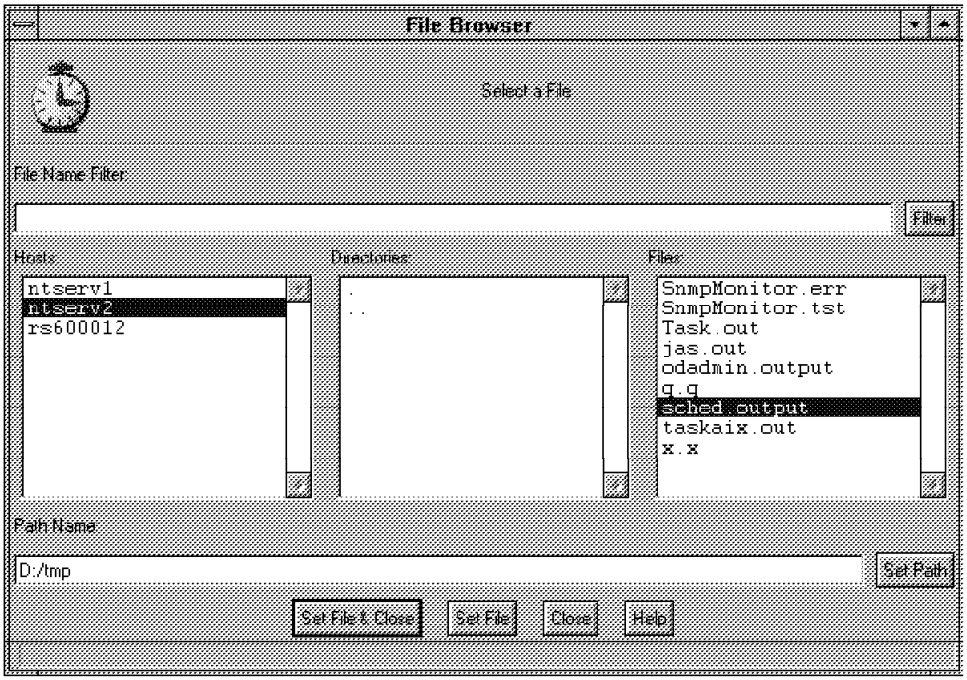


Figure 331. Scheduler - File Browser

There is also an option to set retry and restriction details for the job being scheduled. For this example we have selected to cancel the job if it does not

start within 10 minutes of the scheduled start. The retry values are set to retry 3 times, then log the job as failed, and the retry interval set to every two minutes. We have not set any time restrictions on the job. All these retry and restriction values can be seen in the following figure.

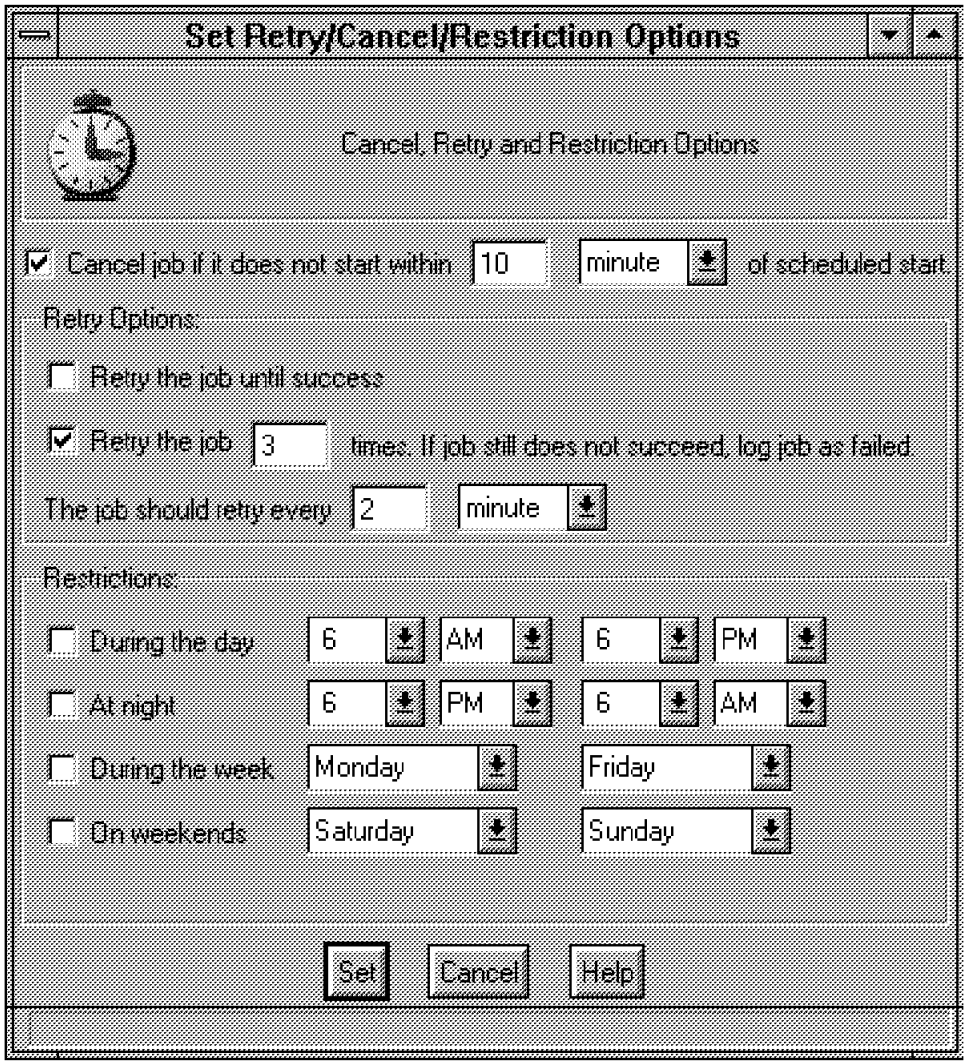


Figure 332. Scheduler - Retry/Cancel/Restriction Options

Finally after creating scheduled jobs, we have the ability to browse those scheduled jobs, by double-clicking on the **Scheduler** icon on the TME desktop. The following window is displayed with each job having a specific job ID. Also from here each job can be dynamically updated or disabled from running by selecting and editing. The following window displays scheduled jobs.

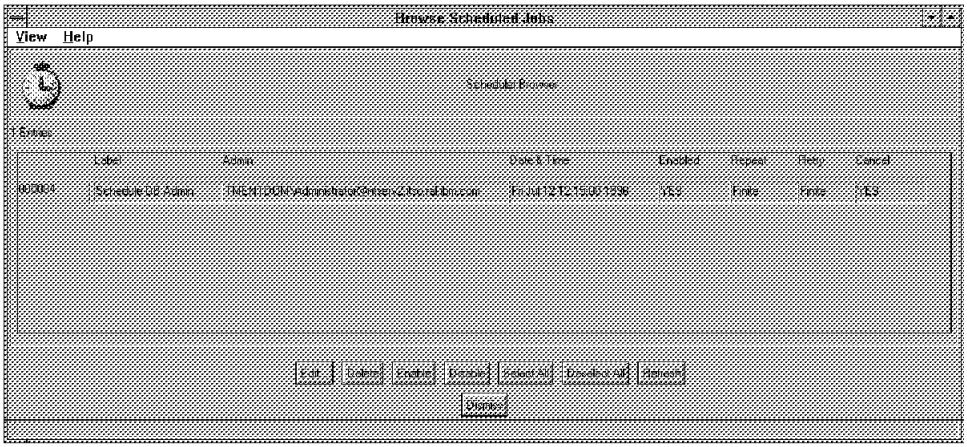


Figure 333. Scheduler - Browsing Scheduled Jobs

The following is the logged output to the file d:\tmp\sched.output as was defined within the scheduled job. As we had chosen to execute the job three times, we have three successful responses logged with time and date stamps.

```
This message was written at Fri Jul 12 12:15:12 1996
Scheduler attempted to execute the following job:

Id          : 4
Name        : All Managed Nodes - DB Admin
Label       : Schedule DB Admin
Description : Schedule the DB Admin on all Managed Nodes

Execution Scheduled Start Time : 07/12/96 12:15:00
Execution Actual Start Time    : 07/12/96 12:15:00
Execution Finish Time          : 07/12/96 12:15:12
Execution of the job completed without error.

This message was written at Fri Jul 12 12:17:11 1996
Scheduler attempted to execute the following job:

Id          : 4
Name        : All Managed Nodes - DB Admin
Label       : Schedule DB Admin
Description : Schedule the DB Admin on all Managed Nodes

Execution Scheduled Start Time : 07/12/96 12:17:00
Execution Actual Start Time    : 07/12/96 12:17:00
Execution Finish Time          : 07/12/96 12:17:11
Execution of the job completed without error.

This message was written at Fri Jul 12 12:19:11 1996
Scheduler attempted to execute the following job:

Id          : 4
Name        : All Managed Nodes - DB Admin
Label       : Schedule DB Admin
Description : Schedule the DB Admin on all Managed Nodes

Execution Scheduled Start Time : 07/12/96 12:19:00
Execution Actual Start Time    : 07/12/96 12:19:00
Execution Finish Time          : 07/12/96 12:19:11
Execution of the job completed without error.
```

Figure 334. Scheduler - Output of Completed Scheduled Job

In addition, information can be placed into the Notification Group on the desktop as shown in the following window:

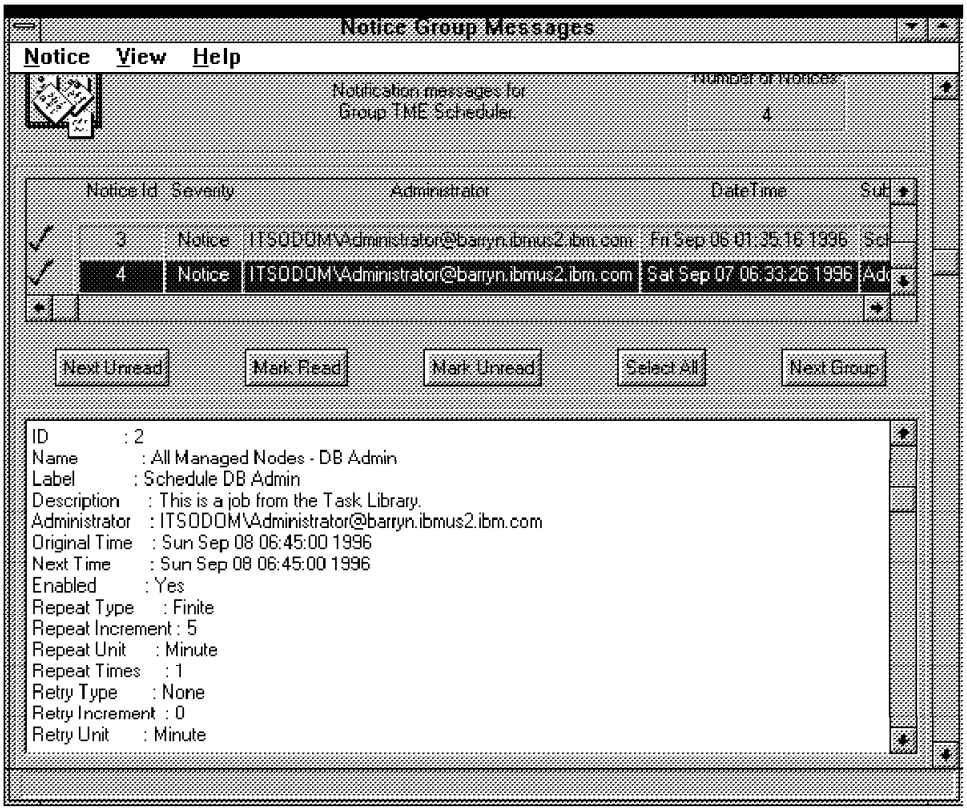


Figure 335. Notification Group on the Desktop

8.5 Command Line Examples

The following gives an insight into the command line usage for task libraries, tasks, jobs and the scheduler. For a more comprehensive coverage please refer to the Tivoli Command Line Reference.

8.5.1 Task Library Commands

The following show how to create and list the contents of a task library.

8.5.1.1 Creating Task Libraries

NAME: wcrttlib

PURPOSE: To create a task library within a Policy Region.

SYNOPSIS:

- wcrttlib *library_name policy_region_name*

EXAMPLE:

```
wcrttlib "Task Library" ntserv2-region
```

Creates a task library called Task Library within the ntserv2-region Policy Region.

8.5.1.2 Listing Task Libraries

NAME: wlstlib

PURPOSE: To list the contents of a task library within the Policy Region.

SYNOPSIS:

- wlstlib *library_name*
- wlstlib -a

EXAMPLE:

```
wlstlib -a
```

Displays all the defined task libraries within the Policy Region. The following shows the output for our Policy Region.

```
Backup Tasks:
SNMP Tasks:
Task library:
(task) Tivoli Object Database Admin
(job) All Managed Nodes - DB Admin
```

Figure 336. Task Library - wlistlib Output

This displays three task libraries and their contents:

- Backup Tasks - Empty
- SNMP Tasks - Empty
- Task library - Containing one task and one job
  - Task - Tivoli Object Database Admin
  - Job - All Managed Nodes - DB Admin

8.5.2 Task Commands

The following show how to create, execute, distribute and list the contents of individual tasks, within task libraries which exist within a Policy Region.

8.5.2.1 Creating Tasks

NAME: wcrttask

PURPOSE: To create a task within a task library.

SYNOPSIS:

- wcrttask -t task_name -l library -r roles {-i interp_type managed_node_name filename} [-u username] [-g group] [comments]
- wcrttask -F import_file -t task_name -l library [-r roles] [-u user] [-g group]

EXAMPLE:

```
wcrttask -t "Tivoli Object Database Admin" -l "Task Library" -r senior -i
w32-ix86 ntserv1 d:-Tivoli-bin-w32-ix86-bin-odadmin.exe
```

Creates a task called Tivoli Object Database Admin within the task library with a role of senior, with a task executable for the w32-ix86 interpreter (platform Windows NT 3.51) on host ntserv1, and the task executable is a Tivoli command called odadmin.exe. An example of this can be seen in Figure 337 on page 304.

