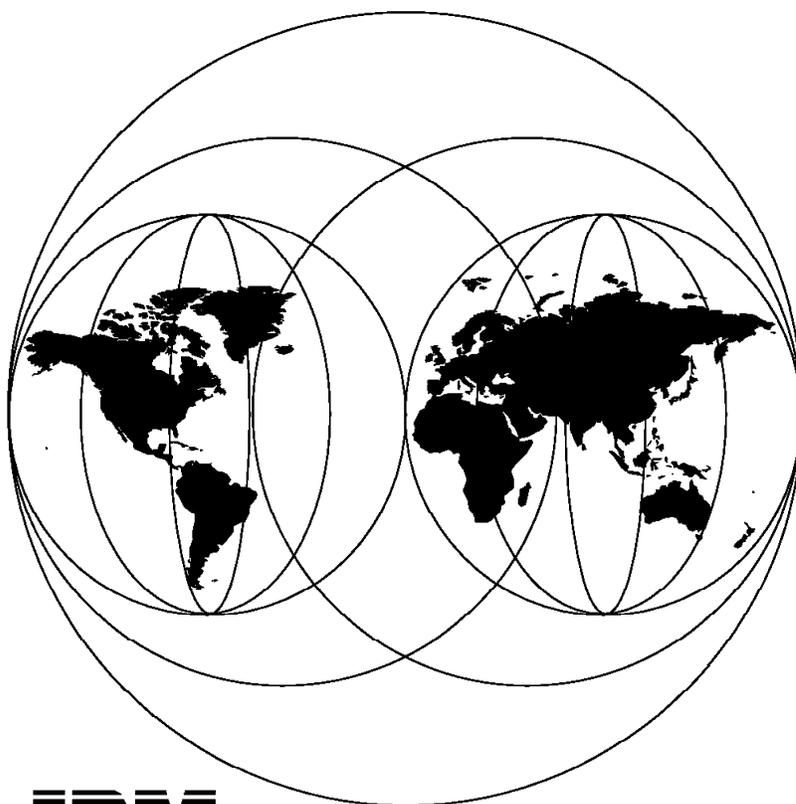# NetFinity V5.0 Command Line and LMU Support

April 1997

**IBM**

**International Technical Support Organization**

**Raleigh Center**

IBM

International Technical Support Organization

# NetFinity V5.0 Command Line and LMU Support

April 1997

**First Edition (April 1997)**

This edition applies to Version 5.0 of NetFinity for use with Windows NT and OS/2 Warp.

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

# Figures

# Preface

This redbook describes the new NetFinity 5.0 command line syntax and shows examples of how to use the new interface. Simple scenarios are provided to show you how to exploit this new interface. In some cases we showed how to use the interface on the local system, and in others how to use it when managing remote clients. The examples will help the reader understand how to use the new command line interface to create scripts to better manage their environment.

As an additional way of providing a management interface for NetFinity 5.0, we have shown several examples of the new features that were added to the Webability service. These new features will help guide the reader through typical management scenarios.

To show the broader implications of the command-line interface, we provided some examples of using the new MIB values to interact with an SNMP manager on the AIX platform.

Finally, we showed several examples of using the command-line interface in situations where you might have used LMU in the past. This should help with migration efforts from LMU to NetFinity 5.0. In addition to using LMU functions, we also incorporated the new SPM/2 functions for performance monitoring.

## 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 four years ago, he worked in Professional Services in the United States as a National Communications Specialist.

**Markus Klein** is a software support specialist for IBM in Germany. He has experience in defect support for systems management software on OS/2 and Windows NT. He holds a degree in Physics from Johannes Gutenberg University in Mainz. His areas of expertise include OS/2, Windows (NT, 95, 3.11) and some UNIX-related networking and network management software.

**Thomas Strauss** is a system software specialist in Germany. He has 10 years of experience in IBM HW and 10 years in SW. He has worked at IBM for 25 years. His areas of expertise include VM, VTAM, NCP, Multiprotocol Networking, OS/2 and all OS/2 systems near SW.

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

Joe Loewengruber and
Mike Coleman
IBM RTP Development

# Comments Welcome

**Your comments are important to us!**

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 299 to the fax number shown on the form.

- Use the electronic evaluation form found on the Redbooks Home Pages at the following URLs:

  For Internet users              `http://www.redbooks.ibm.com`
  For IBM Intranet users          `http://w3.itso.ibm.com/redbooks`

- Send us a note at the following address:

      `redbook@vnet.ibm.com`

# Chapter 1. Project Overview

This chapter provides an overview of this redbook and the systems environment, as well as an overview of the new functions in NetFinity 5.0 and a quick look at the installation. For a review of the functions already implemented in TME 10 NetFinity 4.0 please see the following redbooks:

- *LAN Management Processes (Alerts/Monitoring) Using NetFinity*, SG24-4517

- *Systems Management from an NT Server Point of View*, SG24-4723

- *NetFinity V5.0 Database Support*, SG24-4808

## 1.1 New Functions

NetFinity Version 5.0 planned features include the following:

- New platforms:

  - 32-bit Windows 95 version

    **Note:** No Windows 16-bit manager.

- Functional enhancements:

  1. Global command line support

  2. SNA protocol LU 6.2 for OS/2 platform

  3. Console takeover/DCAF:

     - NT, Win95, OS/2 console
     - NT, Win95, OS/2, Win3.1 client
     - Win 3.1 performance improvements
     - Win 3.1 footprint improvement - 8 MB memory
     - Support all OS-supported video modes

  4. TME 10 integration:

     - HW/SW Inventory to Courier
     - Monitor data to Sentry
     - Alerts to TEC (Tivoli Event Control) events

  5. Mass Administration:

     - Group password change/presence check
     - Alert manager changing thresholds
     - Add/change critical file monitor thresholds
     - Process manager alerts management
     - Security settings
     - Scheduled monitor database export
     - Group reboots/keyword management

  6. Lotus Notes monitors/alerts

  7. Web help links

  8. Web Guru help engine

  9. SNMP/DMI:

     - DMI Server MIF (including RAID)
     - Instrument SNMP MIB
       - Create MIB

- SNMP Agent extension
  - NetFinity-specific information
    - Remote System Manager (in 4.0)
    - Alert Manager log
    - Monitors (current values) (in 4.0)
    - Software Inventory

10. Serial alert port sharing

11. IDE PFA - SMART, All platforms except for NetWare, Win31 (no drivers)

12. NIC monitors

13. Online/Offline bitmaps

14. SysInfo MAC address

15. ODBC for OS/2

16. File transfer enhancements

17. Tiled monitors

18. Software inventory - scheduled search by product type

- Additional items:

  - NETFBASE hidden from task list on OS/2

  - Default groups provided (All and serial)

  - Software inventory - selective volume search

  - Unique SNMP trap IDs - available on BBS

  - Coroner log for failed services/alert actions

  - NetWare TCP/IP support - complete, available on BBS

  - Power down(Win95 APM)/Shutdown(Win3.1/NW)

  - Uninstall

  - Scheduled RAID synchronization

  - NT System Partition Access

  - Disk space percentage monitors

  - Monitor export - definable intervals

  - Scheduled monitor export

  - Scheduled Wake-On-LAN

  - SysInfo CRT VPD info - Win 95 only with limitations documented

  - Scheduled power down/shutdown

  - NF Service manager detail view

  - Identification of rack configurations

- Command line support

  The most important new function is the integration of a command line interface into NetFinity 5.0. The previous versions of the product lacked a command line interface. Even though they had very good GUIs, there were only a few things you could do using the command line. Now you can handle anything from the command line that you could do from the GUI before, plus additional functions that were added to help with the

coexistence of LMU. This will also drive a lot of automation possibilities. The NetFinity functions with this new interface are:

- Remote System Manager
- Security Manager
- System Profile
- Alert Manager
- System Monitor
- Process Manager
- System Information (extended functionality)

- New platform - Windows 95 32-bit

- Console takeover / DCAF

  The Remote Workstation Control that was integrated in NetFinity 5.0 Server, but not in TME 10 NetFinity V.4.0, is now part of the product.

- Mass administration

  With the new command line interface it is much easier to administer a larger number of clients.

- Enhanced Webability

## 1.2 Quick Installation Guide - Service

As a quick guide for those new to NetFinity 5.0 or as a reminder for those familiar with the product, here is how the installation runs. In our example, we installed the NetFinity Services for Windows 95.

The installation is started by running the netfinst.exe program either from the first (Manager or Services) diskette or from the correct directory on the CD-ROM. The first thing you will be asked is to enter the source and the target directories.



Directory to Install from:

A:\

Directory to Install to:

C:\WNETFIN

OK          Cancel          Help
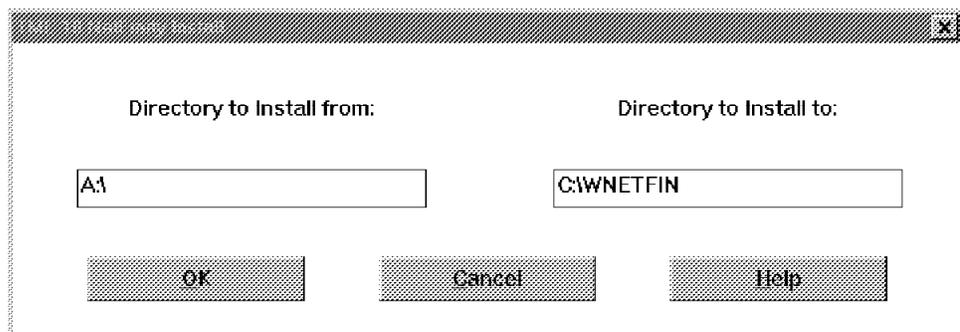
*Figure 1. Entering Source and Target Directories for the Installation*

You can then select the type of managers or clients you want to install:

- NetFinity 5.0 Managing Systems:

  1. NetFinity 5.0 Manager Installation: Installation of the NetFinity 5.0 Managing system.

  2. NetFinity 5.0 Manager Installation with Web Enhancement: Installation of the NetFinity Manager plus management using Webability function.

- NetFinity 5.0 Client Systems:

  1. Stand-Alone Operation (no network support): Managing only the local workstation.

  2. Passive Client Operation: This workstation can only be managed from a remote management station.  No GUIs will be installed.

  3. Active Client Operation: The workstation can be managed either locally or remotely.  All of the GUIs will be installed locally.
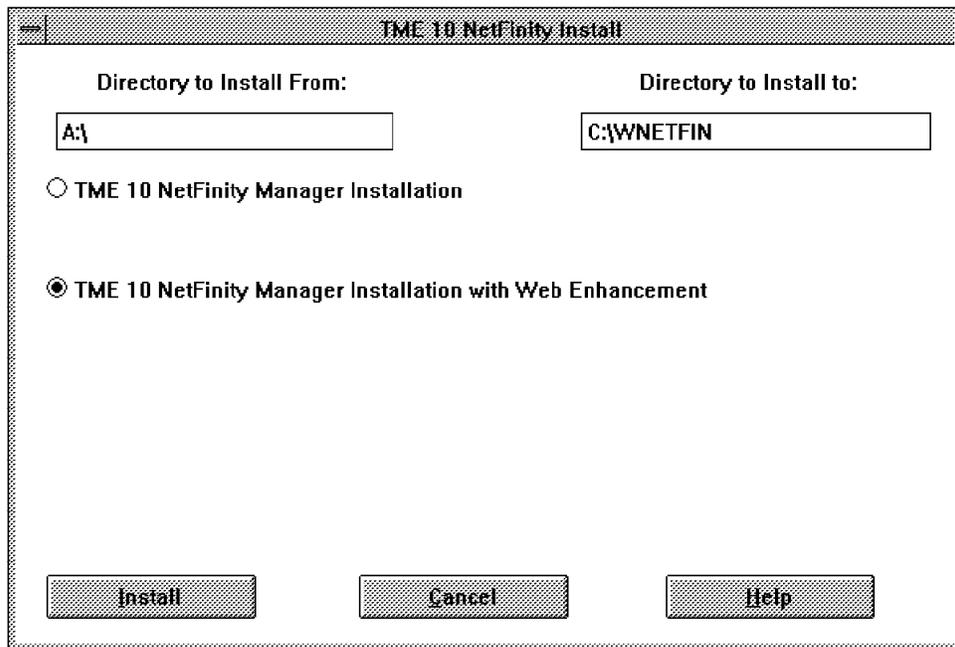