8.5.2.2 Distributing Tasks

NAME: wdisttask

PURPOSE: To control the distribution of task binaries for a task library.

SYNOPSIS:

- wdisttask -q library_name
- wdisttask -s library_name
- wdisttask -d library_name task_name

EXAMPLES:

`wdisttask -q "Task Library"` returns the distribution mode for the task library.

`wdisttask -s "Task Library" LOCAL` sets the distribution mode for this task library to LOCAL.

The -d flag indicates that this distribution takes place immediately.

Note

There are three modes available: ALI, GLOBAL and LOCAL.

- ALI - Distributed only to the local TME Server.
- LOCAL - Distributed to all file servers in the local TMR.
- GLOBAL - Distributed to all file servers in connected TMRs.

8.5.2.3 Listing Tasks

NAME: wgettask

PURPOSE: Lists the properties of a task.

SYNOPSIS:

- wgettask [-F filename] task_name library_name

EXAMPLE:

```
wgettask "Tivoli Object Database Admin" "Task Library"
```

Lists the contents of the Task Tivoli Object Database Admin which is defined within the task library. The following output is received.

```
Task Name             Tivoli Object Database Admin
User Name             *
Group Name
Task ACL              senior

Supported Platforms
          w32-ix86        <install-dir>/w32-ix86/TAS/TASK_LIBRARY
                          /bin/1606608427/Task_Library_ghxygsja
          aix4-r1         <install-dir>/aix4-r1/TAS/TASK_LIBRARY
                          /bin/1606608427/Task_Library_ghxygsja

Task Comments

          Task Name       : Task Library/Tivoli Object Database Admin
          Task Created    : Wed Jul 10 09:18:38 1996
          Task Created By : TMENTDOM\Administrator@ntserv2.itso.ral
                            .ibm.com
          Task Files

          w32-ix86   ntserv2    d:\Tivoli\bin\w32-ix86\bin\odadmin.exe
          aix4-r1    rs600012   /usr/local/Tivoli/bin/aix4-r1/bin/odadmin

          Distribution Mode : ALI

          Task Comments :
          Cross - platform task for receiving Tivoli Management
          Region details using the TME odadmin command.
          ------------------------------------------------------------
```

Figure 337. Task - wgettask Output

8.5.3 Job Commands

There are many commands for creating jobs within TME. These include jobs for creating, deleting, listing and running jobs. Here we show how to list the contents of a job and also how to create a simple job.

8.5.3.1 Creating Jobs

NAME: wcrtjob

PURPOSE: To create a job that can be repeatedly run from the TME to satisfy a recurring task to be performed.

SYNOPSIS:

- wcrtjob -j job_name -l library_name -t task_name -M mode [-s interval -n number] -m timeout -o output_format [-d mannode_name -f filename]]-h mannode_name[[-p prof_mgr_node]

EXAMPLES:

```
wcrtjob -j "All Managed Nodes - DB Admin" -l "Task Library" -t "Tivoli Object
Database Admin" -M parallel -m 20 -o 01 -h ntserv1
```

Creates a job called *All Managed Nodes - DB Admin* in the Task Library associated with the Task Tivoli Object Database Admin. The job will run its tasks in parallel, with a timeout value of 20 seconds on the Managed Node ntserv1.

Another example uses the form:

- wgetjob *job_name library_name*

An example of this would be:

- wgetjob "All Managed Nodes - DB Admin" "Task Library"

This gets the contents of the Job All Managed Nodes - DB Admin defined in the task library.

```
Job Name             : All Managed Nodes - DB Admin
Task Name            : Tivoli Object Database Admin
Execution Mode       : parallel
Timeout              : 60
Output Format        : task header
                       return code
                       standard output
                       standard error output
                       display output to desktop
                       save output to file
                       ntserv2
                       /tmp/odadmin.output

Managed Nodes        : ntserv2 (ManagedNode)
                       rs600012 (ManagedNode)
                       ntserv1 (ManagedNode)

Profile Managers     :
```

Figure 338. Task - wgetjob Output

8.5.4 Scheduler Commands

The scheduler has a rich command line feature. These commands include deleting, editing, creating and listing jobs to be scheduled. The following is an example of displaying the jobs currently scheduled.

8.5.4.1 Listing Scheduler Contents

NAME: wgetsched

PURPOSE: To list all the jobs currently defined within the scheduler.

SYNOPSIS:

- wgetsched

EXAMPLE:

wgetsched lists all the currently scheduled jobs.

```
Job ID Job Label       Admin           Date & Time Enbld Repeat ENTRY
------ --------------- --------------- ----------
000005 All Managed Nod TMENTDOM\Admini Sat Jul 13 06:00:00 1996
```

Chapter 9. Integration

This chapter shows how to integrate other products with the Tivoli TME product set. We worked with:

- TME 10 NetFinity
- NetView for AIX V4.1.4

9.1 TME 10 NetFinity

Before we talk about integration we provide you with a short description about the functions in TME 10 NetFinity.

9.1.1 Basics

TME 10 NetFinity includes the following functions:

- Remote System Manager enables you to access and control all TME 10 NetFinity services installed on remote systems within your network. You can organize the systems in logical groups.
- Remote Session provides access to a remote system using a window that you can enter and receive non-graphical output in.
- File Transfer sends, receives or deletes files and directories remotely or locally.
- Screen View takes a snapshot of a remote system status.
- Power-on error detect warns you if a remote system has startup problems.
- Event scheduler lets you start and stop TME 10 services automatically on remote or local systems.
- Process Manager enables you to see details about all processes that are active on the system. You can define alerts for starting or stopping processes.
- DMI (Desktop Management Interface) browser enables you to examine information about the DMI-compliant hardware and software products installed.
- Software inventory allows you to create and manage software dictionaries.
- System information detects and reports detailed information about the systems installed with a service component.
- Web Manager allows you to access the TME 10 NetFinity Server through a Java-enabled browser.
- Predictive Failure Analysis alerts you if one of the PFA-enabled drives is predicted to fail, letting you identify and replace the drive.
- System Profile allows you to customize user and system information.
- System Monitor displays graphs for several system resources and alerts the user if predefined thresholds are reached.
- Security Manager prevents unauthorized access to your TME 10 NetFinity services.
- Alert Manager receives and processes application-generated alerts.

- ECC (Error conditioning code) memory setup enables you to control ECC memory features on IBM PCs.
- System partition access allows you to update, back up and delete your system partition without a boot diskette.
- Critical file monitor brings up a warning when a file that you have flagged as important is altered or deleted.
- Serial control enables remote manager to access your system through your modem.
- RAID Manager view and configure the RAID subsystem.

Integration Scenario I -

The scenario involves a number of components to integrate. These include the following:

- NT Server v3.51 Operating System
- AIX V4.1 Operating System
- TME 3.0 for NT / AIX
- NetView V4.1 for AIX
- Lotus Notes V4.1 for NT
- Tivoli Sentry.

The scenario involves monitoring the status of a Lotus Notes V4.1 Server running under the NT Server platform. Also on the NT Server we have the TME Platform, TME Desktop and TME Sentry installed. The scenario involves notifying the TME/NT Server as well as the NetView for AIX on the RS6000 that the Notes server has stopped running. We then proceed to restart the Notes server on NT from NetView for AIX using a TME-defined job.

9.1.2 TME Configuration on NT/AIX

For this scenario we have two connected TME servers making two Tivoli Management Regions. The two machines involved are an NT server (ntserv2) and an AIX/RS6000 (rs600012). These TMRs are connected using a two-way secure TMR connection.

```
MODE   NAME             SERVER                     REGION
<----> rs600012-region  rs600012.itso.ral.ibm.com 1899640400
```

Figure 339. Output of wlsconn Command on NT TME Server

```
MODE   NAME             SERVER         REGION
<----> ntserv2-region   ntserv2        1606608427
```

Figure 340. Output of wlsconn Command on AIX TME Server

On the NT server we can configure a Profile Manager resource and populate that resource with a Sentry profile. We call this Sentry profile NT Notes Server.

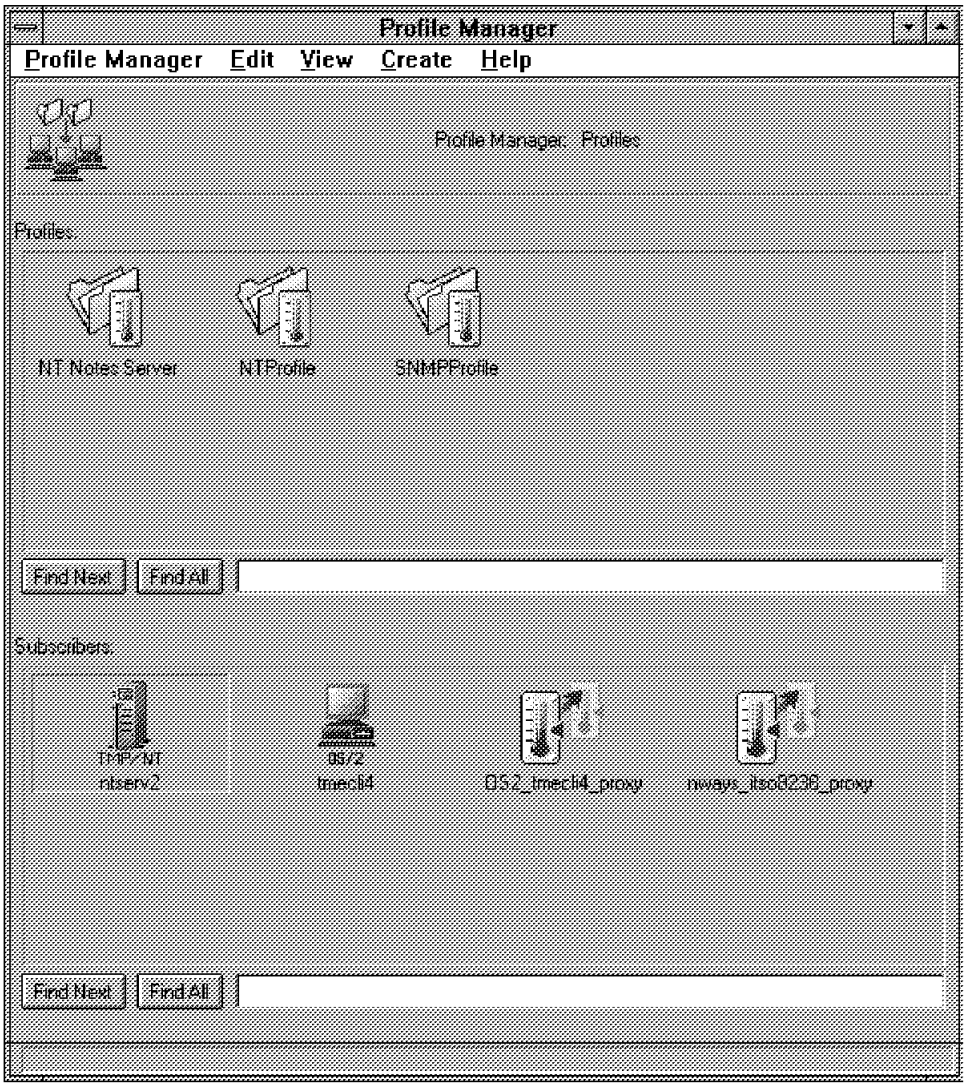


Figure 341. Scenario I - Profile Manager

This profile then has to be configured to monitor the Notes server on our NT server. To configure Sentry to monitor the Notes server, we show the contents of the final monitor both graphically and with the command line using the wlsmon Tivoli command.

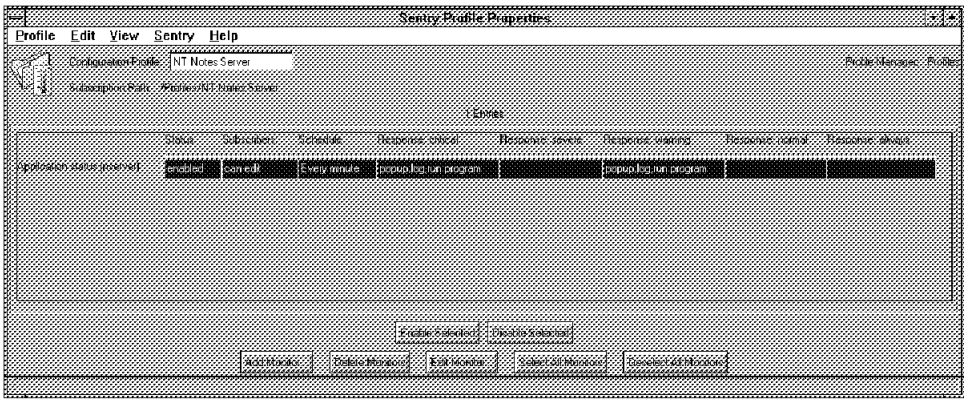


Figure 342. Scenario I - Sentry Profile / Monitor Properties

```
0  Monitor: Application status(nserver)
  Timing:Every minute
  Responses:
    critical
    when probe result  -> down
      popup(Root_rs600012-region,Root_ntserv2-region),
      log(d:\tmp\notes.down),run program(wsnmptrap -h rs600012 -c public
      1.3.6.1.4.1.2.6.9999 6 99 1.3.6.1.4.1.2.6.9999.0
      OctetString NTSERV2/ITSO)
    severe
    warning
    when probe result  -> up
      popup(Root_rs600012-region,Root_ntserv2-region),
      log(d:\tmp\notes.down),run program(wsnmptrap -h rs600012 -c public
      1.3.6.1.4.1.2.6.9999 6 100 1.3.6.1.4.1.2.6.9999.0
      OctetString NTSERV2/ITSO)
    normal
    always
```

Figure 343. Output of wlsmon Command on the TME Server

This output shows the following:

- Monitor used is Application Status.
- Application monitored is nserver.
- The monitor runs every minute.
- A response level of Critical for this monitor indicates that the application is down. This means that the process is no longer running on the NT server.
- Certain actions are executed when this condition is met:
  - Sentry pop-up alerts are sent to TME administrators at both regions rs600012-region and ntserv2-region.
  - The file d:\tmp\notes.down is updated with the Sentry alert.
  - An SNMP trap is sent to rs600012 to enterprise 1.3.6.1.4.1.2.6.9999, generic ID 6, specific ID 99, with the OID 1.3.6.1.4.1.2.6.9999.0 and value of NTSERV2/ITSO.
- A response level of warning means that the probe condition for this monitor meets the up condition. This means that the process is now up and running on the NT server.
- Certain actions are executed when this condition is met:
  - Sentry pop-up alerts are sent to TME administrators at both regions rs600012-region and ntserv2-region.
  - d:\tmp\notes.up gets updated with status information from the Sentry alert.
  - An SNMP trap is sent to rs600012 to enterprise 1.3.6.1.4.1.2.6.9999, generic ID 6, specific ID 100, with the OID 1.3.6.1.4.1.2.6.9999.0 and value of NTSERV2/ITSO.

In addition to the Sentry profile that will take care of all the monitoring during this scenario, we also configured a task/job tfr to restart the Notes server.

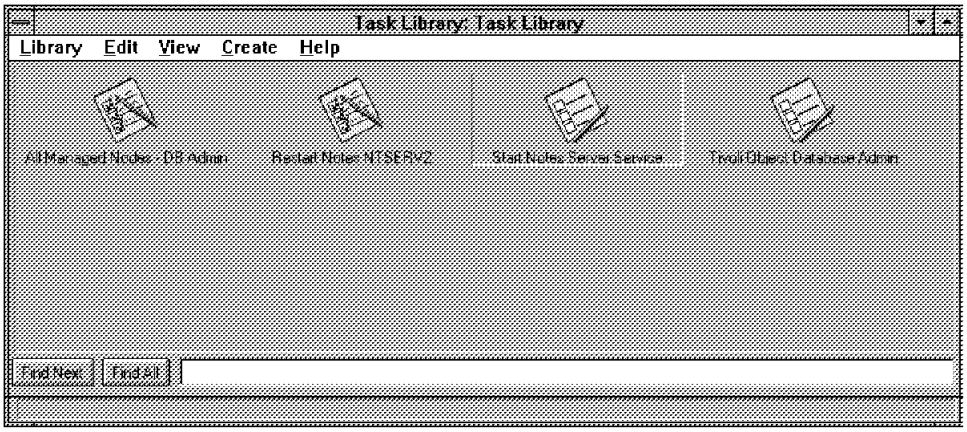


Figure 344. Scenario I - TME Task Library/Task/Job for Notes Server

We can look at the contents of the task and job for restarting the Notes server using the wgettask and wgetjob command.

```
Task Name             Start Notes Server Service
User Name             *
Group Name
Task ACL              user
Supported Platforms
 w32-ix86        <install-dir>/w32-ix86/TAS/TASK_LIBRARY
                 /bin/1606608427/Task_Library_jnadgona
Task Comments
 Task Name            : Task Library/Start Notes Server Service
 Task Created         : Mon Jul 15 17:53:59 1996
 Task Created By      : TMENTDOM\Administrator@ntserv2.itso.ral.ibm.com
 Task Files
     w32-ix86         ntserv2          d:\tmp\restart.cmd
 Distribution Mode    : ALI
 Task Comments        :

 ------------------------------------------------------------
 Task Modified        : Tue Jul 16 15:50:43 1996
 Task Modified By     : TMENTDOM\Administrator@ntserv2.itso.ral.ibm.com
 Task Files
     w32-ix86         ntserv2          c:\winnt35\system32\net.exe
 Distribution Mode    : ALI
 Task Comments        :
```

Figure 345. Scenario I - Output of the wgettask Command

```
Job Name              : Restart Notes NTSERV2
Task Name             : Start Notes Server Service
Execution Mode        : parallel
Timeout               : 60
Output Format         : task header
                        return code
                        standard output
                        standard error output
                        save output to file
                             ntserv2
                             d:\tmp\notessrv.TMEjob

Managed Nodes         : ntserv2 (ManagedNode)

Profile Managers      :
```

Figure 346. Scenario I - Output of the wgetjob Command

This job will run the previously displayed job Restart Notes NTSERV2 on the ManagedNode ntserv2, and log the results to a file on that Managed Node called d:\notes\notessrv.TMEjob.

Note

This job could actually be used for passing any parameters to the net.exe command. It is not restricted to starting the Notes server command. Later we show how to pass parameters to a TME job while running it from the command line.

9.1.3 NetView for AIX Configuration

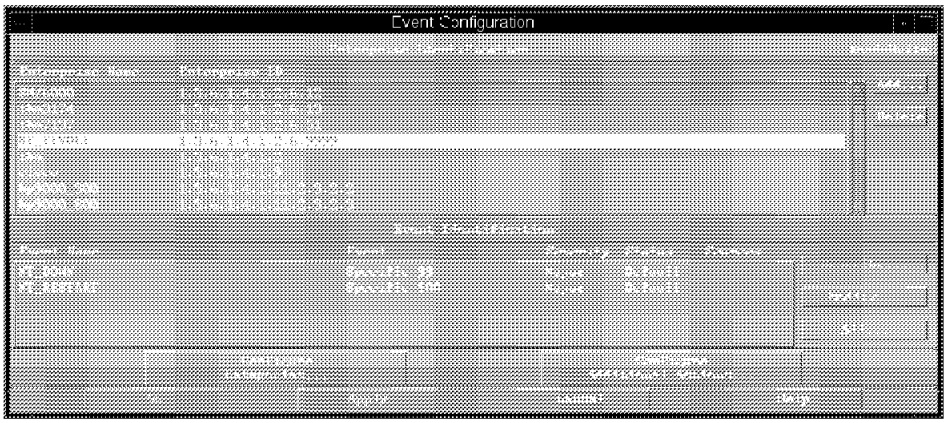


Figure 347. Scenario I - NetView Enterprise/Event Configuration

You can get to the event configuration panel on AIX either by selecting the pull-down **Options, Event Configuration, Trap Customization.. SNMP** or by typing the command: `/usr/OV/bin/xnmtrap &`. In this case we selected the enterprise ID for Tivoli.

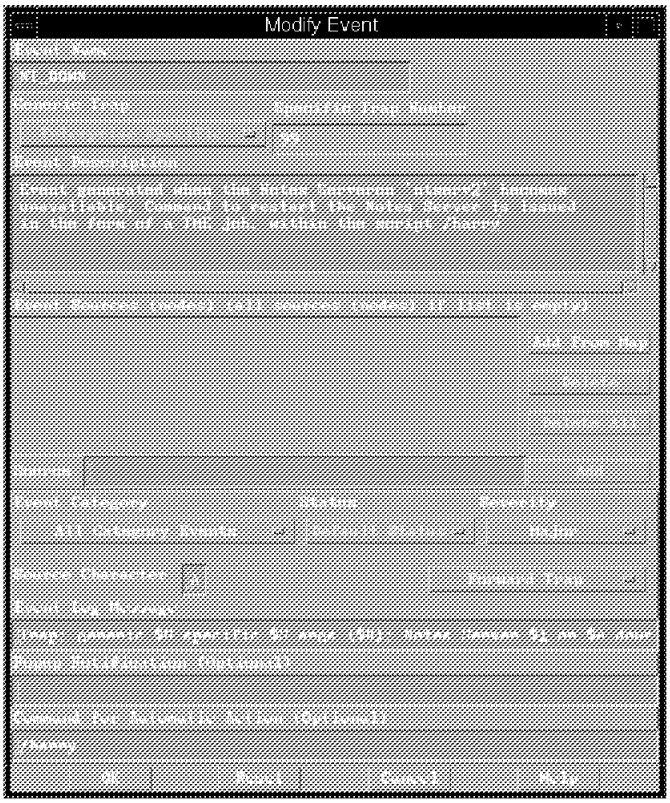


Figure 348. Scenario I - NetView Event - NT_DOWN

Using trap customization, we created an action for that enterprise specific ID, 6 99 which was for NT_DOWN.

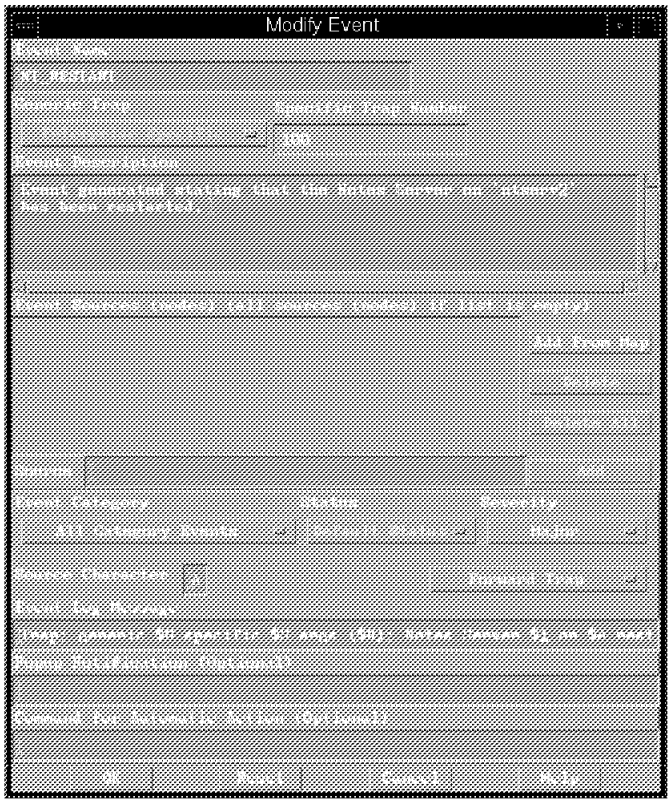


Figure 349. Scenario I - NetView Event - NT_RESTART

Using trap customization, we created an action for that enterprise specific ID, 6 100 which was for NT_RESTART.

```
 ..........
 ..........

######################
#
# Enterprises
#
#

systemsMonitor6000 {1.3.6.1.4.1.2.6.12}
ibm3174 {1.3.6.1.4.1.2.6.13}
ibm7137 {1.3.6.1.4.1.2.6.51}
NT_TIVOLI {1.3.6.1.4.1.2.6.9999}
ibm {1.3.6.1.4.1.2}
cisco {1.3.6.1.4.1.9}
hp9000_300 {1.3.6.1.4.1.11.2.3.2.2}

 ..........
 ..........
 ..........

NT_1 {1.3.6.1.4.1.2.6.9999} 6 99 A 5 0 "All Category Events"
Trap: generic $G specific $S args ($#): Notes Server $1 on $A down.
EXEC /barry
FORWARD
SDESC
EDESC
NT_RESTART {1.3.6.1.4.1.2.6.9999} 6 100 A 5 0 "All Category Events"
Trap: generic $G specific $S args ($#): Notes Server $1 on $A restarted..
FORWARD
SDESC
```

Figure 350. Scenario I - Excerpt from /usr/OV/conf/trapd.conf File on AIX

You can add the enterprise ID for Tivoli using by manually updating the trapd.conf file, or using the graphical user interface. We recommend that you do not manually edit the file. Also, notice in the later part of the trapd.conf file the traps (99 and 100) that we customized.

9.1.4 Output from Scenario

When the Notes server goes down, a Sentry alert is sent to the NT server.

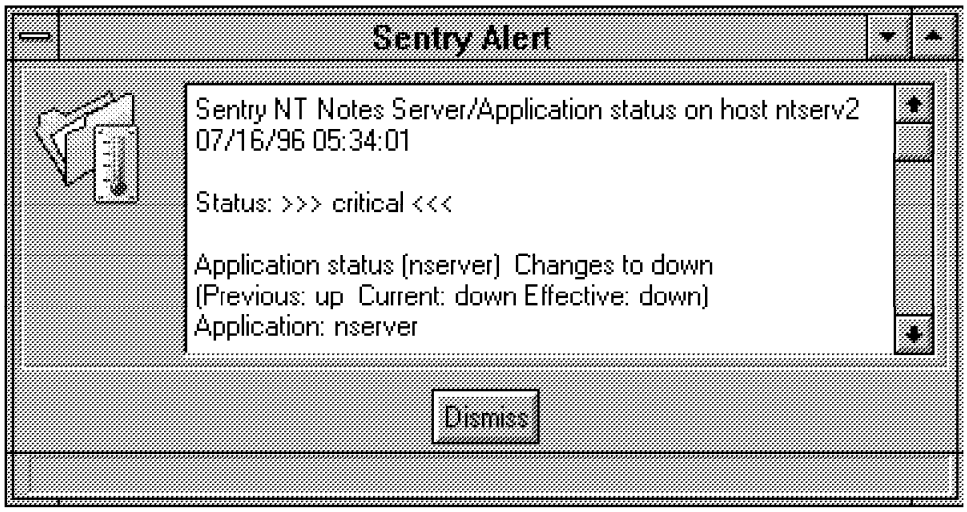


Figure 351. Scenario I - TME Sentry Alert - nserver Unavailable

There is also a copy of this alert sent to ntserv2:d:\tmp\notes.down.

```
Sentry NT Notes Server/Application status on host ntserv2 07/16/96

Status: >>> critical <<<

Application status (nserver)  Changes to down
(Previous: up  Current: down Effective: down)
Application: nserver
```

Figure 352. Scenario I - Output of Alert to ntserv2:d:\tmp\notes.down

In addition, an SNMP trap is sent to NetView for AIX notifying it that the Notes server is now down.