*Figure 2.  Selecting the Type of Manager*

Selecting to have the Web enhancements installed only installs the code.  You will need to perform minor customization after you start up NetFinity 5.0.
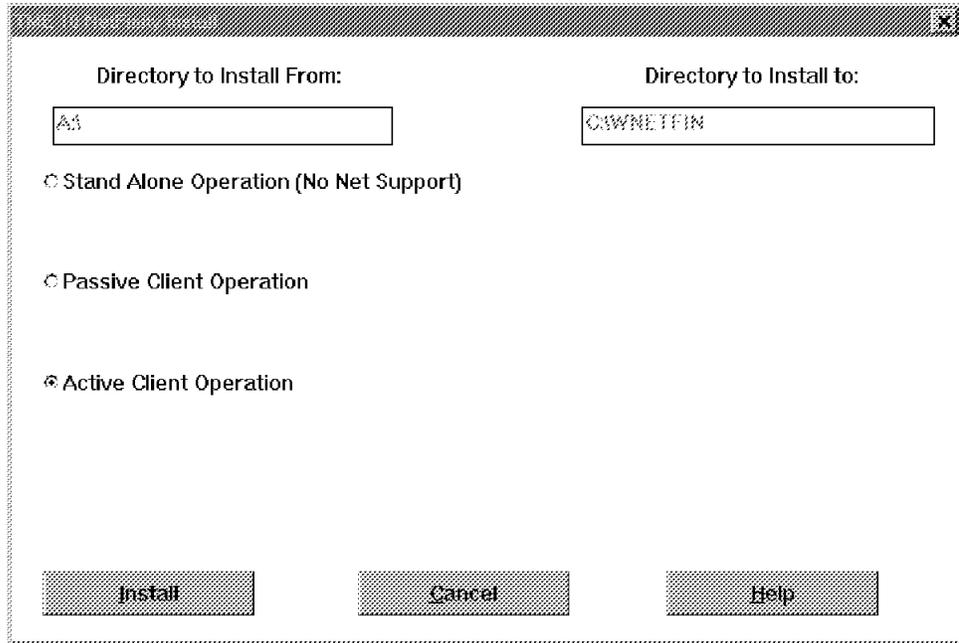
*Figure 3. Selecting the Type of Client*

The following window will appear as the installation progresses:
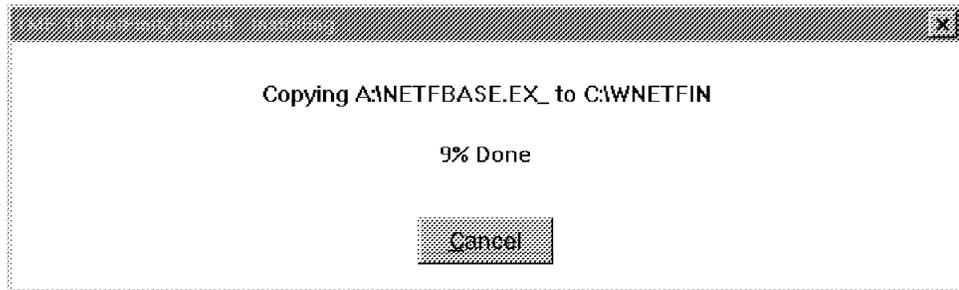


*Figure 4. Installation Progress Window*

After the installation of the code was completed on our Windows 95 system, we were prompted with an option for updating our system files. The next figure warns you that system files are going to be updated. You should note where the old copy is saved.
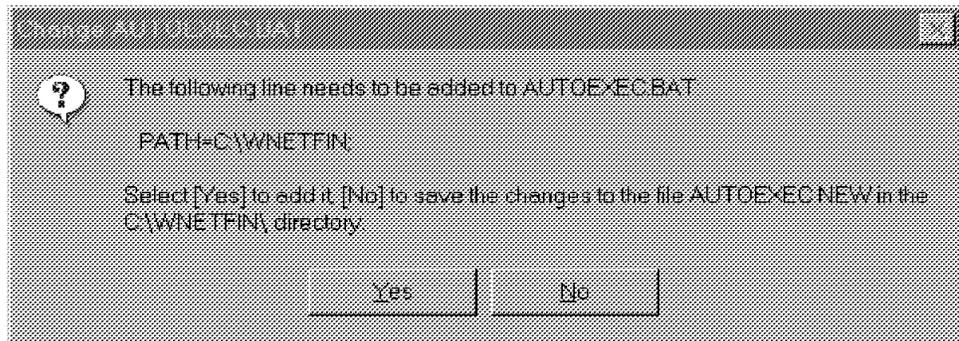


*Figure 5. Updating the autoexec.bat*

One of the most important things to think about is that after the installation is done, your first step after the IPL should be to enable the security. Otherwise this workstation is wide open for anyone to access.

As the final step in the installation you can enable the different transport protocols that can be used to access this workstation. You will only get a list of the protocols that NetFinity 5.0 found on the machine, so if NetBIOS is not installed, it wouldn't be in the list. You can also add some system keywords later on to group several stations together.
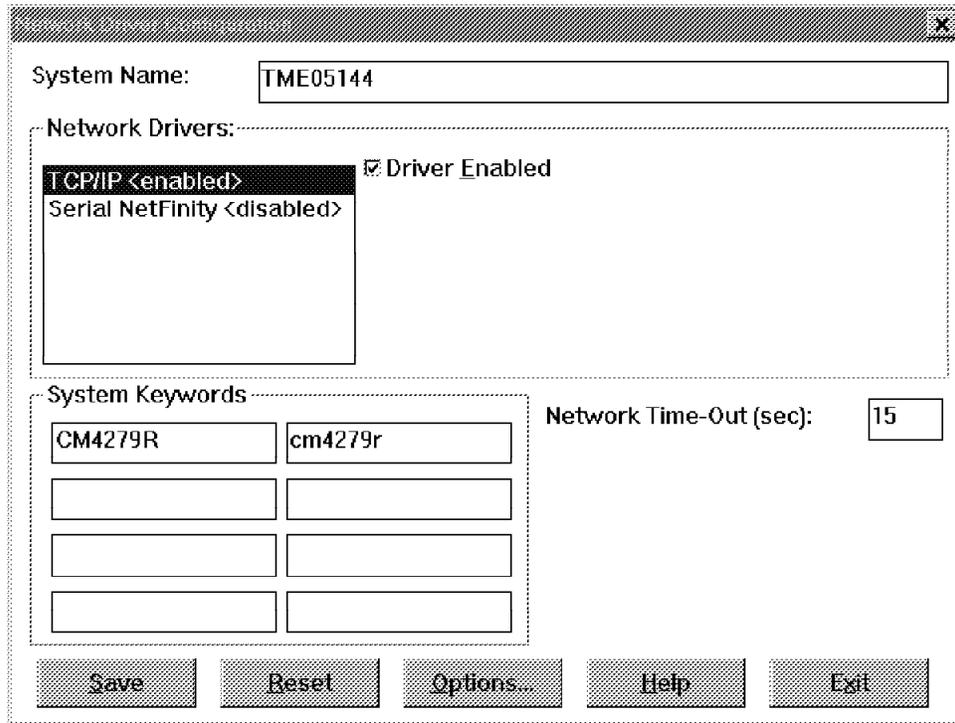


*Figure 6. Configuring the Protocols*

After the installation is complete you should have a new group/folder which looks very similar to this one:
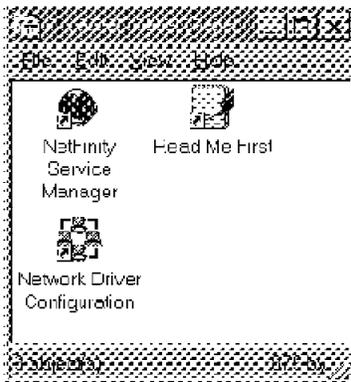


*Figure 7. The NetFinity 5.0 Folder*

Double-clicking on the **NetFinity Service Manager** will result in the following window:
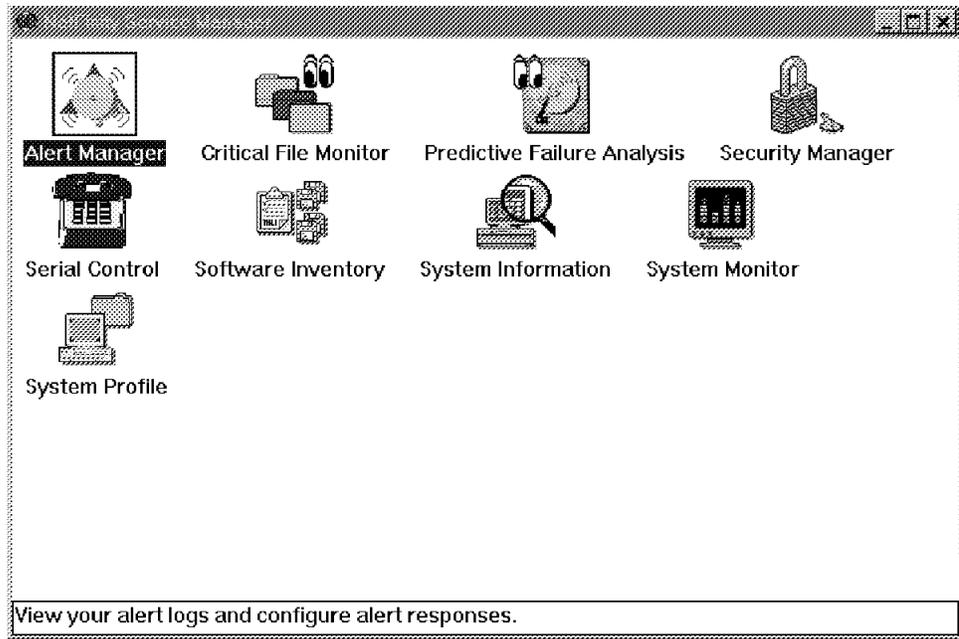
Figure 8. The NetFinity Services for Windows 95 Main Window

# Chapter 2. Managing Clients from the Command Line

This chapter provides examples of managing the following operating systems using NetFinity on the NT Platform:

- NT V3.51 Workstation

- OS/2 Warp Connect V3.0

- Windows 95

- OS/2 Warp V4.0 (Merlin)

The following NetFinity modules now have command line functions:

- Remote System Manager

- Security Manager

- System Profile

- Alert Manager

- System Monitors

- Critical File Monitor

- Process Manager

- System Information Tool

- System Configuration Manager

---
**Note**

- If you run a command from an OS/2 machine on a Win95 machine and get an rc=201, add `command /c` before the command.

- nfproccl - The system name in the /n parameter is case-sensitive.

---

## 2.1  Overview of the Command Line Functions

This chapter provides an overview of the new command line functions in NetFinity 5.0 by showing a short example of the command and some of its parameters.

### 2.1.1  Remote Systems Manager - NFRSYSCL.EXE

An example of all the parameters for the nfrsyscl command follows:

```
The following command-line operations are supported:
  /?           - Returns this command-line help
  /GETGRP      - Returns information about selected groups
  /GETSYS      - Returns information about selected systems
  /RUNSYS      - Executes another tool targeted to selected systems
  /DELSYS      - Delete selected systems
  /ADDSYS      - Create a new system
  /EDITSYS     - Modify attributes of selected systems
  /DELGROUP    - Delete selected groups
  /ADDGROUP    - Create a new group
  /EDITGRP     - Modify attributes of selected groups
  /BOOTSYS     - Restart selected systems
  /SHUTSYS     - Shutdown selected systems
  /WAKESYS     - Attempt wake-on-LAN on selected systems
  /PWRSYS      - Power-down selected systems
  /RESETERRSYS - Reset error conditions on selected systems
  /DODISC      - Initiate discovery in selected groups
  /DOPING      - Initiate presence check on selected systems
  /GETSYSKWD   - Get keywords from selected systems
  /SETSYSKWD   - Set keywords on selected systems
----------------------------------------------------------------
Select systems using /ALL, /SYS:sysname, /SYSTAG:tag,
                     /GRP:grpname, or /GRPTAG:grptag
Select groups using  /ALL, /GRP:grpname, or /GRPTAG:grptag
```

### 2.1.1.1 Remote System Manager Command Line Interface Example

After the installation of NetFinity 5.0 a default group named All is already defined, as can be seen in Figure 10 on page 12. This group has the following settings:

```
[C:\netfin]nfrsyscl /getgrp /grp:All
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ GRPTAG=0x000221E5, GRPNAME="All", COMBO=ALL,
  KWD={  },
  OS_MASK={  }, AUTODISC=DISABLED,
  PROTO_MASK={  },
  DEFONLN=DISABLED, DEFOFFLN=DISABLED, DEFPING=NONE
}
```

---
**Note**

Group names are case-sensitive.  All is not the same as ALL.

---

As you can see, the group has a group tag (GRPTAG) assigned to it, which is unique on the managing system.  All systems have a tag assigned to them.  The combination describes which type of search will be started when discovering the systems in the group.  The possible COMBO settings are ALL, ANY or ONE.  The values for keywords (KWD), selected operating systems (OS_MASK) and protocols (PROTO_MASK) are empty by default.  As you update those fields through either the command line, or the GUI, you will see new values appear there.  In addition, initially Autodiscovery (AUTODISC), online and offline notification (DEFONLN and DEFOFFLN) are disabled and the default ping interval (DEFPING) is set to NONE.

When you start to discover all systems in this group, you will find that every NetFinity 5.0 Manager or Services machine in your LAN will show up with one icon for each of the transport protocols it has that has been enabled in its

configuration menu. For example, in the following figure, you can see that
TME05144 can be managed over TCP/IP as well as NetBIOS.