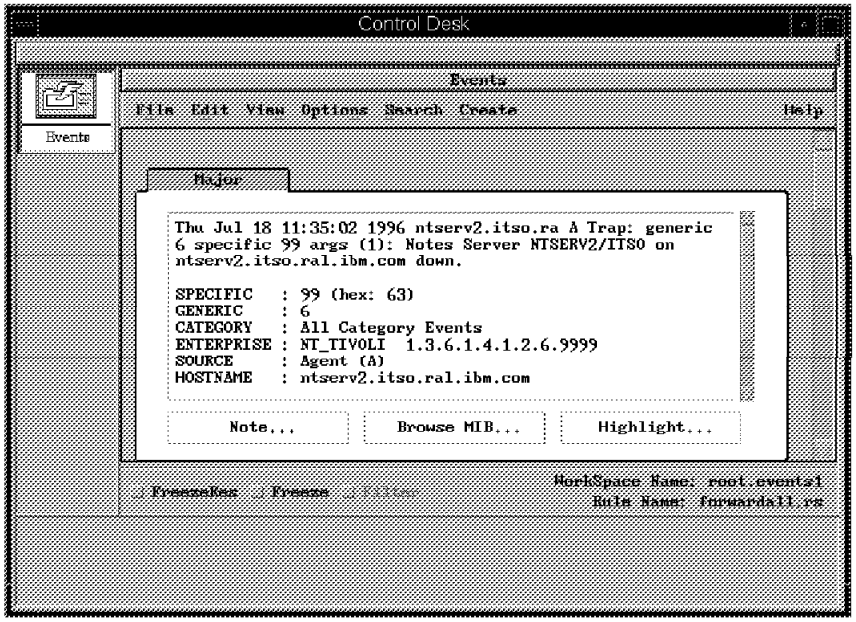


Figure 353. Scenario I - NetView Event - Result of wsnmptrap TME / NT Command

Once this trap is sent to NetView for AIX, the event that was generated had an optional command associated with it to restart the Notes server. This command is actually the command to execute the previously discussed TME job in the TME NT TMR called Restart Notes NTSERV2.

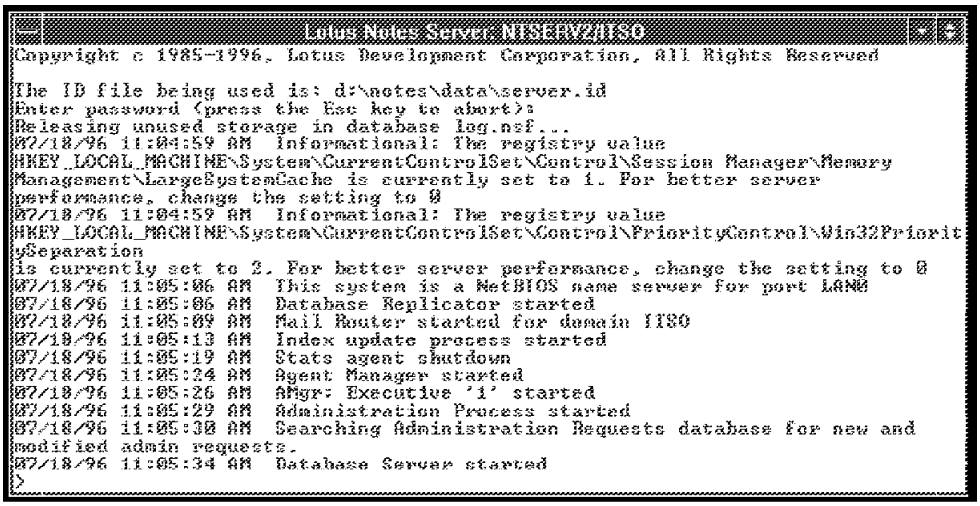


Figure 354. Scenario I - Restarted Notes Server from NetView for AIX

When the Notes server is restarted as a result of the execution of the TME job, we receive notification of the success of the job logged to a file, notessrv.TMEjob.

```
##############################################
Task Name: Start Notes Server Service
Task Endpoint: ntserv2 (ManagedNode)
Return Code: 0
------Standard Output------
The Lotus Notes Server service is starting..
The Lotus Notes Server service was started successfully.


------Standard Error Output------
##############################################
```

Figure 355. Scenario I - Output of Job to ntserv2:d:\tmp\notessrv.TMEjob

If the launch of the Notes server has been successful, we receive a second alert from the Sentry monitor, as well as this being logged under d:\tmp\notes.up.

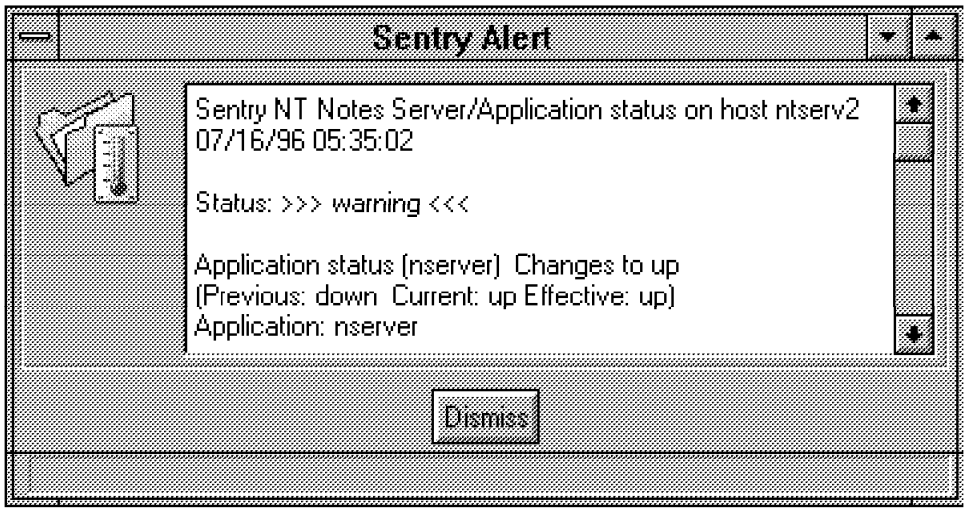


Figure 356. Scenario I - TME Sentry Alert - nserver Available

```
Sentry NT Notes Server/Application status on host ntserv2 07/16/96

Status: >>> warning <<<

Application status (nserver)  Changes to up
(Previous: down  Current: up Effective: up)
Application: nserver
```

Figure 357. Scenario I - Output of Alert to ntserv2:d:\tmp\notes.up

We then sent from the Sentry monitor, when the Notes server became available again, a second SNMP trap to NetView for AIX. This time a different specific trap ID is used, and no command is executed alongside the command, since we were using this as part of our tracking process.

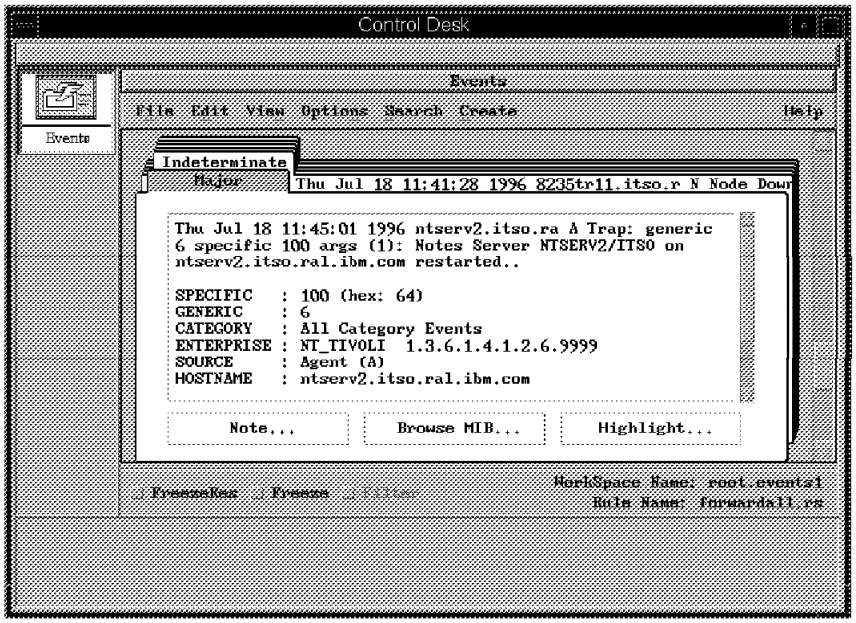


Figure 358. Scenario I - NetView Event - Result of wsnmptrap TME / NT Command

The following diagrams represent the flow of information throughout the scenario.

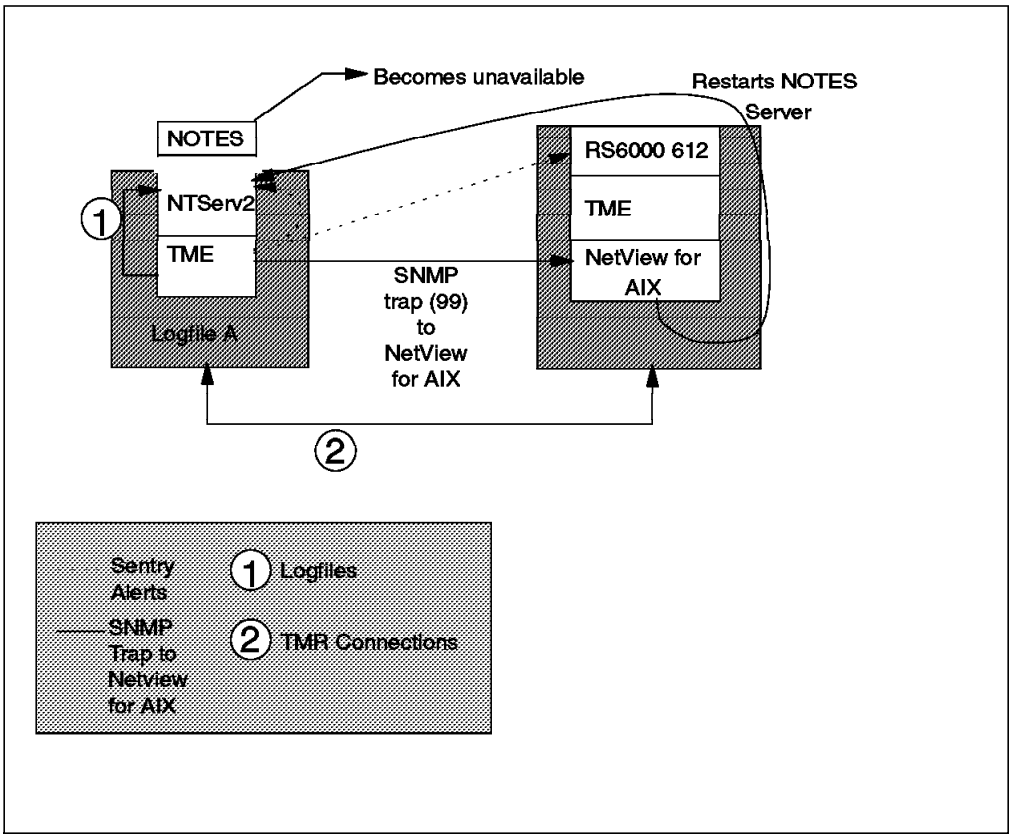


Figure 359. Scenario I - Notes Server Fails, SNMP Trap to NetView for AIX

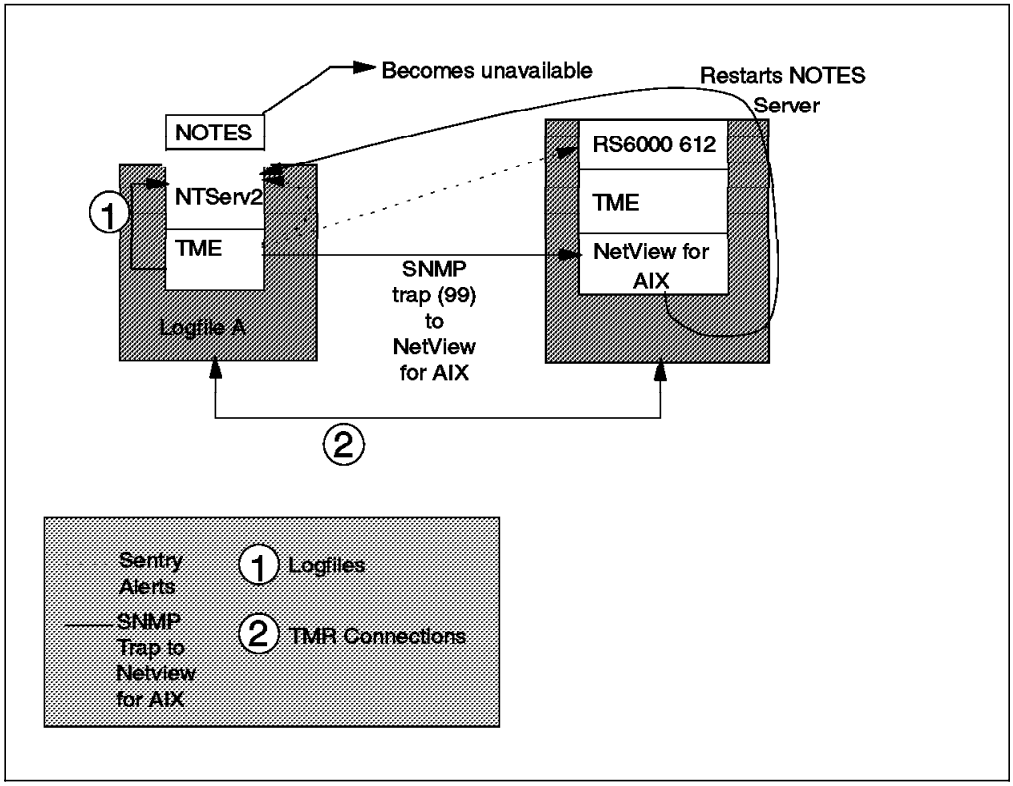


Figure 360. Scenario I - Notes Server Restarted, SNMP Trap to NetView for AIX

Integration Scenario II -

The scenario involves a number of components to integrate. These include the following:

- NT Server V3.51 Operating System
- AIX V4.1 Operating System
- TME 3.0 for NT/AIX
- NetView V4.1 for AIX
- Systems Monitor for AIX V2.3
- Tivoli Sentry Monitor User SNMP

The scenario involves using Tivoli Sentry to monitor an SNMP MIB defined by the Tivoli user. This MIB can be monitored by Sentry for NT using the UserSNMP monitor, as shown in Figure 362 on page 322. We can then issue a command when some pre-defined conditions are met to send a trap to NetView for AIX, or from the NT machine we can issue a command to fix the problem. In our example we monitor two file systems on two AIX machines from our NT TME server.