*Figure 9. Remote System Manager - Detail View of a Group*

In order to have a list of all reachable systems, we recommend that you do not
change or delete this default group. Therefore, the first thing to do is create a
new group for your specific management needs. If the group you add has a
space in the name, as is shown in the following example, you will need to put
double quotes around the entire name. This is only true when adding the group
using the command line. If you add it using the GUI, you will not need to type
the quotes.

```
[C:\netfin]nfrsyscl /addgroup:"Windows Clients"
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
GRPTAG=0x042E747F

[C:\netfin]nfrsyscl /getgrp /grp:"Windows Clients"
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ GRPTAG=0x042E747F, GRPNAME="Windows Clients", COMBO=ALL,
  KWD={  },
  OS_MASK={  }, AUTODISC=DISABLED,
  PROTO_MASK={  },
  DEFONLN=NONE, DEFOFFLN=NONE, DEFPING=NONE
}
```

If you take a look at the Remote System Manager GUI after issuing the above
command, you will see that the new group is added.

*Figure 10. Remote System Manager - Defined Groups*

To show the default settings, we only defined a name for the group. Of course we need to change some of the settings. The keywords are not case-sensitive. Examples of the keywords are:

- /combo
- /addkwd
- /autodisc
- /addos
- /addproto

We updated our default entry with the following command:

```
[C:\netfin]nfrsyscl /editgrp /grp:"Windows Clients" /COMBO:ANY /ADDKWD:Windows
/AUTODISC:10 /ADDOS:NETWARE /ADDOS:OS2 /ADDPROTO:TCPIP
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
GRPTAG=0x042E747F

[C:\netfin]nfrsyscl /getgrp /grp:"Windows Clients"
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ GRPTAG=0x042E747F, GRPNAME="Windows Clients", COMBO=ANY,
  KWD={ "Windows" },
  OS_MASK={ OS2, NETWARE }, AUTODISC=10,
  PROTO_MASK={ TCPIP },
  DEFONLN=NONE, DEFOFFLN=NONE, DEFPING=NONE
}
```

Using the GUI, click on the group icon to see what changes have occurred.
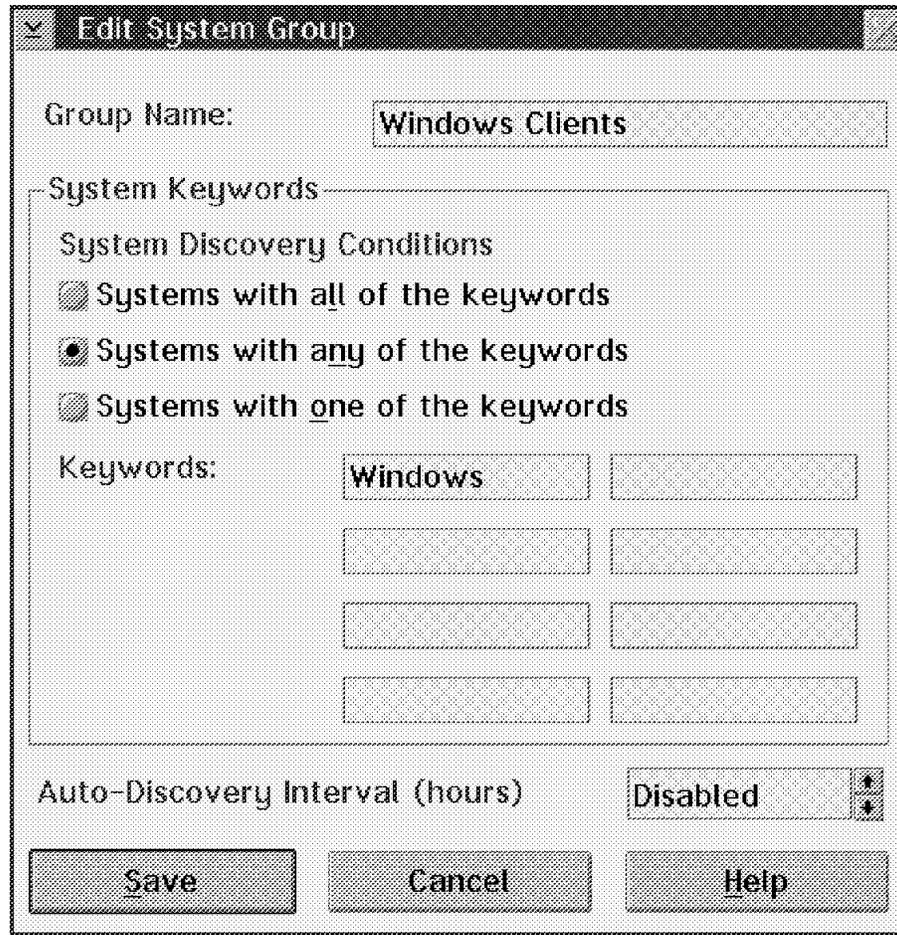
*Figure 11. Remote System Manager - Group Definitions*

Now let's take a look how many systems in our network are in this newly defined group using the dodisc parameter.

```
[C:\netfin]nfrsyscl /dodisc /grp:"Windows Clients"
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
GRPTAG=0x042E747F

[C:\netfin]nfrsyscl /getsys /grp:"Windows Clients"
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
```

The /getsys command showed us no systems discovered. This means that there are no systems out on our LAN that match the given criteria for this group. Therefore, we might change our definitions for the group.

```
[C:\netfin]nfrsyscl /editgrp /grp:"Windows Clients" /ADDKWD:"SG24-4925"
/DELPROTO:TCPIP
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
GRPTAG=0x042E747F

[C:\netfin]nfrsyscl /getgrp /grp:"Windows Clients"
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ GRPTAG=0x042E747F, GRPNAME="Windows Clients", COMBO=ANY,
  KWD={ "Windows", "SG24-4925" },
  OS_MASK={ OS2, NETWARE }, AUTODISC=10,
  PROTO_MASK={  },
  DEFONLN=NONE, DEFOFFLN=NONE, DEFPING=NONE
}
```

We added another keyword and deleted the protocol mask. Now let's see if we have any matches.

```
[C:\netfin]nfrsyscl /dodisc /grp:"Windows Clients"
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
GRPTAG=0x042E747F

[C:\netfin]nfrsyscl /getsys /grp:"Windows Clients"
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ SYSTAG=0x00490943, SYSNAME="TME05185", PROTO=TCPIP,
  ADDR="NTSERVB.itso.ral.ibm.com",
  ONLINE=TRUE, ONLN=DISABLED, OFFLN=DISABLED, PING=600,
  GRPLIST={ 0x000221E5, 0x042E747F },
  ERRORCOND={  },
  OS=WINDOWS_NT, OS_VER=3.51, MANAGER, SERVER
}
{ SYSTAG=0x00490309, SYSNAME="TME05185", PROTO=NETBIOS,
  ADDR="WTR05185",
  ONLINE=TRUE, ONLN=DISABLED, OFFLN=DISABLED, PING=600,
  GRPLIST={ 0x000221E5, 0x042E747F },
  ERRORCOND={  },
  OS=WINDOWS_NT, OS_VER=3.51, MANAGER, SERVER, MAC=855100520040
}
```

The result is that we discovered a single system that has two transport protocols. One is over NetBIOS and the other is TCP/IP. Note that the SYSTAG for each is different. This means that you can access the workstation either way. If there is a problem (performance or connectivity) over one transport protocol, you still have the capability to access it over the other one. You can also see that both are members of two different groups. The groups are for Windows clients and members of the group called ALL. Another important part of the output is the ONLINE= parameter. It shows if a system is online (TRUE) or offline (FALSE). The last line gives some information about the operating system and what kind of NetFinity 5.0 is installed.

Other then deleting a group, you now have seen the different ways to manipulate a NetFinity 5.0 group from the command line. The next thing to do is work with the systems that we can discover. In order to do that, we deleted one of the two systems in our group and rebuilt it back up to its current status. As we showed earlier, we can address the systems one after the other by using its unique SYSTAG.

```
[C:\netfin]nfrsyscl /delsys /systag:00490309
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
SYSTAG=0x00490309

[C:\netfin]nfrsyscl /getsys /grp:"Windows Clients"
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ SYSTAG=0x050F97C0, SYSNAME="TME05185", PROTO=TCPIP,
  ADDR="NTSERVB.itso.ral.ibm.com",
  ONLINE=TRUE, ONLN=DISABLED, OFFLN=DISABLED, PING=600,
  GRPLIST={ 0x000221E5, 0x042E747F },
  ERRORCOND={   },
  OS=WINDOWS_NT, OS_VER=3.51, MANAGER, SERVER
}
```

The NETBIOS representation of our system is now deleted, but the TCPIP system
is still in the group.  To get the system back, we use the /ADDSYS parameter as
shown in the following example:

```
[C:\netfin]nfrsyscl /addsys:"TME05185" /PROTO:NETBIOS /ADDR:WTR05185
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
ERROR: ADDSYS command requires one ADDGRP or ADDGRPTAG parameter

[C:\netfin]nfrsyscl /addsys:"TME05185" /proto:NETBIOS /addr:WTR05185 /addgrp:All
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
SYSTAG=0x050F22B0
```

---
**Note**

When a new system is to be added to a NetFinity 5.0 group, the system must
be online when it is added, otherwise /addsys ends with an error message.

---

The configuration information that is contained in the fields for keywords,
protocols and other parameters are tied to the remote system.  When we added
the system back into our database, it created all new entries for it.  The SYSTAG
value will also be new.  If you had chosen to use that SYSTAG value as part of
some REXX program or other routine, you would have to update all those
routines after you add it back in.  We now have to put the system into the
Windows clients group again.

```
[C:\netfin]nfrsyscl /editsys /sys:"TME05185" /addgrp:"Windows Clients"
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
SYSTAG=0x050F97C0
SYSTAG=0x050F22B0
```

We then selected the system by its system name.  Therefore, the command will
be executed for both of its representations.  Checking the group again shows
that we are back to where we started.

```
[C:\netfin]nfrsyscl /getsys /grp:"Windows Clients"
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ SYSTAG=0x050F97C0, SYSNAME="TME05185", PROTO=TCPIP,
  ADDR="NTSERVB.itso.ral.ibm.com",
  ONLINE=TRUE, ONLN=DISABLED, OFFLN=DISABLED, PING=600,
  GRPLIST={ 0x042E747F, 0x000221E5 },
  ERRORCOND={  },
  OS=WINDOWS_NT, OS_VER=3.51, MANAGER, SERVER
}
{ SYSTAG=0x050F22B0, SYSNAME="TME05185", PROTO=NETBIOS,
  ADDR="WTR05185",
  ONLINE=TRUE, ONLN=DISABLED, OFFLN=DISABLED, PING=600,
  GRPLIST={ 0x000221E5, 0x042E747F },
  ERRORCOND={  },
  OS=WINDOWS_NT, OS_VER=3.51, MANAGER, SERVER, MAC=855100520040
}
```

In addition to this, there are some other commands to manipulate systems.

/RUNSYS starts a NetFinity 5.0 command line procedure on a remote machine.

```
   You need to make sure that you have access to use that service
for the ID that you are connected with.  The default ID would be public
and if you have turned off all services, you will not be able to use
the RUNSYS command.
[C:\netfin]nfrsyscl /runsys /systag:050F22B0 /ACTION "nfsmoncl /GETMON
/MONNAME:"CPU Utilization"
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ SYSTAG=0x050F22B0, SYSNAME="TME05185", PROTO=NETBIOS,
  ADDR="WTR05185",
  ONLINE=TRUE, OUTPUT={
NetFinity System Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ MONITOR_ID=0xAB348194, NAME="CPU Utilization",
  SAMPLE=5000, UNITS_LBL="Percent", RECORDING=DISABLED,
  MINVAL=0.000000, MAXVAL=100.000000, VALUE=0.801889
}
}, RC=0
}
```

In this example we are using the /RUNSYS option to get the value for the CPU Utilization monitor of a remote system. It is important to note that you must make sure that the machine is reachable over the transport protocol. In addition, you can use any command, not just NetFinity 5.0 commands. A very simple command to see the version of NT Server we were using and its service pack level would be:

nfrsyscl /runsys /systag:050f22b0 /action winver

---
**Important**

If you run a command from an OS/2 machine on any type of Windows client, you have to add COMMAND /C in front of the command, because you need to specify the shell where the /RUNSYS is executed.

---

/RESETERRSYS resets the error conditions on all selected systems.

All other options are related to restarting a remote system:

- /bootsys reboots a system.
- /shutsys shuts down a system.
- /pwrsys powers down the selected system.
- /wakesys performs a Wake-on-LAN function.

## 2.1.2  Security Manager - NFSECCL.EXE

The parameters that are associated with the nfseccl command are listed below:

```
The following command-line operations are supported:
  /?      - Returns this command-line help
  /LISTIN  - Returns information about inbound user-ids
  /ADDIN   - Defines a new inbound user-id
  /DELIN   - Delete an inbound user-id
  /LISTOUT - Returns information about outbound host-ids
  /ADDOUT  - Defines a new outbound host-id
  /DELOUT  - Delete an outbound host-id
  ---------------------------------------------------------
Select inbound user IDs using /ALL or /USERID:uid
Select outbound node-ids using /ALL or /NODE:nodeid
```

### 2.1.2.1  Security Manager Command Line Interface Examples

The function of the Security Manager is to manage incoming and outgoing user ID/password combinations.  In our examples we added, edited and deleted user IDs and passwords.

```
[C:\netfin]nfseccl /listin /all
NetFinity Security Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ USERID="<PUBLIC>", PWD="*", SVC={   }
{ USERID="MAKLEIN", PWD="*******", SECMGR, SVC={ ALL }
{ USERID="BLSTRAS", PWD="*******", SECMGR, SVC={ ALL }
{ USERID="JJACOB", PWD="*******", SECMGR, SVC={ ALL }

[C:\netfin]nfseccl /listout /all
NetFinity Security Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ HOST="<DEFAULT>", USERID="<PUBLIC>", PWD="*" }
{ HOST="NTSERVB.itso.ral.ibm.com", USERID="MAKLEIN", PWD="*******" }
{ HOST="TMECLI1.itso.ral.ibm.com", USERID="THOMAS", PWD="*******" }
{ HOST="WTR05185", USERID="MAKLEIN", PWD="*******" }
{ HOST="9.24.104.100", USERID="MAKLEIN", PWD="*******" }
```

> **Note**
>
> We crossed out all passwords, but be aware that these commands show them all without any encoding.  This is another reason to strictly control the access to the Security Manager.

The following is an example of how to add a new user ID and password for both inbound and outbound commands.

```
[C:\netfin]nfseccl /addin:"MAGGUS" /pwd:"NCC1701D" /ADDSECMGR
/ADDSVC:ALL
NetFinity Security Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.

[C:\netfin]nfseccl /addout:"TME05185" /setuid:"MAGGUS" /pwd:"NCC1701D"
NetFinity Security Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
```

The double quotes around the user ID or password are not needed if there is no space in the text. The user ID and the password are case-sensitive when entered from the command line, but always uppercase when entered in the GUI. This leads to problems when editing/deleting user IDs created from the command line in the GUI. Our recommendation is to use uppercase only.

Looking at the GUI, you find the user ID/password combination added to the lists.
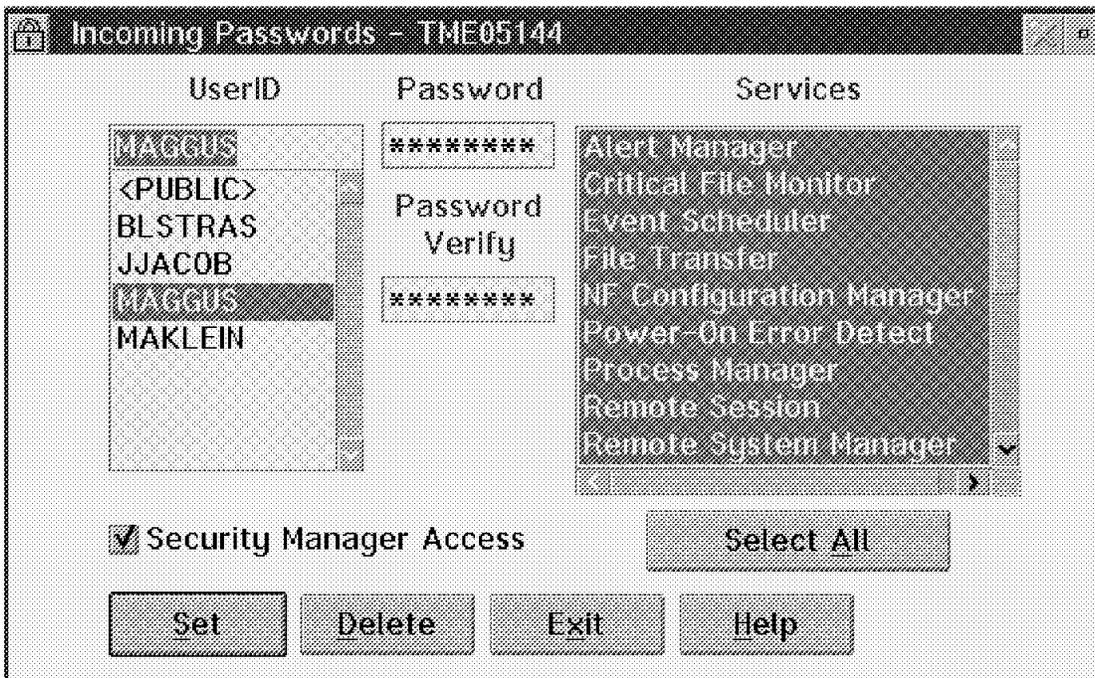


*Figure 12. Security Manager - Incoming Passwords*

```
┌──────────────────────────────────────────────────────────────┐
│ 🔒  Outgoing Passwords — TME05144                        ▨  □  │
├──────────────────────────────────────────────────────────────┤
│ Select to edit an entry.                                     │
│                                                              │
│ Network Address            UserID                            │
│ ┌──────────────────────────────────────────────────────┐─┐  │
│ │ <DEFAULT>                  <PUBLIC>                   │▨│  │
│ │ NTSERVB.itso.ral.ibm.com   MAKLEIN                    │ │  │
│ │ TMECLI1.itso.ral.ibm.com   THOMAS                     │ │  │
│ │ WTR05185                   MAKLEIN                    │ │  │
│ │ 9.24.104.100               MAKLEIN                    │ │  │
│ │ TME05185                   MAGGUS                     │ │  │
│ │                                                      │ │  │
│ │                                                      │▨│  │
│ └──────────────────────────────────────────────────────┘─┘  │
│                                                              │
│   ┌───────┐  ┌─────────┐  ┌───────┐  ┌───────┐               │
│   │  Add  │  │ Delete  │  │  Exit │  │  Help │               │
│   └───────┘  └─────────┘  └───────┘  └───────┘               │
└──────────────────────────────────────────────────────────────┘
```
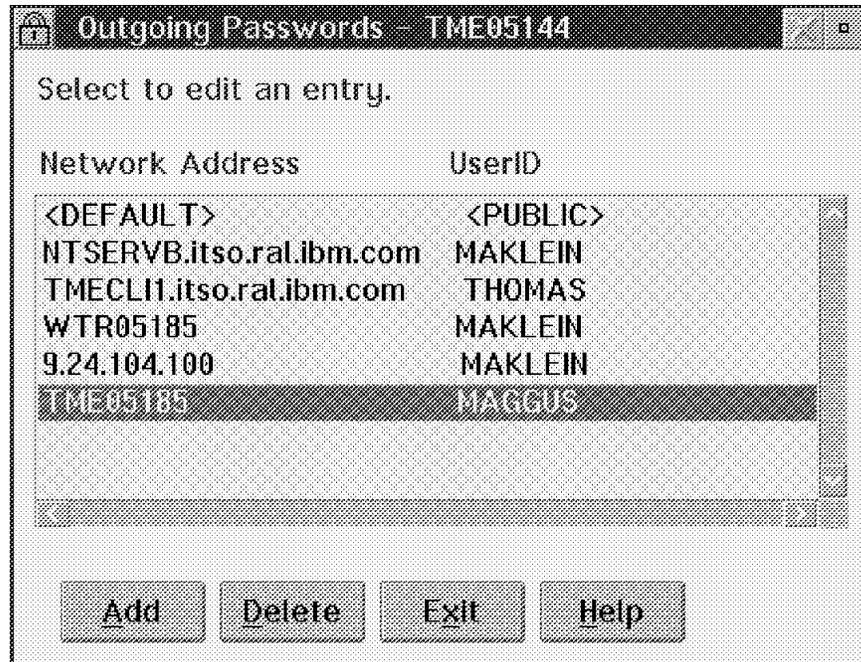
*Figure 13. Security Manager - Outgoing Passwords*

In addition, existing user ID/password couples can be modified with the edit
function.

```
[C:\netfin]nfseccl /editin /userid:"MAGGUS" /DELSECMGR /DELSVC:ALL
NetFinity Security Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
```

This removed all rights from the user ID MAGGUS. Now let's give it something
back:

```
[C:\netfin]nfseccl /editin /userid:"MAGGUS" /ADDSVC:"AlertMgr" /ADDSVC:"CFMBase"
NetFinity Security Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
```

The GUI shows us that things are as we want them to be.
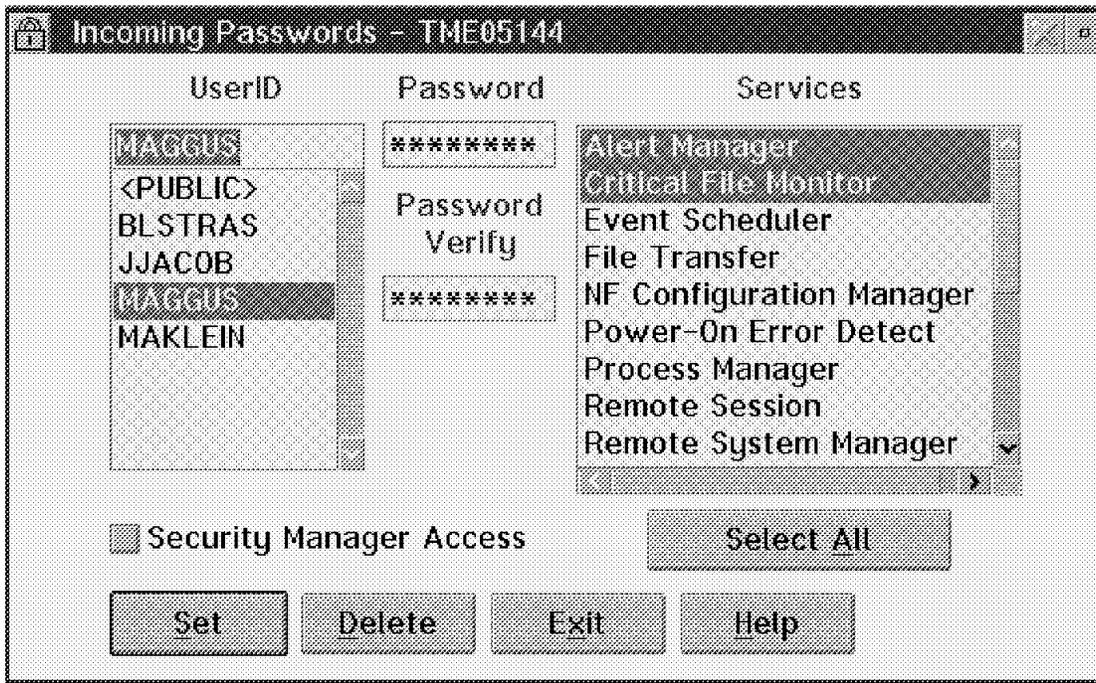
*Figure 14. Security Manager - Incoming Passwords after Manipulation*

Finally, to clean up the ID we delete the created user ID/password combinations.

```
[C:\netfin]nfseccl /delin /userid:"MAGGUS"
NetFinity Security Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.

[C:\netfin]nfseccl /delout /host:"TME05185"
NetFinity Security Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
```

If you had the GUI open on the desktop, the user ID will still appear there. You need to close the Security Manager window, and then re-open it to see the true state of all the user IDs.