9.1.5 TME Configuration under NT

The TME configuration to achieve this required us to define the NT Sentry Profile for monitoring file systems. We defined a Sentry Monitor using the UserSNMP Monitor. We created a profile and added a monitor of this type. We were then able to monitor any MIB value on any SNMP-managed device, whether that

value be numeric or a string. We monitored an AIX Systems Monitor MIB value and checked it against different thresholds.

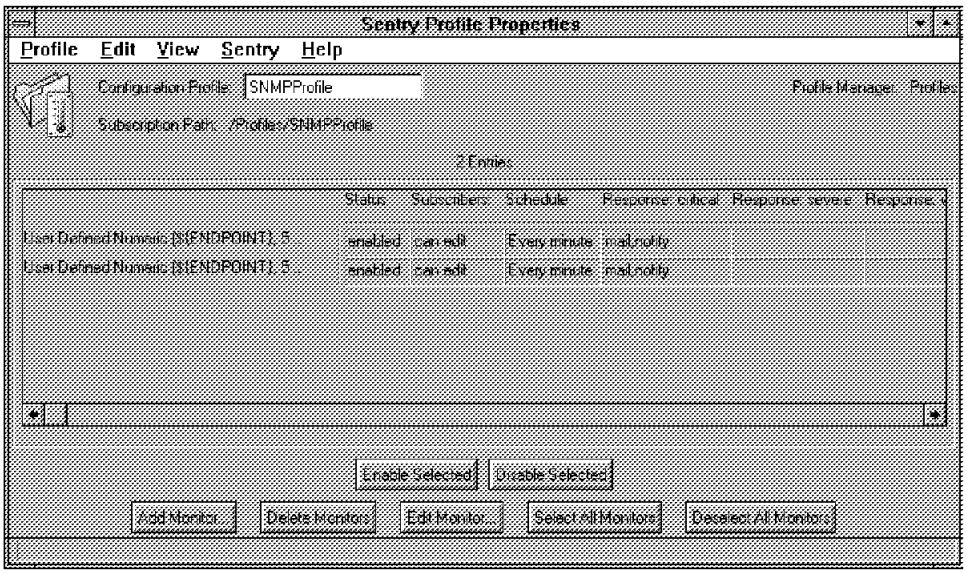


Figure 361. Scenario II - Sentry UserSNMP Monitor

When we add the UserSNMP Monitor, we have a number of arguments that need to be configured. First, there are two types of SNMP monitors:

- User Defined Numeric
- User Defined String

These can be used as probes to satisfy certain conditions that are to be monitored. Within both of these there are four parameters to be configured:

- System Name - Either a ManagedNode or a Sentry proxy endpoint.
- Timeout - A value for timeout purposes.
- SNMP Community name - The SNMP community name for the query.
- SNMP MIB entry - The SNMP MIB variable to query.

After these values are determined and set we can then configure the monitor in a similar fashion as before.

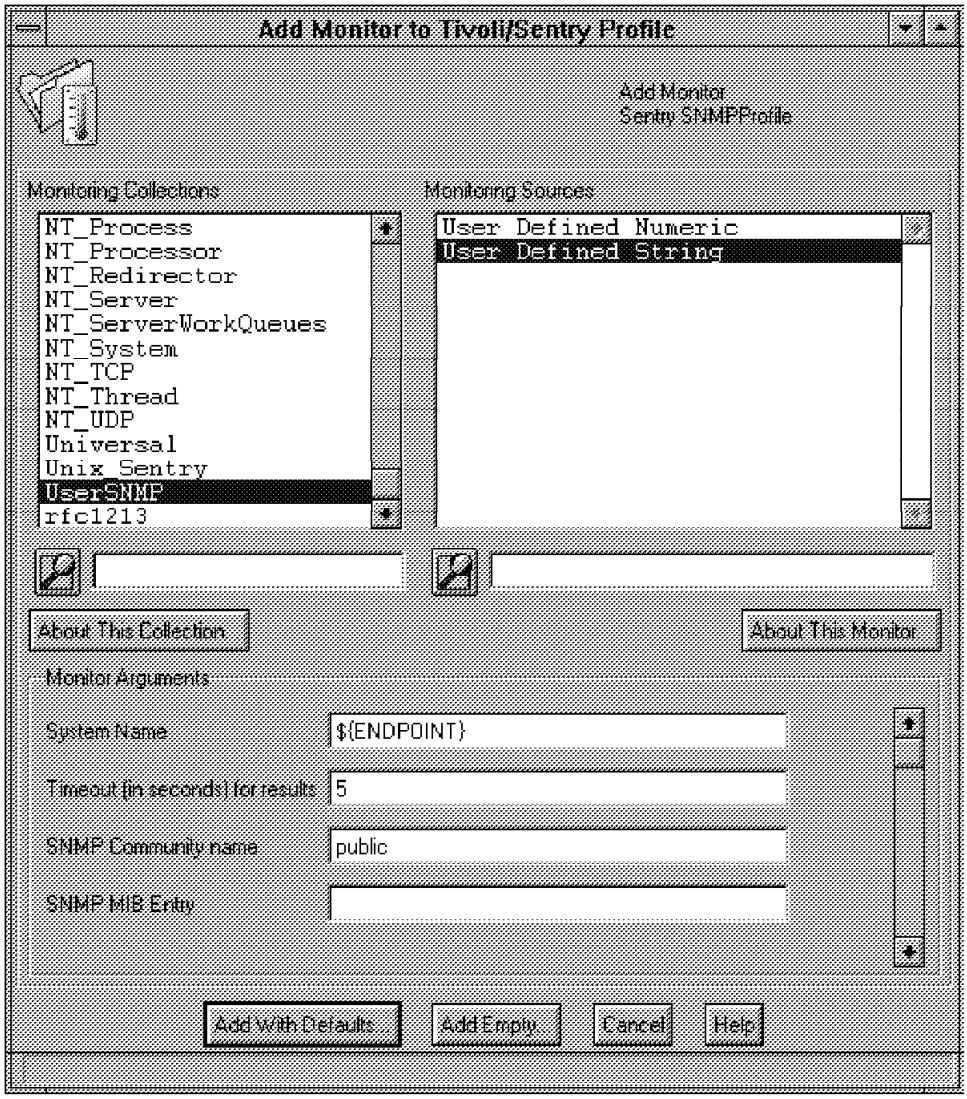


Figure 362. Scenario II - Sentry UserSNMP Monitor Arguments

After filling in the SNMP MIB entry of 1.3.6.1.4.1.2.6.12.2.5.2.1.4.47.117.115.114 click on **Add Empty** and customize.

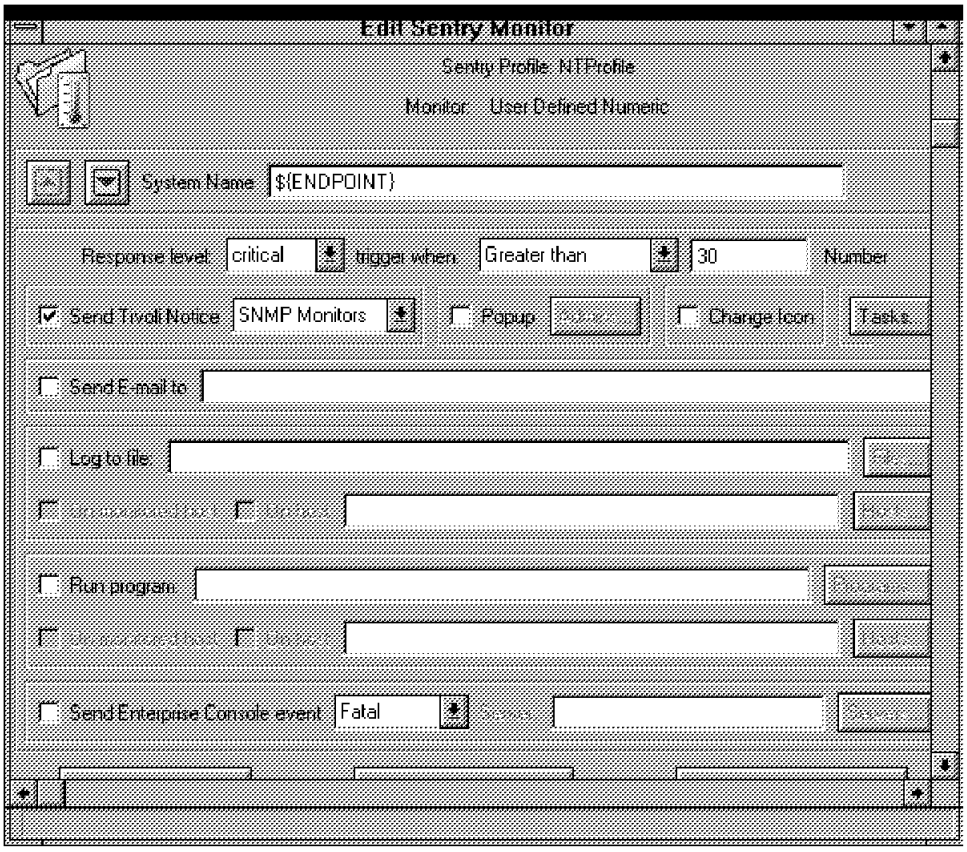


Figure 363. Scenario II - Sentry UserSNMP Edit Monitor

```
0  Monitor: User Defined Numeric(${ENDPOINT}, 5, public,
           1.3.6.1.4.1.2.6.12.2.5.2.1.4.47.117.115.114)
  Timing:Every minute
  Responses:
    critical
    when probe result  > 30
      notify(SNMP Monitors)
    severe
    warning
    normal
    always
1  Monitor: User Defined Numeric(${ENDPOINT}, 5, public,
           1.3.6.1.4.1.2.6.12.2.5.2.1.4.47.118.97.114)
  Timing:Every minute
  Responses:
    critical
    when probe result  > 30
      notify(SNMP Monitors)
    severe
    warning
    normal
    always
```

Figure 364. Scenario II - Output of wlsmon Command on UserSNMP Monitor

The above says that we are going to monitor two user-defined numeric values, such that when the results of these probes is greater than 30 we send a notice to a desktop notice group. In our example, the values that we are monitoring are on two RS/6000 AIX machines. One RS/6000 (rs600012) has Sentry/Tivoli installed, and the other (rs60001) does not, although both have Systems Monitor for AIX installed. The notice group we will notify is SNMP Monitors. To view these monitors we can select the **Notices** icon on the TME desktop, and as long as the user subscribes to the Notice group SNMP Monitors, we can view any notices received.