### 2.1.3 System Profile - NFPROFCL.EXE

The parameters that are associated with the nfprofcl command are listed below:

```
The following command-line operations are supported:
  /?                       - Returns this command-line help
  /GET:attr                - Returns an attribute's value
  /GETALL                  - Returns all attributes and their values
  /SET:attr /SETVAL:val    - Set attribute to value
  /SETMANY /INPUT:filename - Set attributes to values listed in input file
  /N:netpath               - Optional network path to target system
  /S:sysname               - Optional system name for target system
```

### 2.1.3.1 Examples for the System Profile Command Line Interface

The system profile stores some vital product data (VPD) for the local machine. This example shows how to access and edit this information.

```
[C:\netfin]nfprofcl /getall
NetFinity System Profile Command-Line Tool
Copyright (C) IBM Corp. 1996.
sysmodelname=NULL
sysmodelnum=NULL
sysserialnum="9577DNG23TKKLB  "
boardserialnum=" B10Z63950AR"
procserialnum=NULL
⋮
firstname=NULL
⋮
lastname=NULL
⋮
```

This gives us a lengthy list of all the attributes that can be set. Let's add some data.

```
[C:\netfin]nfprofcl /set:sysmodelname /setval:"my PC"
NetFinity System Profile Command-Line Tool
Copyright (C) IBM Corp. 1996.

[C:\netfin]nfprofcl /get:sysmodelname
NetFinity System Profile Command-Line Tool
Copyright (C) IBM Corp. 1996.
sysmodelname="my PC"
```

We could add every single attribute with a command like that, but there is an easier way. If you create a file like this:

```
attribs.txt:

sysmodelname="My PC"
sysmodelnum="0815"
sysserialnum="9577DNG23TKKLB  "
boardserialnum=" B10Z63950AR"
procserialnum="0123456789"
firstname="Markus"
lastname="Klein"
```

*Figure 15. Profile Attributes*

It is possible to get all this information into the system profile with a single command.

```
[C:\netfin]nfprofcl /setmany /input:attribs.txt
NetFinity System Profile Command-Line Tool
Copyright (C) IBM Corp. 1996.
```

The system profile now looks like this:

```
[C:\netfin]nfprofcl /getall
NetFinity System Profile Command-Line Tool
Copyright (C) IBM Corp. 1996.
sysmodelname="My PC"
sysmodelnum="0815"
sysserialnum="9577DNG23TKKLB  "
boardserialnum=" B10Z63950AR"
procserialnum="0123456789"
sysdatepurch=NULL
dispmodelnum=NULL
dispserialnum=NULL
dispdatepurch=NULL
prnmodelnum=NULL
prnserialnum=NULL
prndatepurch=NULL
modemmodelnum=NULL
modemserialnum=NULL
modemdatepurch=NULL
firstname="Markus"
middlename=NULL
lastname="Klein"
⋮
```

In the GUI we also find the new entries.
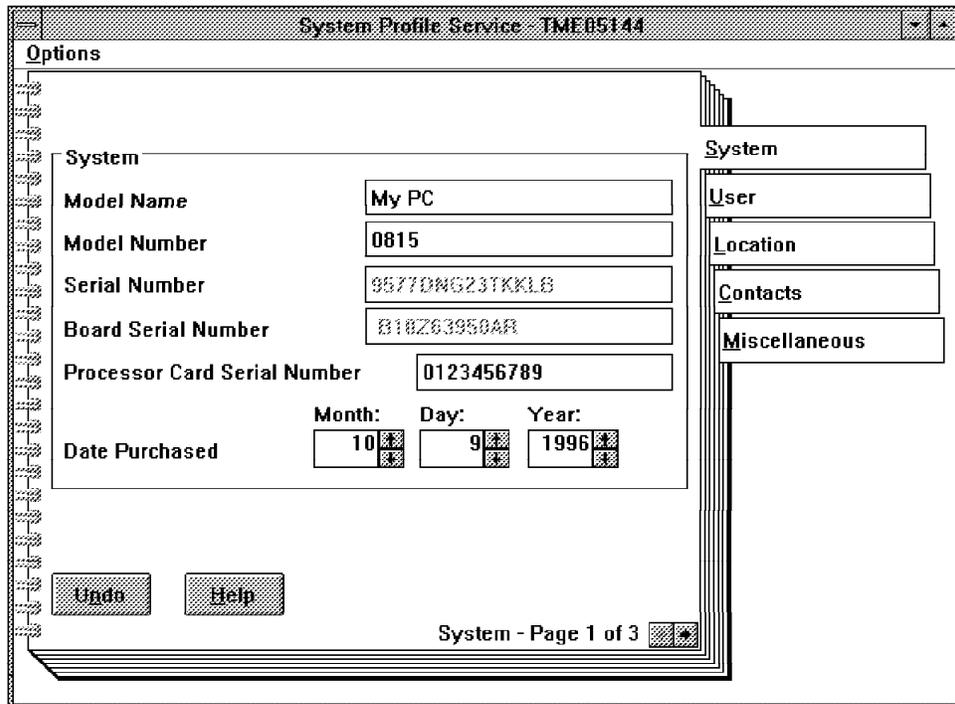


Figure 16. System Profile - System Page
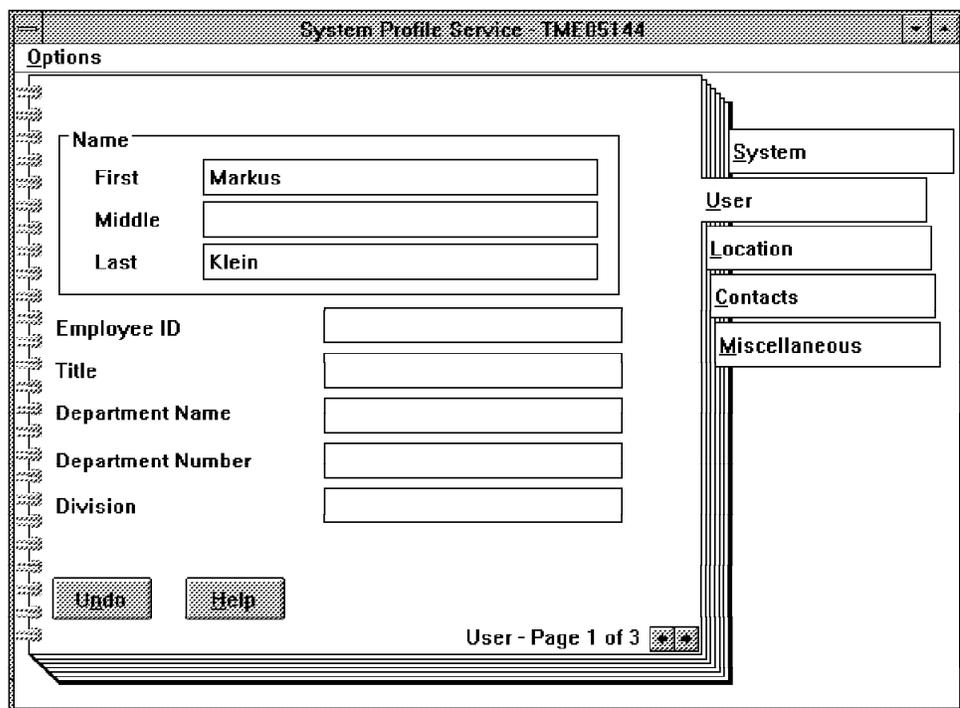
*Figure 17. System Profile - User Page*

## 2.1.4 Alert Manager - NFALRTCL.EXE

The parameters that are associated with the nfalrtcl command are listed below:

```
The following command-line operations are supported:
   /LISTLOG     - List alerts from alert log
   /DELLOG      - Delete alerts from alert log
   /LISTHAND    - List action handlers
   /LISTACT     - List actions
   /DELACT      - Delete actions
   /ADDPACT     - Add action matching on profiles
   /ADDACT      - Add action matching on attributes
   /EDITPACT    - Edit actions matching on profiles
   /EDITACT     - Edit actions matching on attributes
   /LISTPROF    - List profiles
   /DELPROFILE  - Delete profiles
   /ADDPROFILE  - Add profile based on attributes
   /ADDCPROF    - Add profile based on other profiles
   /EDITPROFILE - Edit profile based on attributes
   /EDITCPROF   - Edit profile based on other profiles
   -------------------------------------------------------
   Select logs using /ALL or /TAG
   Select actions using /ALL, /TAG, or /HASHAND
   Select profiles using /ALL, /TAG, or /NAME
   Select handlers using /ALL or /NAME
```

### 2.1.4.1 Examples for the Alert Manager Command Line Interface

One of the most powerful parts of NetFinity 5.0 is the Alert Manager. It is a service that is able to start different actions based on the type of alert that has been received. The Alert Manager is also highly customizable. You can create new alert profiles and define how NetFinity 5.0 will react when these alerts occur.

In this example we defined a new alert profile and the action we wanted taken when an incoming alert fits into it. First, we show what profiles already exist.

```
[C:\netfin]nfalrtcl /listprof /all
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ TAG=0x0001919B, NAME="All Alerts",
  TYPES=ANY,    APPS=ANY,
  ATYPES=ANY,
  SEVS=ANY,
  SENDERS=ANY
}
{ TAG=0x00018121, NAME="Power-On Error Detect Error Alerts",
  TYPES={ UNKFLT },
  APPS={ "POED    " },
  ATYPES={ 0x0201 },
  SEVS={ 4 },
  SENDERS=ANY
}
:
```

After issuing that command you will see that there are many predefined alert profiles in NetFinity 5.0. Looking at the output we can get some useful information. First we see the tag for this profile. As with groups and systems, all profiles have unique tags. We can see the profile name and the profile type. This type is always six letters long and is made of two, three-letter characters:

| Table 1. Alert Types and Classes | | | |
|---|---|---|---|
| **Alert Types** | | **Alert Classes** | |
| UNK | Unknown | UNK | Unknown |
| SYS | System | FLT | Fault or Failure |
| DSK | Disk or DASD | ERR | Error |
| NET | Network | WRN | Warning |
| OS_ | Operating System | INF | Information |
| APP | Application | | |
| DEV | Device | | |
| SEV | Severity | | |

You can also see the applications where the alert comes from, the application alert type (ATYPE), the severities and the machine where the alert was generated.

After reviewing the default profiles that come with NetFinity 5.0 you can then define some of your own. There are two possibilities. You can either create a profile based on the predefined profiles or you can build a new profile definition

based on alerts. In order to show the new command line functions for the
NetFinity 5.0 Alert Manager, we show both. Let's start with a new profile, which
combines all possible errors.

```
[C:\netfin]nfalrtcl /addprofile:"ERRORS" /addtype:UNKERR /addtype:SYSERR
/addtype:DSKERR /addtype:NETWRN /addtype:OS_ERR /addtype:DEVERR /addtype:SECERR
/addapp:Any /addatype:Any /ADDSEV:Any /ADDSENDER:Any
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0A2DD8AE
```

Let's check to see if the new profile is in the GUI.
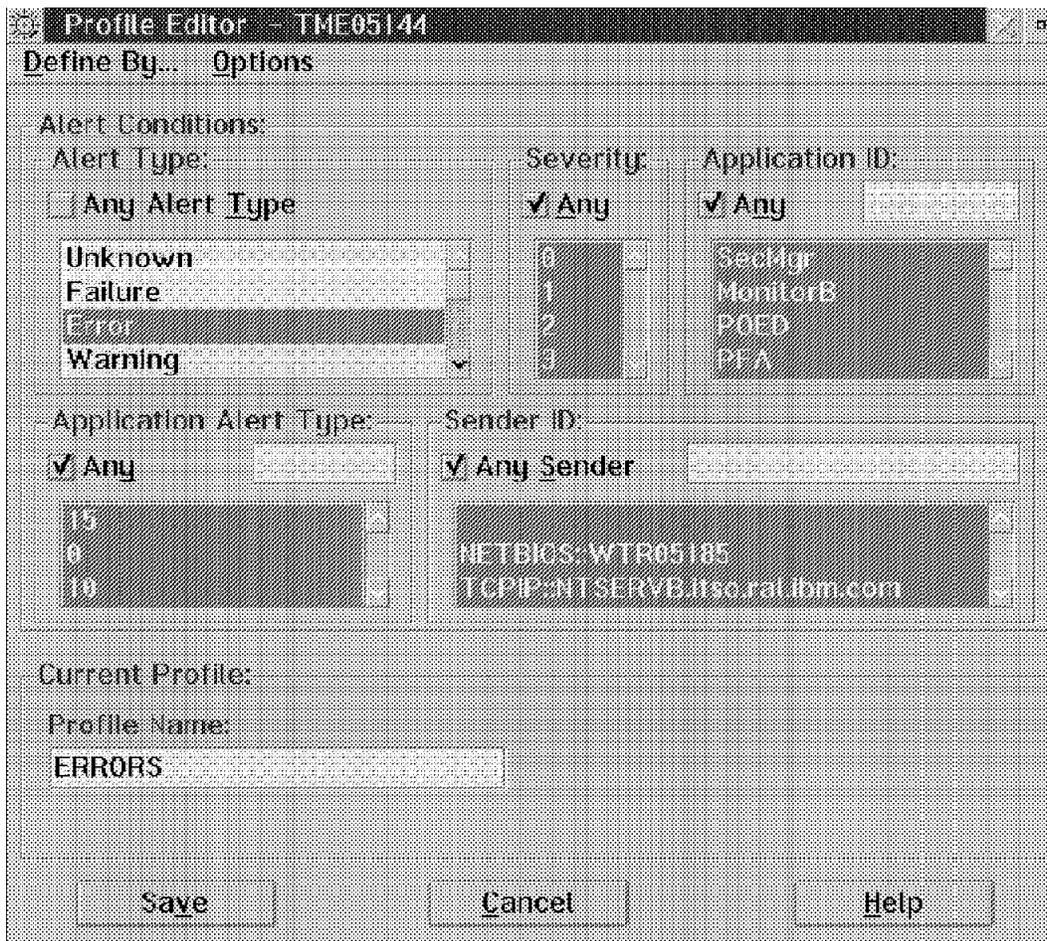


Figure 18. Alert Manager - New Profile

In looking at Figure 18, it seems like we made a mistake with the network error.
We forgot the application error. It is easy to edit the profile.

```
[C:\netfin]nfalrtcl /editprof /tag:0A2DD8AE /addtype:APPERR /addtype:NETERR
/addtype:NETWRN /delsev:Any /addsev:0 /addsev:1 /addsev:2
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0A2DD8AE
```

We also changed the severities parameter to values 0, 1, and 2 instead of Any.
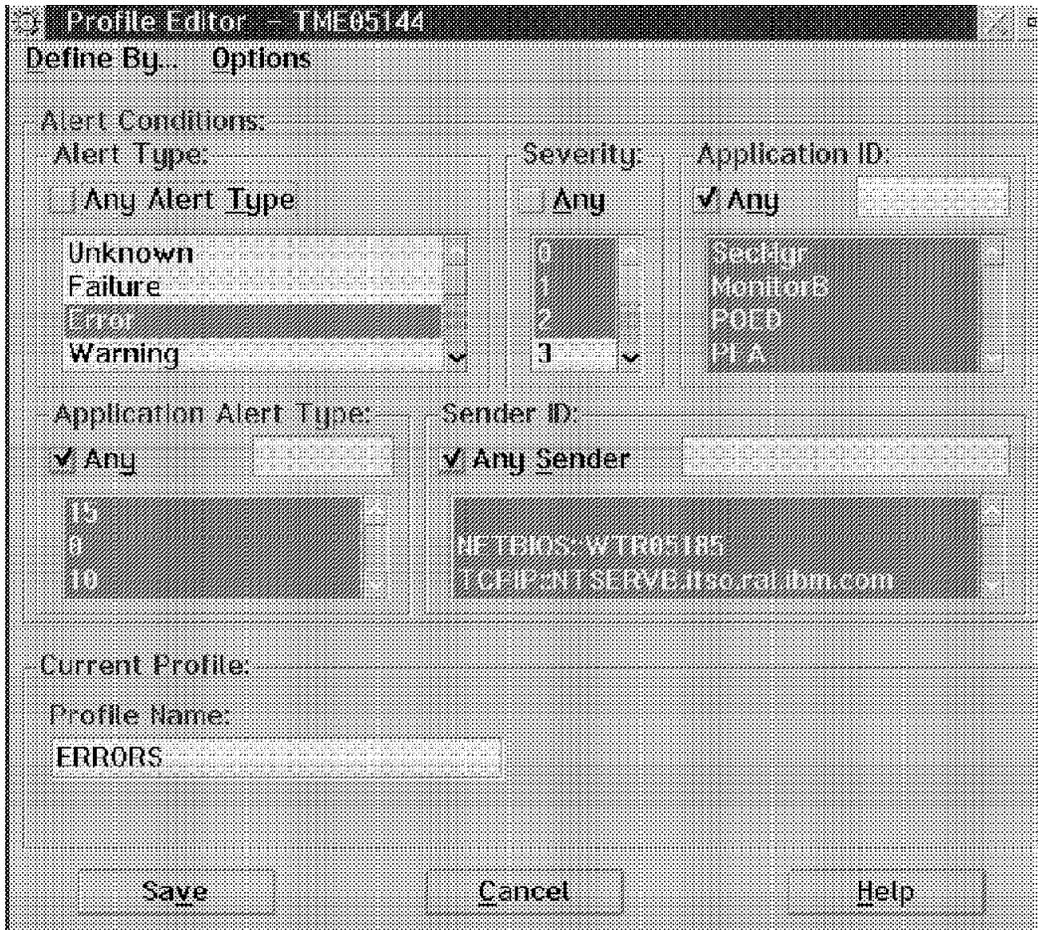
*Figure 19. Alert Manager - New Profile Edited*

Following the same process as above, we created a profile called WARNINGS. Now we can create another profile from these two profiles:

```
[C:\netfin]nfalrtcl /addcprof:"ERR_WRN" /addprof:"WARNINGS" /addproftag:0A2DD8AE
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0A63500E
```

In the GUI this looks like:
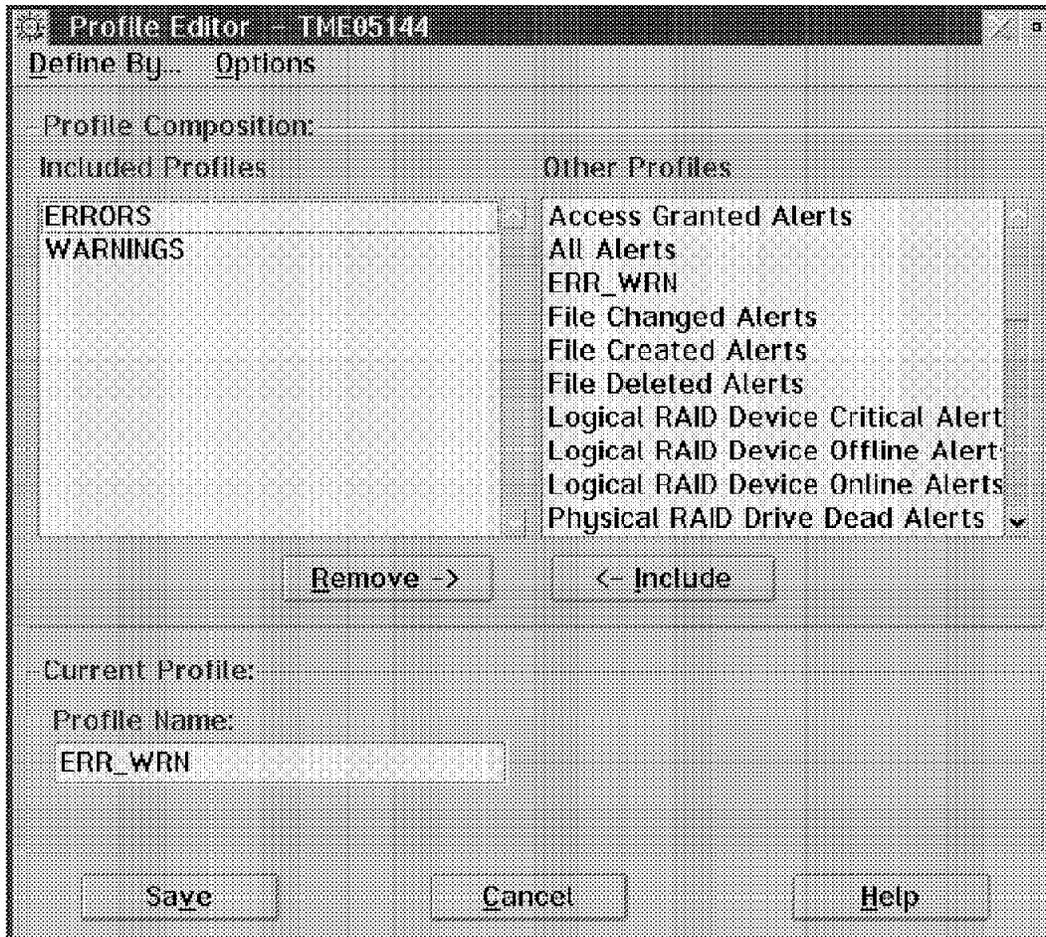
*Figure 20. Alert Manager - New Composite Profile*

We also want to get the alerts with alert class unknown, so we add another profile and edit the composite profile.

```
[C:\netfin]nfalrtcl /editcprof /name:"ERR_WRN" /newname:"ERR_WRN_UNK"
/addprof:"UNKNOWNS"
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0A63500E
```

Once the profiles are set up, we can take some actions. Let's first check which actions are predefined.

```
[C:\netfin]nfalrtcl /listact /all
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ TAG=0x0959C992, HANDLER="ALERTPOPUP",
  TYPES=ANY,
  APPS=ANY,
  ATYPES=ANY,
  SEVS={ 0, 1, 2, 3 },
  SENDERS=ANY
}
{ TAG=0x00017FOD, HANDLER="ALERTLOG",
  TYPES=ANY,
  APPS=ANY,
  ATYPES=ANY,
  SEVS=ANY,
  SENDERS=ANY
}
```

From the system installation, there were two predefined alert actions. The first
pops up the NetFinity 5.0 Alert window for any type of error, from any
application, with any application ID, from any sender and for severities 0, 1, 2
and 3. The second logs any alerts in the NetFinity 5.0 Alert log.

To define a new alert action it is necessary to know which handler IDs we have.
To get a list, type:

C:\netfin]nfalrtcl /listhand /all

```
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ HANDLER="WEBACT/O", NAME="Send alert as TCP/IP Web mail to '%s' at '%s'
  PARMLAB0="Target user ID",
  PARMLAB1="Target host address"
}
{ HANDLER="RUNMINCOMMAND", NAME="Execute minimized command '%s'",
  PARMLAB0="Command Line:"
}
⋮
{ HANDLER="FFSTALT/O", NAME="Forward Alert to FFST/2"
}
⋮
{ HANDLER="PAGERACT/O", NAME="Activate a numeric pager using a modem",
  PARMLAB0="Modems Com port (EX:COM1)",
  PARMLAB1="Pager number",
  PARMLAB2="Digital pager display"
}
{ HANDLER="PAGERACT/1", NAME="Send alert to alphanumeric pager through TA
  PARMLAB0="Modems Com port (EX:COM1)",
  PARMLAB1="TAP access number",
  PARMLAB2="Pager ID",
  PARMLAB3="Additional text to send"
}
⋮
```

You can see there are quite a lot of actions. They can have 0-4 parameters,
which must be defined. Similar to the profiles, actions can be defined according
to profiles or directly for special alert conditions.

```
[C:\netfin]nfalrtcl /addpact:"ALERTFORWARD" /newname:"There is something wrong"
/PARM0:"NETBIOS" /PARM1:"NETBIOS::WTR05185" /ADDPROF:"ERR_WRN_UNK"
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0AC6BADD
```

The initial name that we chose was not very clear, but we can edit it to make it
more descriptive.

```
[C:\netfin]nfalrtcl /editpact /hashand:"ALERTFORWARD"
/hasprof:"ERR_WRN_UNK" /newname:"ERROR/WARNING or UNKNOWN condition"
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0AC6BADD
```



*Figure 21. Alert Definition Based on Profiles*

Now let's define an alert action for certain conditions.

```
[C:\netfin]nfalrtcl /addact:"SETERRORCONDITION" /parm0:"Application Error Occured"
/addtype:APPERR /addapp:Any /addatype:Any /addsev:Any /addsender:Any
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0ADE762E
```

Errors normally have severity 0, 1, or 2, so we can continue to use these.

```
[C:\netfin]nfalrtcl /editact /hashand:"SETERRORCONDITION" /hastype:APPERR
/delsev:Any /addsev:0 /addsev:1 /addsev:2
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0ADE762E
```

*Figure 22. Alert Definition Based on Error Conditions*

After these definitions the Alert Actions window looksfile like this.

*Figure 23. Alert Action Window*

The final example shows how to delete all of the profiles that we just defined.

```
[C:\netfin]nfalrtcl /delact /hashand:"SETERRORCONDITION"
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0ADE762E

[C:\netfin]nfalrtcl /delact /hashand:"ALERTFORWARD" /hasprof:"ERR_WRN_UNK"
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0AC6BADD

[C:\netfin]nfalrtcl /delprofile /name:"UNKNOWNS"
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0AB28CD9

[C:\netfin]nfalrtcl /delprofile /name:"WARNINGS"
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0A6056D3

[C:\netfin]nfalrtcl /delprofile /tag:0a2dd8ae
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0A2DD8AE

[C:\netfin]nfalrtcl /delprofile /tag:0a63500e
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0A63500E
```

The last option left for the NetFinity 5.0 Alert Manager Command Line Interface is the /LISTLOG option:

```
[C:\netfin]nfalrtcl /listlog /all
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ TAG=0x0AE8577A, TEXT="The following file has changed: "C:\TEMP\MARKUS.TXT"
  TIME=18:52:13, DATE=02/10/1996, SEV=1, TYPE=APPWRN,
  APP="MonCritF", ATYPE=0x0000,
  SENDER=""
}
```

You get all the information about the alert(s) in the log.  In this case a file named C:\TEMP\MARKUS.TXT was monitored for changes and someone edited it.