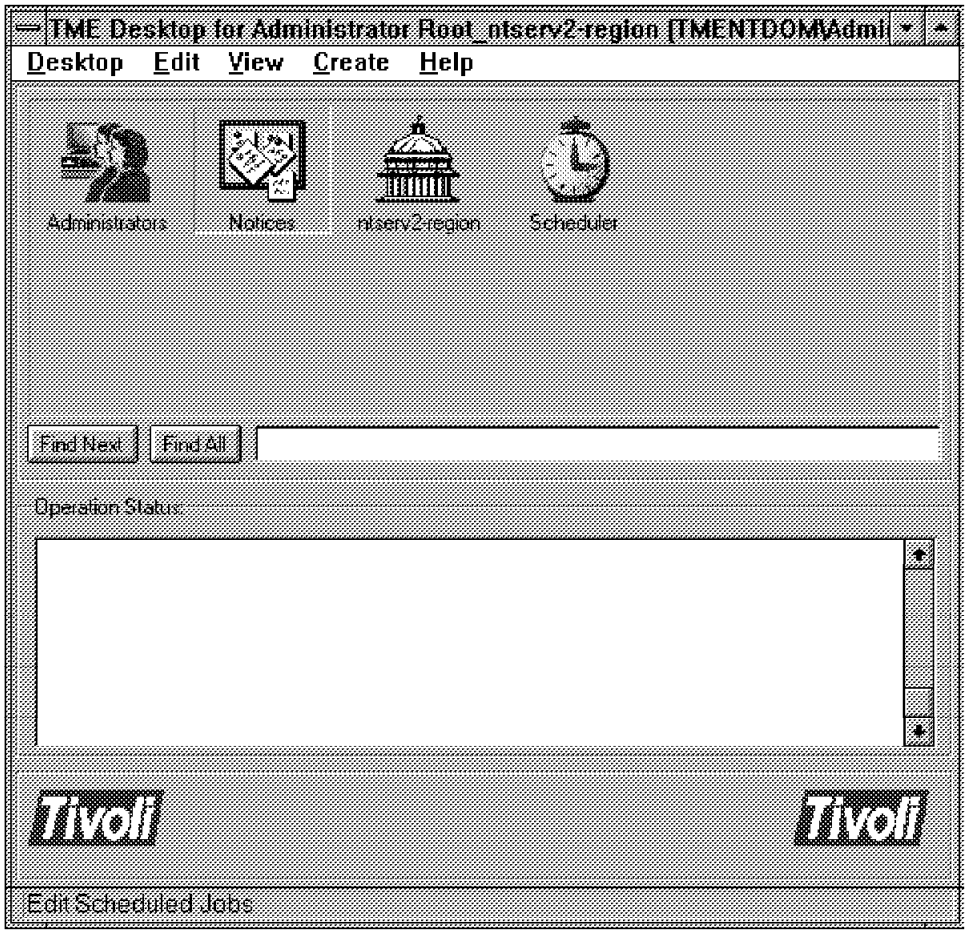


Figure 365. Scenario II - Selecting the TME Notices Service

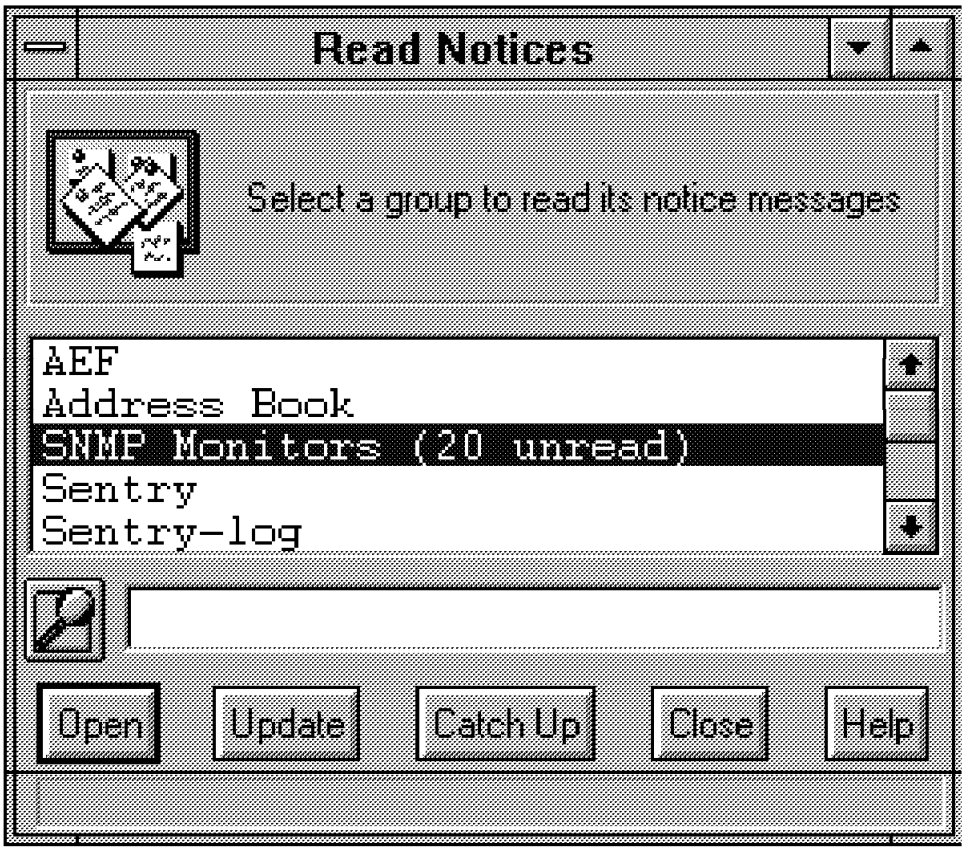


Figure 366. Scenario II - Selecting the SNMP Monitors Group

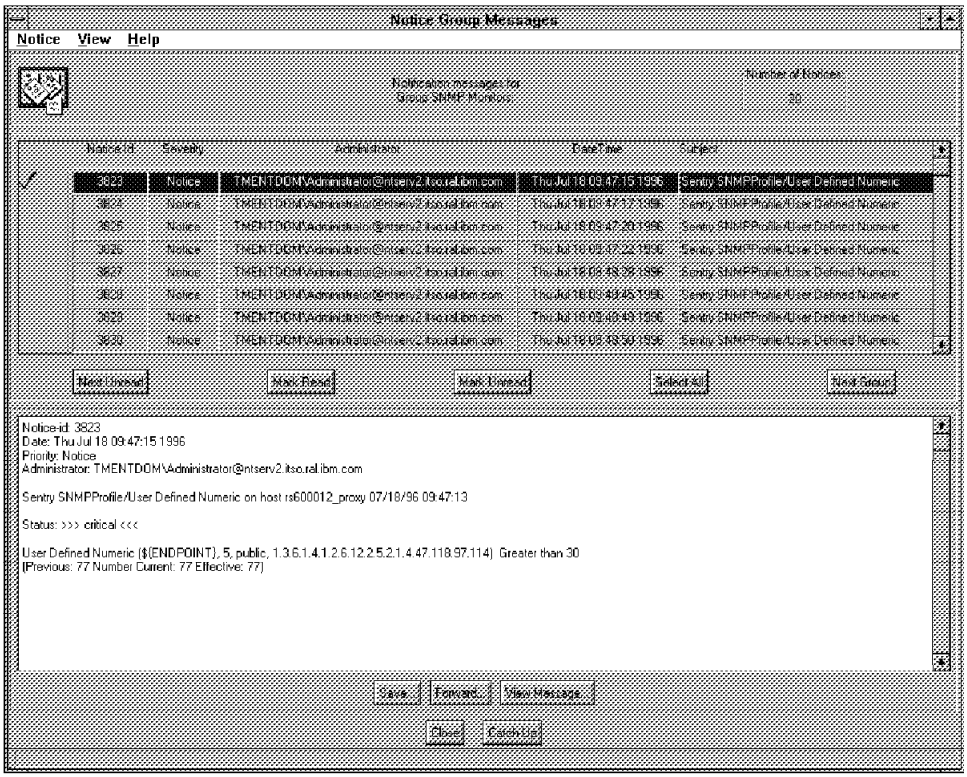


Figure 367. Scenario II - Viewing the SNMP Monitors Notices

To subscribe to the Notice group, open up the **Administrators** icon on the desktop, and with the right mouse button, select **Edit Notice Group Subscriptions** and select which notice groups you wish to subscribe to as per the previous three windows.

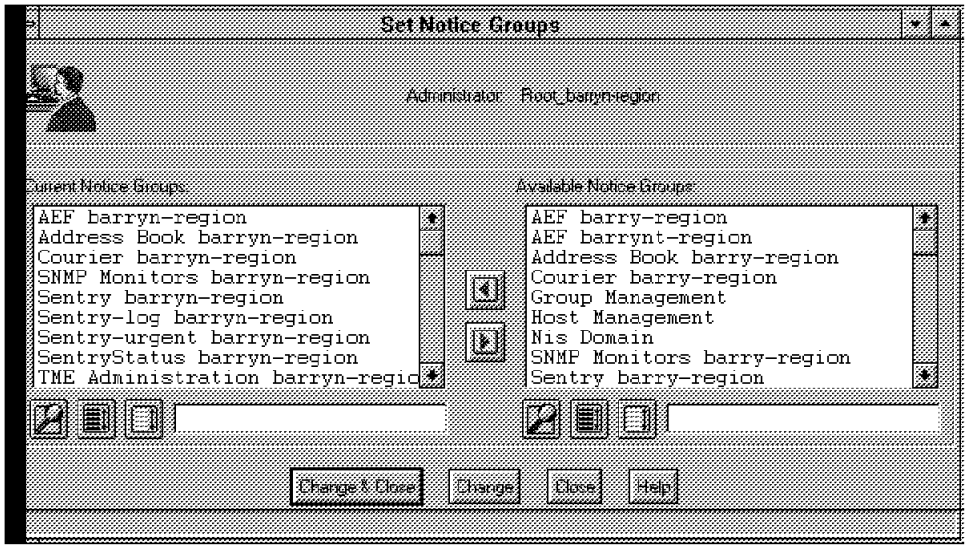


Figure 368. Scenario II - Subscribing to Notice Groups

By distributing this profile to ManagedNode rs600012 and by creating a Sentry proxy node for rs60001 and distributing the profile there, we can obtain any valid MIB value we desire. In our case, we show a simple query on two Systems Monitor for AIX MIB values. These values query the percent utilization of two file systems that we selected to monitor.

- 1.3.6.1.4.1.2.6.12.2.5.2.1.4.47.117.115.114 - /usr filesystem
- 1.3.6.1.4.1.2.6.12.2.5.2.1.4.47.118.97.114 - /var filesystem

While we showed a specific example of monitoring the file systems, /usr and /var, what is important to note is the process that was used. The example showed the process for obtaining any SNMP information from any SNMP managed device. With this information you could take further actions. For example, with the file system example on different flavors of UNIX we could create tasks and jobs to monitor these file systems and when the file systems reach a certain threshold, issue the command for each UNIX to either extend the file system or clean that file system up.

One feature to note here is that there is no requirement for any Tivoli agent to obtain this information. All of this can be done with SNMP.

Appendix A. TME Configuration Files

Table 5 (Page 1 of 2). Resources Available within the Tivoli Management Environment

Resource	Product	Exchanged?
ActiveDesktopList	Platform	No
Admistrator	Platform	Yes
AdmistratorCollection	Platform	Yes
Classes	Platform	No
Distinguished	Paltform	No.
EnterpriseClient	TEC	Yes
EventServer	TEC	Yes
FilePackage	Courier	Yes
GroupNameDB	Admin	No
GroupProfile	Admin	No
HostNamespace	Admin	Yes
IdDatabase	Admin	No
IndicatorCollection	Sentry	Yes
Job	Platform	Yes
ManagedNode	Platform	Yes
MoniteringCapabilityCollection	Sentry	No
NameDatabase	Admin	No
NISDomain	Admin	Yes
PatchInfo	Platform	No
PCManagedNode	Platform	Yes
PolicyRegion	Platform	Yes
Presentation	Platform	No
ProductInfo	Platform	No
ProfileManager	Platform	Yes
ProfileEndpoint	Platform	No
Repeater	Platform	Yes
Scheduler	Platform	No
SentryChannel_n	Sentry	Yes
SentryChannel_s	SEntry	Yes
SentryCustom_n	Sentry	Yes
SentryCustom_s	Sentry	Yes
SentryEngine	Sentry	Yes
SentryPofile	Sentry	Yes
SentryProxy	Sentry	Yes
TMF_Notice	Platform	Yes
TaskLibrary	Platform	Yes
TopLevelPolicyRegion	Platform	Yes

Table 5 (Page 2 of 2). Resources Available within the Tivoli Management Environment		
Resource	**Product**	**Exchanged?**
UserNameDB	Admin	No
UserProfile	Admin	Yes
Workload	Workload	No

Figure 369. Tivoli Management Enterprise Icon - Administrators Group

This icon represents all the administrators available. These include all the administrators defined within the local TMR and in all connected TMRs.

Figure 370. Tivoli Management Enterprise Icon - Administrator

Here the icon represents a specific administrator, containing all its defined roles and resources including login names, notice group subscription lists, TMR roles, resources and properties.

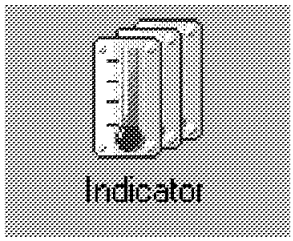


Figure 371. Tivoli Management Enterprise Icon - IndicatorCollection Resource

An IndicatorCollection lets you quickly determine the condition of monitored resources.

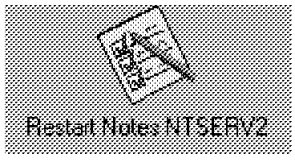


Figure 372. Tivoli Management Enterprise Icon - Job

This icon represents a TME job that has been defined to perform a task which can be independent of platform. A job is always associated to a predefined task.

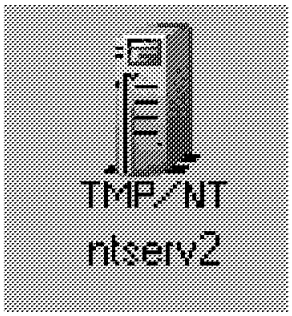


Figure 373. Tivoli Management Enterprise Icon - ManagedNode Resource

Here we have the representation of a ManagedNode resource. This can be on various platforms (above it is on the NT platform). A ManagedNode resource is one with the TME platform installed.

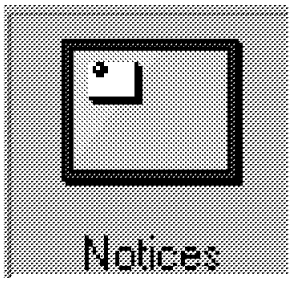


Figure 374. Tivoli Management Enterprise Icon - Empty Notice Board

This icon represents the TME Notice board. This representation of the Notice board occurs only when the Notice board is empty.

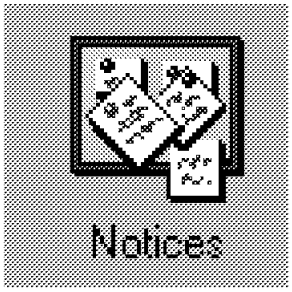


Figure 375. Tivoli Management Enterprise Icon - Populated Notice Board

This representation of the Notice board depicts a Notice board populated with unread messages.

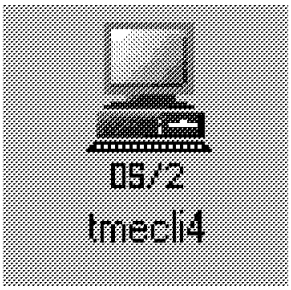


Figure 376. Tivoli Management Enterprise Icon - PcManagedNode Resource

This icon represents the PcManagedNode resource. This managed resource depicts a computer with a TME agent installed. In our example this is an OS/2 PcManaged node. This changes depending on the operating system.

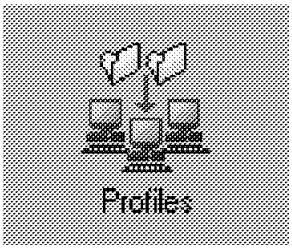


Figure 377. Tivoli Management Enterprise Icon - ProfileManager Resource

The managed resource for ProfileManager is depicted as above. This resource needs to be created to hold all the profiles that are to be used for subscribers.