## 2.1.5  System Monitor - NFSMONCL.EXE

The parameters that are associated with the nfsmoncl command are listed below:

```
The following command-line operations are supported:
   /GETMON      - List monitors and attribute groups
   /SETMONREC   - Enable/disable data recording
   /GETMONTHR   - List monitor thresholds
   /DELMONTHR   - Delete monitor thresholds
   /ADDMONTHR   - Add new monitor threshold
   /EDITMONTHR  - Edit monitor thresholds
   ------------------------------------------------------
   Select monitors using /ALL, /MONID, or /MONNAME
   Select thresholds using /ALL, /MONID, or /MONNAME
     (with optional /THRNAME)
```

### 2.1.5.1 Examples for the System Monitor Command Line Interface

Without the new NetFinity 5.0 command line interface it was very difficult to monitor system resources on a large number of remote machines. This section shows the basics for the monitoring process. In Chapter 4, "Command Line Automation" on page 127 the setting up of monitors on many remote machines is covered.

```
[C:\netfin]nfsmoncl /getmon /all
NetFinity System Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ MONITOR_ID=0xAB348194, NAME="CPU Utilization",
  SAMPLE=5000, UNITS_LBL="Percent", RECORDING=DISABLED,
  MINVAL=0.000000, MAXVAL=100.000000, VALUE=1.332972
}
{ MONITOR_ID=0xAB348195, NAME="Process Count",
  SAMPLE=15000, UNITS_LBL="processes", RECORDING=DISABLED,
  MINVAL=0.000000, MAXVAL=256.000000, VALUE=42.000000
}
{ MONITOR_ID=0x6327AEE2, NAME="Drive C: Space Used",
  SAMPLE=300000, UNITS_LBL="Megabytes Used", RECORDING=DISABLED,
  MINVAL=0.000000, MAXVAL=266.000000, VALUE=181.004395
}
{ MONITOR_ID=0x6327B0E2, NAME="Drive C: Space Remaining",
  SAMPLE=300000, UNITS_LBL="Megabytes Free", RECORDING=DISABLED,
  MINVAL=0.000000, MAXVAL=266.000000, VALUE=84.995605
}
  ⋮
```

As you can see, there are a lot of monitors in NetFinity 5.0 and the actual amount depends on what NetFinity 5.0 finds in your system environment. If, for example, you have installed TCP/IP, you will get a lot of monitors related to that protocol. If you don't have it installed, these monitors will not show up in your list. Another important point is that all of the monitors are constantly monitoring the system. You can see sample values in the field VALUE in the above screen. We can enable the recording of the values that we are capturing. An example of this follows:

```
[C:\netfin]nfsmoncl /setmonrec:enabled /monname:"CPU Utilization"
NetFinity System Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
MONITOR_ID=0xAB348194
```

If we now look at the CPU Utilization Monitor, we find that recording is enabled.

```
[C:\netfin]nfsmoncl /getmon /monid:ab348194
NetFinity System Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ MONITOR_ID=0xAB348194, NAME="CPU Utilization",
  SAMPLE=5000, UNITS_LBL="Percent", RECORDING=ENABLED,
  MINVAL=0.000000, MAXVAL=100.000000, VALUE=0.384234
}
```

---
**Important**

Enabling recording does not mean you can export this data to a database. It is only for NetFinity 5.0 internal use (for example, to create line graphs). If you want to analyze your system resources over a period of time, you have to set up the NetFinity 5.0 Scheduler to export data from one or more monitors on one or more machines to a database accessible from your managing system.

---

The more important part of the NetFinity 5.0 System Monitor command line interface deals with thresholds for monitors. The thresholds that are already defined for this system are shown below:

```
[C:\netfin]nfsmoncl /getmonthr /all
NetFinity System Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ MONITOR_ID=0x6327B0E2, THRESHNAME="Diskspace",
  DUR=5, RPT=0, HIERRVAL=NONE,
  HIWRNVAL=NONE,
  LOWRNVAL=NONE,
  LOERRVAL=125.000000, LOERRSEV=2, LOERRNTFY=ENABLED,
  RTNSEV=0, RTNNTFY=DISABLED, LOCALNTFY=DISABLED
}
```

After a new installation of NetFinity 5.0 no thresholds are set. In order to show something more than an empty list, we set up a threshold on the Drive C: Space Remaining Monitor, to send out an alert if the free disk space on drive C: is less than 125 MB for a duration of 5 seconds. To get the monitor name from the MONITOR_ID use the /GETMON parameter with the nfsmoncl command.

```
[C:\netfin]nfsmoncl /getmon /monid:6327B0E2
NetFinity System Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ MONITOR_ID=0x6327B0E2, NAME="Drive C: Space Remaining",
  SAMPLE=300000, UNITS_LBL="Megabytes Free", RECORDING=DISABLED,
  MINVAL=0.000000, MAXVAL=266.000000, VALUE=86.926758
}
```

As an example, we want a warning whenever the CPU Utilization is higher than 90% for more than a minute and an alert when it's higher than 99% for more than 15 seconds. Just to show what's possible, we set a threshold to generate a warning whenever the CPU utilization is less than 1% for more than 30 minutes. That could be because the station is no longer being used.

```
[C:\netfin]nfsmoncl /addmonthr:"CPU Utilization WARNING"
/monid:AB348194 /dur:60 /hiwrnval:90 /hiwrnntfy:enabled
NetFinity System Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
 { MONITOR_ID=0xAB348194, THRESHNAME="CPU Utilization WARNING" }

[C:\netfin]nfsmoncl /addmonthr:"CPU Utilization ERROR"
/monid:AB348194 /dur:15 /hierrval:99 /hierrntfy:enabled
NetFinity System Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
 { MONITOR_ID=0xAB348194, THRESHNAME="CPU Utilization ERROR" }

[C:\netfin]nfsmoncl /addmonthr:"Sleeper WARNING"
/monid:AB348194 /dur:1800 /lowrnval:1  /lowrnntfy:enabled
NetFinity System Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
 { MONITOR_ID=0xAB348194, THRESHNAME="Sleeper WARNING" }
```

Figure 24. Threshold Definition

We had to define three different thresholds because of the duration parameter. If all of the durations were the same we could have put all of the threshold values into one definition. The thresholds are set up and the alert action appropriate to the alert conditions should be run.

```
┌─────────────────────────────────────────────────────────────────┐
│ ⌐ Alert Received                                                 ▨│
│ Current Alert                                                     │
│ Alert Text:      Error Alert CPU Utilization ERROR: Monitor 'CPU │
│                  Utilization' has been above or equal to 99% for 15 │
│                  seconds.                                         │
│                                                                   │
│ Type of Alert:  Error                                            │
│ Severity:        2                        Time of Alert: 01:37:11p │
│ Application ID:  MonitorB                  Date of Alert: 10-03-1996 │
│ Application Alert Type:  0000                                     │
│ Received From:                                                   │
│ System Name:    TME05144                                         │
│                                                                   │
│ Alerts Received                                                  │
│    Time        Date                           Text               │
│ 01:37:11p 10-03-1996 Error Alert CPU Utilization ERROR: Monitor 'CPU Ut│
│ 01:37:56p 10-03-1996  Warning Alert CPU Utilization WARNING: Monitor 'C │
│                                                                   │
│                                                                   │
│                                                                   │
│                                                                   │
│                                                                  ▶│
└─────────────────────────────────────────────────────────────────┘
```

*Figure 25. Alert Window with Threshold Alerts*

In this case, we had the NetFinity 5.0 alert window come up with both the
warning and the error.

┌─ **Hint** ──────────────────────────────────────────────────────────┐
│                                                                      │
│ If you want to rebuild this scenario, in order to get the CPU Utilization up to │
│ 100% simply open a DOS window and run a program in it.  In our example a │
│ DIR /S was enough to get the error.                                  │
│                                                                      │
└──────────────────────────────────────────────────────────────────────┘

Deleting these thresholds is again very easy.

```
[C:\netfin]nfsmoncl /delmonthr /monname:"CPU Utilization ERROR"
NetFinity System Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
```

The same basic command would be true for the other monitors we added.

┌─ **Note** ──────────────────────────────────────────────────────────┐
│                                                                      │
│ If you have been a NetFinity user in previous releases, you were possibly │
│ aware that if you had another CPU monitor running in parallel to NetFinity's │
│ monitor, the utilization was not accurate.  Often it just showed 100% │
│ utilization.  The picture below shows, that beginning with OS/2 Warp Version │
│ 4 (Merlin) the NetFinity 5.0 CPU Utilization Monitor and the System Activity │
│ Monitor of OS/2 can coexist.                                         │
│                                                                      │
└──────────────────────────────────────────────────────────────────────┘
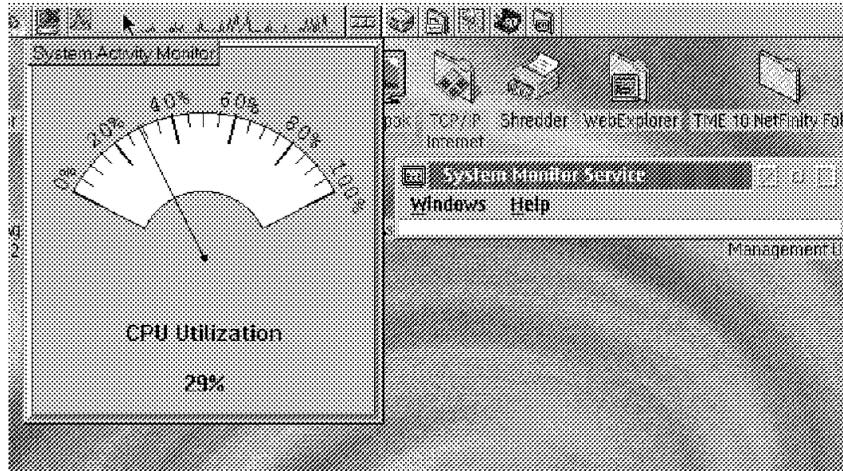
*Figure 26. Coexistence of NetFinity 5.0 CPU Utilization Monitor and OS/2 Warp Version 4 System Activity Monitor*

## 2.1.6 Critical File Monitor - NFCRTFCL.EXE

The parameters that are associated with the nfcrtfcl command are listed below:

```
The following command-line operations are supported:
  /?      - Returns this command-line help
  /LIST   - Returns information about the monitored files
  /DELSYS - Delete a system file from the list of monitored files
  /DELSTD - Delete a standard file from the list of monitored files
  /SETSYS - Add a new system file to be monitored or change the alert
            severity of a system file already being monitored
  /SETSTD - Add a new standard file to be monitored or change the alert
            severity of a standard file already being monitored
  /N:netpath - Optional network path to target system
  /S:sysname - Optional system name for target system
```

### 2.1.6.1 Critical File Monitor Command Line Interface Example

This example shows how to monitor important files. These can be system files or user files. The difference is that you don't need to specify the full path for system files, while you must for all others. Of course the system files are different for the different operating systems:

- OS/2 3.x: CONFIG.SYS, STARTUP.CMD, AUTOEXEC.BAT

- OS/2 4.x: CONFIG.SYS, STARTUP.CMD, AUTOEXEC.BAT

- Windows 3.x: WIN.INI, SYSTEM.INI

- Windows NT 3.51: WIN.INI, SYSTEM.INI

- Windows NT 4.0: WIN.INI, SYSTEM.INI

- NetWare 3.x: AUTOEXEC.NCF, STARTUP.NCF, VOL$LOG.ERR, SYS$LOG.ER

- NetWare 4.x: AUTOEXEC.NCF, STARTUP.NCF, VOL$LOG.ERR, SYS$LOG.ER

The following file is for an OS/2-related file:

```
[C:\netfin]nfcrtfcl /setsys:config.sys /sev:1
NetFinity Critical File Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.

[C:\netfin]nfcrtfcl /setstd:c:\temp\markus.txt /sev:1
NetFinity Critical File Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.

[C:\netfin]nfcrtfcl /list
NetFinity Critical File Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ FILE="C:\CONFIG.SYS", SEV=1, TYPE=SYSTEM_FILE }
{ FILE="C:\TEMP\MARKUS.TXT", SEV=1, TYPE=STANDARD_FILE}
```

The above commands set up a monitor on the system and user file such that, if
they are changed, a genalert with severity 1 will be sent to the alert manager
component. Figure 27 shows the how to take the same action using the GUI as
we just did with the command-line interface.

---

**Important**

It is possible to use the /n:systemname parameter. If you set up the Critical
File Monitor to check a remote file, the alert will not occur on the system
where the file is changed but on the system where the monitoring was
defined. If you want to set up the remote system to also get the alert, you
can set the NetFinity 5.0 Alert Editor to forward that alert.