Figure 378. Tivoli Management Enterprise Icon - Policy Region/Subregion

The dome shaped icon represents both the Top Level Policy Region and the Subregions within Policy Regions. These regions contain all the other managed resources that are allowed for that region.

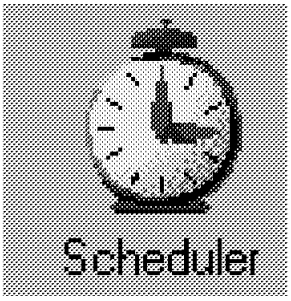


Figure 379. Tivoli Management Enterprise Icon - TME Scheduler

The scheduler is represented by the above icon. The TME scheduler is a service used by TME. It is not a managed resource.

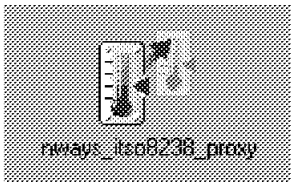


Figure 380. Tivoli Management Enterprise Icon - Sentry Proxy Resource

The Sentry proxy managed resource is depicted by the above. This is used when a device is required to be managed, but does not have a Tivoli agent/platform software installed. It can also be used for other devices.

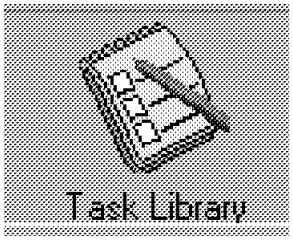


Figure 381. Tivoli Management Enterprise Icon - TaskLibrary Resource

The above icon is used for the Task Library managed resource. This is used to store all the tasks and subsequent jobs that are defined.

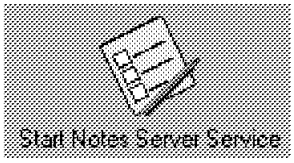


Figure 382. Tivoli Management Enterprise Icon - Task

Tasks are represented by this icon. A task is used to simplify commonly used or run procedures regardless of the operating system used.

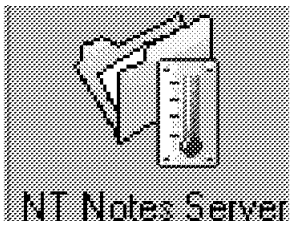


Figure 383. Tivoli Management Enterprise Icon - Sentry Profile

The Sentry Profile is displayed with the above icon. This profile is made up of a number of Sentry monitors and distributed to willing subscribers or managed resources, which could be ManagedNodes, ProfileManagers, SentryProxies.

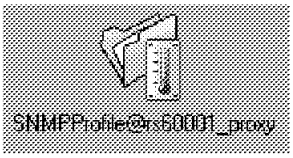


Figure 384. Tivoli Management Enterprise Icon - Sentry Profile @ <ENDPOINT>

This icon represents again a Sentry profile, but this time it states on which endpoint it is operating. In the above example this is on the SentryProxy resource for an RS6000 with no Tivoli software installed.

```
[Notes]
Directory=D:\NOTES\DATA
KitType=2
FileDlgDirectory=D:\NOTES\DATA
OS2IconPath=D:\NOTES\DATA\OS2
$$HasLANPort=1
Preferences=-2146956175
Passthru_LogLevel=0
Console_LogLevel=2
VIEWIMP1=Lotus 1-2-3 Worksheet,0,_IWKSV,,.WKS,.WK1,.WR1,.WRK,.WK3,.WK4,
VIEWIMP3=Structured Text,0,_ISTR,,.LTR,.CGN,.STR,
VIEWIMP4=Tabular Text,0,_ITAB,,.PRN,.RPT,.TXT,.TAB,
VIEWEXP1=Lotus 1-2-3 Worksheet,0,_XWKS,,.WKS,.WK1,.WR1,.WRK,
VIEWEXP3=Structured Text,0,_XSTR,,.LTR,.CGN,.STR,
VIEWEXP4=Tabular Text,1,_XTAB,,.LTR,.RPT,.CGN,.TAB,
EDITIMP1=ASCII Text,0,_ITEXT,,.TXT,.PRN,.C,.H,.RIP,
EDITIMP2=MicrosoftWord RTF,0,_IRTF,,.DOC,.RTF,
EDITIMP3=Lotus 1-2-3 Worksheet,0,_IWKSE,,.WKS,.WK1,.WR1,.WRK,.WK3,.WK4,
EDITIMP4=Lotus PIC,0,_IPIC,,.PIC,
EDITIMP5=CGM Image,0,_IFL,,.GMF,.CGM,
EDITIMP6=TIFF 5.0 Image,0,_ITIFF,,.TIF,
EDITIMP7=BMP Image,0,_IBMP,,.BMP,
EDITIMP8=Ami Pro,0,_IW4W,W4W33F/V0,.SAM,
EDITIMP22=PCX Image,0,_IPCX,,.PCX,
EDITIMP28=Binary with Text,0,_ISTRNGS,,.*,
EDITIMP29=WordPerfect 6.0/6.1,0,_IW4W,W4W48F/V0,.WPD,.WPT,.DOC,
EDITIMP30=Excel 4.0/5.0,0,_IW4W,W4W21F/V4C,.XLS,
EDITIMP31=Word for Windows 6.0,0,_IW4W,W4W49F/V0,.DOC,
EDITIMP32=GIF Image,0,_IGIF,,.GIF,
EDITIMP33=JPEG Image,0,_IJPEG,,.JPG,
EDITEXP1=ASCII Text,2,_XTEXT,,.TXT,.PRN,.C,.H,.RIP,
EDITEXP2=MicrosoftWord RTF,2,_XRTF,,.DOC,.RTF,
EDITEXP3=CGM Image,2,_XCGM,,.CGM,.GMF,
EDITEXP4=TIFF 5.0 Image,2,_XTIFF,,.TIF,
EDITEXP5=Ami Pro,2,_XW4W,W4W33T/V0,.SAM,
EDITEXP21=WordPerfect 6.0,2,_XW4W,W4W48T/V0,.DOC,
EDITEXP22=WordPerfect 6.1,2,_XW4W,W4W48T/V1,.WPD,.WPT,.DOC,
EDITEXP23=Word for Windows 6.0,2,_XW4W,W4W49T/V0,.DOC,
DDETimeout=10
NAMEDSTYLE0=020042617369630000000000000000000000000000000000000000000000000000000000000001010100000A0000000000000100
NAMEDSTYLE1=020042756C6C65740000000000000000000000000000000000000000000000000000000000001010100000A0000000000000000
NAMEDSTYLE2=0200486561646C696E65000000000000000000000000000000000000000000000000000000000010101010B0C0000000000000100
$$$OpenSpecial=NotesNIC, InterNotes
$$$NotesNIC=CN=Home/OU=Notes/O=NET, welcome.nsf, Notes NIC Welcome, Notes Network Information Center on the Internet
$$$InterNotes=,web.nsf,Web Navigator, Local InterNotes Web Navigator Database
ServerTasks=Replica,Router,Update,Stats,AMgr,Adminp
ServerTasksAt1=Catalog,Design
ServerTasksAt2=UpdAll,Object Collect mailobj.nsf
ServerTasksAt5=Statlog
TCPIP=TCP,0,15,0,,12288,
Ports=LAN0,TCPIP,COM1
LAN0=NETBIOS,0,15,0,,12288,
SPX=NWSPX, 0, 15, 0
VINES=VINES, 0, 15, 0
IBM_LAN0=IBMEENB, 0, 50, 0
AppleTalk=ATALK, 0, 15, 0
COM1=XPC,1,15,0,,12292,19200,,auto.mdm,60,15
COM2=XPC,2,15,0,
COM3=XPC,3,15,0,
COM4=XPC,4,15,0,
COM5=XPC,5,15,0,
DisabledPorts=SPX,VINES,IBM_LAN0,AppleTalk,COM2,COM3,COM4,COM5
LOG_REPLICATION=0
LOG_SESSIONS=0
KeyFilename=user.id
CertificateExpChecked=D:\NOTES\DATA\user.id 05-30-96
ZONE_SET=1
Timezone=5
```

Figure 385. Notes Server for OS/2 Conf. File - Notes DC (Part 1 of 2)

```
DST=1
CertifierIDFile=D:\NOTES\DATA\cert.id
Domain=ITSO
MailServer=CN=NOTESOS2/O=ITSO
MailSystem=0
ServerKeyFileName=server.id
Admin=CN=Jason Forsyth/O=ITSO
TemplateSetup=1
Setup=47
ServerSetup=6
PhoneLog=2
Log=log.nsf, 1, 0, 7, 40000
ADMINWINDOWSIZE=32 54 326 453
EmptyTrash=0
SDI_WINDOW=0
StackedIcons=1
DESKWINDOWSIZE=4 0 788 456
WINDOWSIZEPM=92 55 796 512
MAXIMIZED=1
OS2IconCommonConfig=Universal
OS2IconSize=2
OS2IconPos=2
OS2IconHidden=0
OS2IconRect=0 696 1024 720
OS2InfoboxPos=2 110
MDICHILDRENRESTORED=0
TCPIP_TcpConnectTimeout=0,5
BCASEWINDOWSIZE=4 0 788 456
```

Figure 386. Notes Server for OS/2 Conf. File - Notes DC (Part 2 of 2)

```
[Notes]
KitType=2
Directory=d:\notes\data
CONSOLE_Lotus_Notes_Server=80 25 7 28 21 676 348
FileDlgDirectory=D:\NOTES
WinNTIconPath=d:\notes\data\W32
$$HasLANPort=1
Preferences=2148011121
Passthru_LogLevel=0
Console_LogLevel=2
VIEWIMP1=Lotus 1-2-3 Worksheet,0,_IWKSV,,.WKS,.WK1,.WR1,.WRK,.WK3,.WK4,
VIEWIMP3=Structured Text,0,_ISTR,,.LTR,.CGN,.STR,
VIEWIMP4=Tabular Text,0,_ITAB,,.PRN,.RPT,.TXT,.TAB,
VIEWEXP1=Lotus 1-2-3 Worksheet,0,_XWKS,,.WKS,.WK1,.WR1,.WRK,
VIEWEXP3=Structured Text,0,_XSTR,,.LTR,.CGN,.STR,
VIEWEXP4=Tabular Text,1,_XTAB,,.LTR,.RPT,.CGN,.TAB,
EDITIMP1=ASCII Text,0,_ITEXT,,.TXT,.PRN,.C,.H,.RIP,
EDITIMP2=MicrosoftWord RTF,0,_IRTF,,.DOC,.RTF,
EDITIMP3=Lotus 1-2-3 Worksheet,0,_IWKSE,,.WKS,.WK1,.WR1,.WRK,.WK3,.WK4,
EDITIMP4=Lotus PIC,0,_IPIC,,.PIC,
EDITIMP5=CGM Image,0,_IFL,,.GMF,.CGM,
EDITIMP6=TIFF 5.0 Image,0,_ITIFF,,.TIF,
EDITIMP7=BMP Image,0,_IBMP,,.BMP,
EDITIMP8=Ami Pro,0,_IW4W,W4W33F/V0,.SAM,
EDITIMP22=PCX Image,0,_IPCX,,.PCX,
EDITIMP28=Binary with Text,0,_ISTRNGS,,.*,
EDITIMP29=WordPerfect 6.0/6.1,0,_IW4W,W4W48F/V0,.WPD,.WPT,.DOC,
EDITIMP30=Excel 4.0/5.0,0,_IW4W,W4W21F/V4C,.XLS,
EDITIMP31=Word for Windows 6.0,0,_IW4W,W4W49F/V0,.DOC,
EDITIMP32=GIF Image,0,_IGIF,,.GIF,
EDITIMP33=JPEG Image,0,_IJPEG,,.JPG,
EDITEXP1=ASCII Text,2,_XTEXT,,.TXT,.PRN,.C,.H,.RIP,
EDITEXP2=MicrosoftWord RTF,2,_XRTF,,.DOC,.RTF,
EDITEXP3=CGM Image,2,_XCGM,,.CGM,.GMF,
EDITEXP4=TIFF 5.0 Image,2,_XTIFF,,.TIF,
EDITEXP5=Ami Pro,2,_XW4W,W4W33T/V0,.SAM,
EDITEXP21=WordPerfect 6.0,2,_XW4W,W4W48T/V0,.DOC,
EDITEXP22=WordPerfect 6.1,2,_XW4W,W4W48T/V1,.WPD,.WPT,.DOC,
EDITEXP23=Word for Windows 6.0,2,_XW4W,W4W49T/V0,.DOC,
DDETimeout=10
NAMEDSTYLE0=020042617369630000000000000000000000000000000000000000000000000000000000000000000000001010100000A0000000000000100
NAMEDSTYLE1=020042756C6C6574000000000000000000000000000000000000000000000000000000000000000000000000001010100000A0000000000000000
NAMEDSTYLE2=02004865616 46C696E650000000000000000000000000000000000000000000000000000000000000000001010101 0B0C0000000000000100
$$$OpenSpecial=NotesNIC, InterNotes
$$$NotesNIC=CN=Home/OU=Notes/O=NET, welcome.nsf, Notes NIC Welcome, Notes Network Information Center on the Internet
$$$InterNotes=,web.nsf,Web Navigator, Local InterNotes Web Navigator Database
ServerTasks=Replica,Router,Update,Stats,AMgr,Adminp
ServerTasksAt1=Catalog,Design
ServerTasksAt2=UpdAll,Object Collect mailobj.nsf
ServerTasksAt5=Statlog
LAN0=NETBIOS, 0, 15, 0
Ports=LAN0
SPX=NWSPX, 0, 15, 0
VINES=VINES, 0, 15, 0
TCPIP=TCP, 0, 15, 0
AppleTalk=ATALK, 0, 15, 0
COM1=XPC,1,15,0,
COM2=XPC,2,15,0,
COM3=XPC,3,15,0,
COM4=XPC,4,15,0,
COM5=XPC,5,15,0,
DisabledPorts=SPX,VINES,TCPIP,AppleTalk,COM1,COM2,COM3,COM4,COM5
LOG_REPLICATION=1
LOG_SESSIONS=1
ExistingServerName=CN=notesos2/O=itso
KeyFilename=d:\notes\data\server.id
CertificateExpChecked=d:\notes\data\server.id 07/18/96
Domain=ITSO
MailServer=CN=NTSERV2/O=ITSO
Admin=CN=Jason Forsyth/O=ITSO
TemplateSetup=1
Setup=47
ServerSetup=6
ZONE_SET=1
Timezone=5
DST=1
PhoneLog=2
Log=log.nsf, 1, 0, 7, 40000
```

Figure 387. Notes Server for NT Conf. File - Notes Additional Server

```
[TCPIPAGENT]
UpdateIPInterval=1440
UpdateIPAtBootup=YES
DefaultServer=ntserv2
NTClientObjectID=ntserv1
ClientObjectID=ntserv1
ClientName=ntserv1
CurrentBaseDir=D:\TIVOLI\TMEAGENT\win32
OperatingMsgCatDir=D:\TIVOLI\TMEAGENT\win32\msgcat
LinkStatus=BAD

[AGENT]
MachineID=T+D8379GX11C92CKTMRYOOOOO527
```

Figure 388. TME Agent Configuration for NT Server

```
[AGENT]

[TCPIPAGENT]

[IPXAGENT]

[IPX]
CLIENTNAME=
QUERYMODE=AUTO
QUERYSTATUS=
PREFERREDTNWR=
LOGINMODE=
```

Figure 389. TME Agent Configuration for OS/2 Client

```
D:\Tivoli\bin
D:\Tivoli\db
D:\Tivoli\desktop
D:\Tivoli\help
D:\Tivoli\include
D:\Tivoli\install_bundle
D:\Tivoli\lib
D:\Tivoli\msg_cat
D:\Tivoli\src
D:\Tivoli\bin\aix4-r1
D:\Tivoli\bin\client_bundle
D:\Tivoli\bin\generic
D:\Tivoli\bin\generic_unix
D:\Tivoli\bin\w32-ix86
D:\Tivoli\bin\aix4-r1\TAS
D:\Tivoli\bin\aix4-r1\TAS\TASK_LIBRARY
D:\Tivoli\bin\aix4-r1\TAS\TASK_LIBRARY\bin
D:\Tivoli\bin\aix4-r1\TAS\TASK_LIBRARY\bin\1606608427
D:\Tivoli\bin\client_bundle\.installed
D:\Tivoli\bin\client_bundle\bin
D:\Tivoli\bin\client_bundle\bin\aix3-r2
D:\Tivoli\bin\client_bundle\bin\aix4-r1
D:\Tivoli\bin\client_bundle\bin\hpux10
D:\Tivoli\bin\client_bundle\bin\hpux9
D:\Tivoli\bin\client_bundle\bin\solaris2
D:\Tivoli\bin\client_bundle\bin\sunos4
D:\Tivoli\bin\client_bundle\bin\w32-ix86
D:\Tivoli\bin\generic\.installed
D:\Tivoli\bin\generic\SentryMonitors
D:\Tivoli\bin\generic\SNMP
D:\Tivoli\bin\generic_unix\.installed
D:\Tivoli\bin\w32-ix86\.installed
D:\Tivoli\bin\w32-ix86\ADE
D:\Tivoli\bin\w32-ix86\AEF
D:\Tivoli\bin\w32-ix86\bin
D:\Tivoli\bin\w32-ix86\contrib
D:\Tivoli\bin\w32-ix86\mslib
D:\Tivoli\bin\w32-ix86\TAS
D:\Tivoli\bin\w32-ix86\TME
D:\Tivoli\bin\w32-ix86\TMF
D:\Tivoli\bin\w32-ix86\tools
D:\Tivoli\bin\w32-ix86\UI
D:\Tivoli\bin\w32-ix86\contrib\.installed
D:\Tivoli\bin\w32-ix86\TAS\Administrator
D:\Tivoli\bin\w32-ix86\TAS\BACKUP
D:\Tivoli\bin\w32-ix86\TAS\BBOARD
D:\Tivoli\bin\w32-ix86\TAS\CCMS
D:\Tivoli\bin\w32-ix86\TAS\ENDPOINTUI
D:\Tivoli\bin\w32-ix86\TAS\FILEIO
D:\Tivoli\bin\w32-ix86\TAS\INSTALL
D:\Tivoli\bin\w32-ix86\TAS\InterRegion
D:\Tivoli\bin\w32-ix86\TAS\MANAGED_NODE
D:\Tivoli\bin\w32-ix86\TAS\NAVIG
D:\Tivoli\bin\w32-ix86\TAS\NETWARE
D:\Tivoli\bin\w32-ix86\TAS\PC_MANNODE
D:\Tivoli\bin\w32-ix86\TAS\PRDO
```

Figure 390. Directory Structure for TME Platform/Desktop/Sentry on TMR Server (Part 1 of 2)

```
D:\Tivoli\bin\w32-ix86\TAS\SCHEDULER
D:\Tivoli\bin\w32-ix86\TAS\SharedPolicyRegions
D:\Tivoli\bin\w32-ix86\TAS\TASK_LIBRARY
D:\Tivoli\bin\w32-ix86\TAS\TGC
D:\Tivoli\bin\w32-ix86\TAS\TASK_LIBRARY\bin
D:\Tivoli\bin\w32-ix86\TAS\TASK_LIBRARY\bin\1606608427
D:\Tivoli\bin\w32-ix86\TME\PHONE
D:\Tivoli\bin\w32-ix86\TME\SENTRY
D:\Tivoli\bin\w32-ix86\TMF\BASESVCS
D:\Tivoli\bin\w32-ix86\TMF\NOTIF
D:\Tivoli\bin\w32-ix86\tools\lib
D:\Tivoli\bin\w32-ix86\tools\lib\perl
D:\Tivoli\db\ntserv2.db
D:\Tivoli\db\ntserv2.db\.installed
D:\Tivoli\db\ntserv2.db\file_versions
D:\Tivoli\db\ntserv2.db\tmp
D:\Tivoli\db\ntserv2.db\tmp\task2
D:\Tivoli\db\ntserv2.db\tmp\task3
D:\Tivoli\db\ntserv2.db\tmp\task4
D:\Tivoli\db\ntserv2.db\tmp\task5
D:\Tivoli\include\w32-ix86
D:\Tivoli\include\w32-ix86\.installed
D:\Tivoli\include\w32-ix86\cpl
D:\Tivoli\include\w32-ix86\kerberos
D:\Tivoli\include\w32-ix86\tivoli
D:\Tivoli\include\w32-ix86\trans
D:\Tivoli\include\w32-ix86\cpl\arpa
D:\Tivoli\include\w32-ix86\cpl\netinet
D:\Tivoli\include\w32-ix86\cpl\rpc
D:\Tivoli\include\w32-ix86\cpl\sys
D:\Tivoli\install_bundle\misc
D:\Tivoli\lib\w32-ix86
D:\Tivoli\lib\w32-ix86\.installed
D:\Tivoli\msg_cat\.installed
D:\Tivoli\msg_cat\C
D:\Tivoli\src\.installed
D:\Tivoli\src\sample_app
D:\Tivoli\src\sample_app\cli
D:\Tivoli\src\sample_app\config
D:\Tivoli\src\sample_app\plbo
D:\Tivoli\src\sample_app\ppo
D:\Tivoli\src\sample_app\ppo\Bitmaps
D:\Tivoli\src\sample_app\ppo\Dialogs
```

Figure 391. Directory Structure for TME Platform/Desktop/Sentry on TMR Server (Part 1 of 2)

```
Administrators
CurrentNtRepeat
Installation
InterRegion
InterfaceRepository
Library
MessageCatalog
NameRegistry
NotificationServer
Regions
RepeaterManager
Scheduler
ServerManagedNode
TME_server
TMRBackup
TaskRepository
lost-n-found
```

Figure 392. TME Database Root/ - Obtained by Issuing the wls Command

```
1606608427.1.168#TMF_Administrator::Collection_GUI#      Administrators
1606608427.1.380#TMF_Install::NtRepeat#  CurrentNtRepeat
1606608427.1.367#TMF_Install::Engine#    Installation
1606608427.1.353#TMF_InterRegion::Connection# InterRegion
1606608427.1.4      InterfaceRepository
1606608427.1.14#TMF_SysAdmin::Library#   Library
1606608427.1.75#TMF_Message::Catalog#    MessageCatalog
1606608427.1.26     NameRegistry
1606608427.1.88#TMF_SysAdmin::InstanceManager#      NotificationServer
1606608427.1.194#SharedPolicyRegions::Engine# Regions
1606608427.1.340    RepeaterManager
1606608427.1.157#TMF_Scheduler::scheduler#    Scheduler
1606608427.1.322#TMF_ManagedNode::Managed_Node#     ServerManagedNode
1606608427.0.0      TME_server
1606608427.1.346#TMF_SysAdmin::InstanceManager#     TMRBackup
1606608427.1.214#TMF_Task::TaskRepository#    TaskRepository
1606608427.1.411#TMF_TGC::CollectionGUI#      lost-n-found
```

Figure 393. TME Database Root/ - Obtained by Issuing the wls Command

```
1606608427.1.195#TMF_PolicyRegion::GUI#  ntserv2-region
1606608427.1.481#TMF_PolicyRegion::GUI#  Tivoli/Sentry Defaults-ntserv2-region
1606608427.1.656#TMF_PolicyRegion::GUI#  TivoliDefaultPhoneRegion
1899640400.1.196#TMF_PolicyRegion::GUI#  rs600012-region
1899640400.1.506#TMF_PolicyRegion::GUI#  Tivoli/Sentry Defaults-rs600012-region
1908279517.1.195#TMF_PolicyRegion::GUI#  ntserv1-region
1908279517.1.483#TMF_PolicyRegion::GUI#  Tivoli/Sentry Defaults-ntserv1-region
```

Figure 394. TME Database /Regions - Using wcd and wls Commands

```
#!/bin/ksh -f

#if [[ "$TERM" = *%* ]] ; then
#  export  DISPLAY=${TERM#${TERM%\%*}\%}
#  export  TERM=${TERM%\%*}
#fi

# read Hints - foreach do a wcd
# if OK { add to final as 'd' (with ls) & add newnodes to newhints }
#  else { add to final as 'n' (with ls) }

rm nodes hints newhints final

echo / > hints

while [ -f hints ]
do
  [ -f newhints ] && rm newhints

for p in `cat hints`
do
  echo HINT: $p
  if wcd $p
  then
    echo $p d `wls -lod $p` >> final
    echo BASE $p >> newhints
    wls $p >> newhints
  else
    echo $p n `wls -lod $p` >> final
  fi
done

if [ `wc -l newhints|awk '{print $1}'` -gt 0 ]
then
  awk -f odump.awk newhints > hints
  rm newhints
else
  echo Done
fi

done

sort final > nodes

rm final hints newhints

exit 1

wrls()
{
    $1
    if wcd $1
    then
      echo $1 d `wls -lod $1`
      for i in `wls $1`
      do
        wrls $1/$i
      done
    else
      echo $1 n `wls -lod $1`
    fi
}

wcd /

wrls
```

Figure 395. Shell Script for AIX / NT for Dumping the Object Database - odump

```
BEGIN {  }
$1 == "BASE" { BASE=$2; if (BASE == "/") BASE=""; next; }
             { print BASE"/"$1 }
```

Figure 396. AWK File for AIX/ NT odump Shell Script

Appendix B. Special Notices

This publication is intended to help technical support personnel implement systems management using TME 3.0 NT and associated systems management functions on different Intel and UNIX platforms. The information in this publication is not intended as the specification of any programming interfaces that are provided by TME 3.0 NT. See the PUBLICATIONS section of the IBM Programming Announcement for NetFinity for more information about what publications are considered to be product documentation.

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Dept. 600A, Mail Drop 1329
Somers, NY 10589 USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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.

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

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

AIX	AnyNet
AS/400	Client Access
DB2/2	NetFinity
NetView	OS/2
RISC System/6000	RMONitor
System/390	Trouble Ticket

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

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

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

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

Intel	Intel Corporation
NetWare	Novell, Incorporated
Networking	UNISYS, Incorporated
PostScript	Adobe Systems, Incorporated
TME	Tivoli Systems Inc., an IBM Company
TME 10	Tivoli Systems Inc., an IBM Company
X/Open	X/Open Company Limited

Other trademarks are trademarks of their respective companies.

Appendix C. Related Publications

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

C.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How To Get ITSO Redbooks” on page 347.

- *LAN Management Processes Using NetFinity*, SG24-4517
- *Workgroup Management Using SystemView for OS/2*, SG24-2596
- *Systems Management from an NT Server Point of View*, SG24-4723

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

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

C.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RISC System/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RISC System/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection (available soon)	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection (available soon)	SBOF-7250	SK2T-8042

C.3 Other Publications

These publications are also relevant as further information sources.

- *Mastering Windows NT Server 3.51 - Second Edition*, SR28-5688-01
- *Tivoli/Sentry Documentation Kit*, SK2T-6052
- *Tivoli/Management Platform Documentation Kit*, SK2T-6058

How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL http://www.redbooks.ibm.com.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

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

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **ITSO4USA category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	bookshop at dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269	
Canada	1-403-267-4455	
Outside North America	(+45) 48 14 2207	(long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 415 855 43 29 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com
- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

- **Please put me on the mailing list for updated versions of the IBM Redbook Catalog.**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number _____
- Credit card number _____

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

Index

N

O

P

Q

R

S

T

U

V

W

Printed in U.S.A.