---
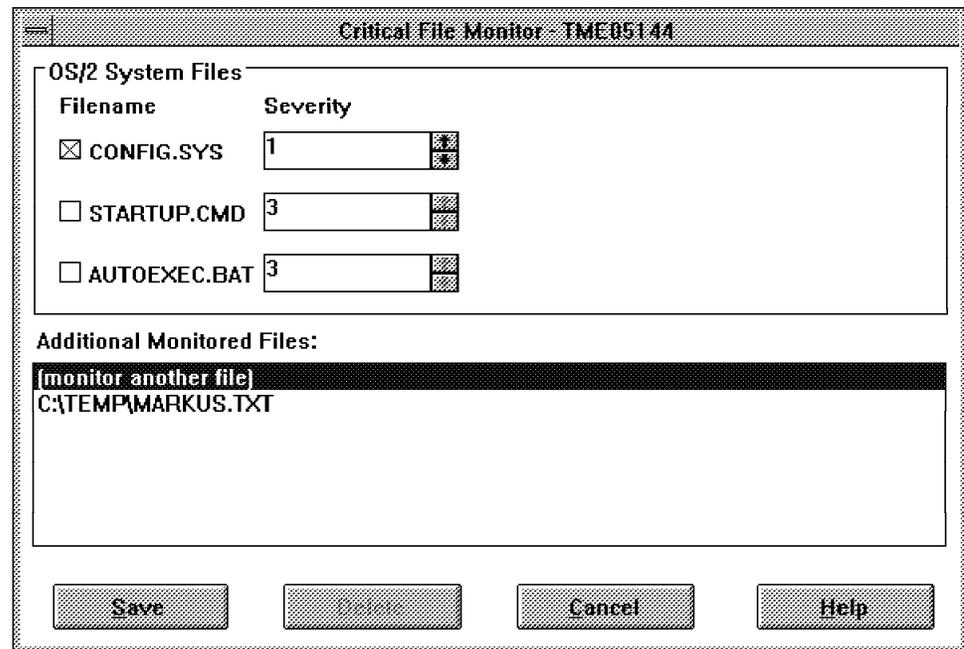


Figure 27. Critical File Monitor Enabled

```
[C:\netfin]nfcrtfcl /delsys:config.sys
NetFinity Critical File Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.

[C:\netfin:nfcrtfcl /delstd:c:\temp\markus.txt
NetFinity Critical File Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
```

## 2.1.7  Process Manager - NFPROCCL.EXE

The parameters that are associated with the nfproccl command are listed below:

```
The following command-line operations are supported:
  /?       - Returns this command-line help
  /GETPROC - Returns current process information
  /RUNCMD  - Execute a command-line program
  /LISTMON - List process monitor information
  /ADDMON  - Create a new process monitor record
  /DELMON  - Delete existing process monitor records
  /EDITMON - Modify existing process monitor records
----------------------------------------------------------
Select process monitor records using /ALL, /PROCNAME, or /TAG
```

### 2.1.7.1  Process Manager Command Line Interface Example

An important aspect of remotely managing systems is to check for certain
processes to be sure that they are up and running.  This example shows how to
set a monitor on a special process in order to generate an alert if it's down.  To
get a list of the processes that are running on your system, you need to use the
/GETPROC parameter.

```
[C:\netfin]nfproccl /getproc
NetFinity Process Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
⋮
{ PROCESS_ID=0x0008,
  EXENAME="C:\IBMCOM\LANMSGEX.EXE",
  THREADS=1,
  PARENT_ID=0x0000,
  SESSION_ID=0x0000
}
⋮
```

This list is similar to a PSTAT output on OS/2 or Windows NT.  The GUI shows
this information in a clearly arranged format.

| Program Name | Process ID | Parent Process ID | Session ID | Num. Threads |
|---|---|---|---|---|
| C:\EFCOMM\COMM.EXE | 008Ch | 0018h | 001Fh | 5 |
| C:\IBMCOM\LANMSGEX.EXE | 0008h | 0000h | 0000h | 1 |
| C:\IBMCOM\PROTOCOL\LANDLL.EXE | 0017h | 0000h | 0000h | 1 |
| C:\IBMLAN\NETPROG\LSDAEMON.EXE | 0014h | 0000h | 0000h | 1 |
| C:\IBMLAN\NETPROG\NETMSG.EXE | 0027h | 0018h | 0013h | 4 |
| C:\IBMLAN\NETPROG\VNRMINIT.EXE | 0015h | 0000h | 0000h | 1 |
| C:\IBMLAN\SERVICES\MSRV.EXE | 0039h | 0000h | 0011h | 2 |
| C:\IBMLAN\SERVICES\PEER.EXE | 003Bh | 0000h | 0011h | 6 |
| C:\IBMLAN\SERVICES\PEER.EXE | 003Ch | 0000h | 0011h | 3 |
| C:\IBMLAN\SERVICES\WKSTA.EXE | 0024h | 0000h | 0011h | 4 |
| C:\IBMLAN\SERVICES\WKSTAHLP.EXE | 0025h | 0000h | 0011h | 4 |
| C:\MPTN\BIN\CNTRL.EXE | 000Ah | 0000h | 0000h | 8 |

*Figure 28. Process List*

For our example, we want to get a warning if the LAN Messenger Services are not up and running. To accomplish this we set a monitor on that process:

```
[C:\netfin]nfproccl /addmon:"C:\IBMCOM\LANMSGEX.EXE"
/sev:4 /onnorun:5 /onstop:enabled /addnotify:local
NetFinity Process Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x138B13F6

[C:\netfin]nfproccl /listmon /all
NetFinity Process Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ TAG=0x138B13F6, PROCNAME="C:\IBMCOM\LANMSGEX.EXE", SEV=4,
  ONSTART=DISABLED, ONSTOP=ENABLED, ONNORUN=5
  NOTIFY={ LOCAL }
}
```

The graphical equivalent of the above would be:
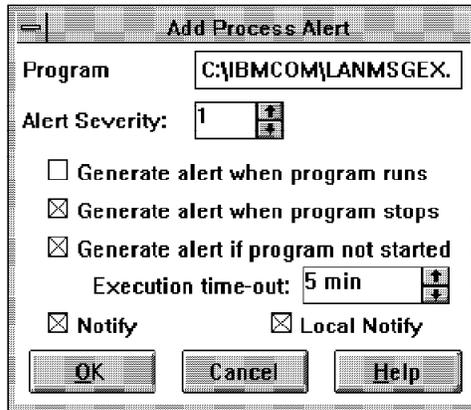
*Figure 29. Process Monitor Enabled*

We also want to be notified when the process either starts or restarts. To set that up, we need to edit the monitor. This can be done through the command line interface.

```
[C:\netfin]nfproccl /editmon /procname:"C:\ibmcom\lanmsgex.exe"
/onstart:enabled  /sev:0
NetFinity Process Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x138B13F6

[C:\netfin]nfproccl /listmon /tag:138B13F6
NetFinity Process Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ TAG=0x138B13F6, PROCNAME="C:\IBMCOM\LANMSGEX.EXE", SEV=0,
  ONSTART=ENABLED, ONSTOP=ENABLED, ONNORUN=5
  NOTIFY={ LOCAL }
}
```

If you no longer need the monitor, you can delete it.

```
[C:\netfin]nfproccl /delmon /tag:138B13F6
NetFinity Process Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x138B13F6
```

Finally there is one more thing you can do with the NetFinity 5.0 Process Manager command line interface. You can execute any command.

```
[C:\netfin]nfproccl /runcmd:"netscape"
NetFinity Process Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
```

This would start Netscape on your local system. You will not get any output on your screen from the command. For example, if you run a DIR for example, no directory listing will appear on the screen. This command gets more powerful when you use it in combination with the /N:systemname parameter to execute a command on a remote machine.

## 2.1.8  System Information - NFSYSICL.EXE

The parameters that are associated with the nfsysicl command are listed below:

```
The following command-line operations are supported:
  /?              - Returns this command-line help
  /HIST:fname     - Generate history file 'fname'
  /HISTOUT        - Generate history file with unique name
  /RPT:fname      - Generates report file 'fname'
  /RPTOUT         - Generates report file with unique name
  /EXPORT:dbi_name /DBNAME:database - Export data to select database
```

### 2.1.8.1  System Information Command Line Interface Example

The System Information tool was one of the few NetFinity services that had a command line interface in earlier versions.  You could run the SINFG30.EXE with certain parameters to either create or import a history or report file.  With NetFinity 5.0 the creation of a history or report file and the export to a database is made by NFSYSICL.  The difference between the HIST/RPT and HISTOUT/RPTOUT parameters is that when using HIST or RPT you can specify a filename.  When you use HISTOUT or RPTOUT a filename is automatically created for you.

```
[C:\netfin]nfsysicl /hist:sysinfo.hst
NetFinity System Information Command-Line Tool
Copyright (C) IBM Corp. 1996.
HSTFILE="sysinfo.hst"

[C:\netfin]nfsysicl /histout
NetFinity System Information Command-Line Tool
Copyright (C) IBM Corp. 1996.
HSTFILE="14020609.HST"
```

You can also export the database into a file.

```
[C:netfin\]nfsysicl /export:db2os2fi /dbname:"nfdb.txt"
NetFinity System Information Command-Line Tool
Copyright (C) IBM Corp. 1996.
```

# Chapter 3.  Webability

This chapter shows examples of launching the Webability feature on different platforms as alerts flow into NetFinity NT.

## 3.1  Enabling and Customizing the Web Interface

In order to have a NetFinity 5.0 client managed from a Web browser, you first have to enable the function and optionally do some minor customization that can limit who can use the Web interface.  This is done in the Web Manager, which can be accessed from the NetFinity 5.0 desktop.
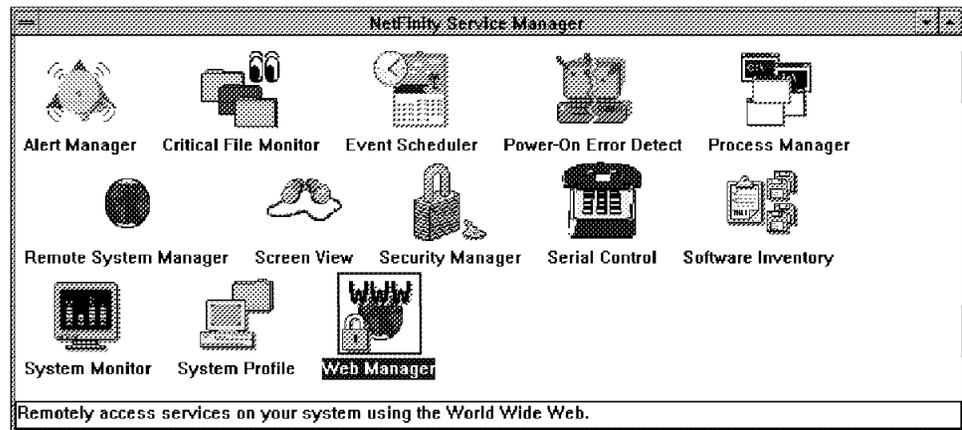
*Figure 30.  Selecting the Web Manager from the NetFinity 5.0 Main Window*

The Web Manager Configuration window consists of three sections:
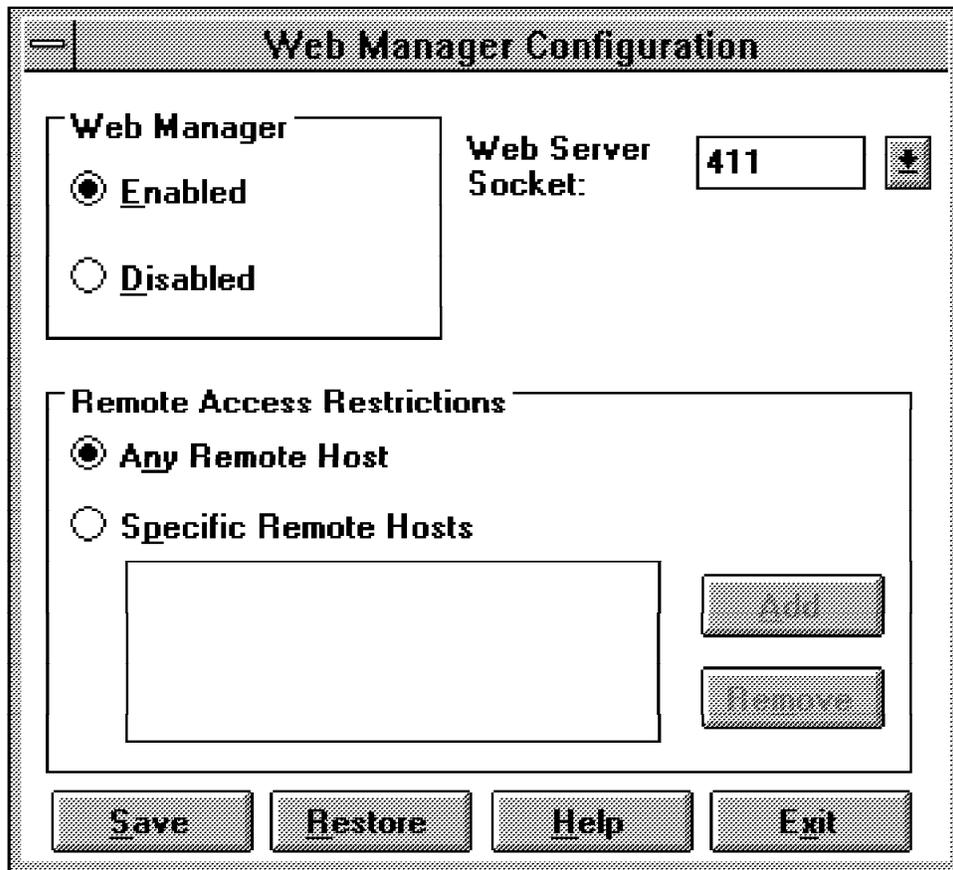
*Figure 31. Web Manager Configuration*

1. Web Manager: This is where the Web functions of NetFinity 5.0 can be enabled or disabled.

2. Web Server Socket: Specify the TCP/IP port that will be used by this client to communicate with the managing system. It is also needed if you have multiple Web servers on the same system.

3. Remote Access Restrictions: This lets you restrict access to this client by specific address, or you can let anyone have access to the port.
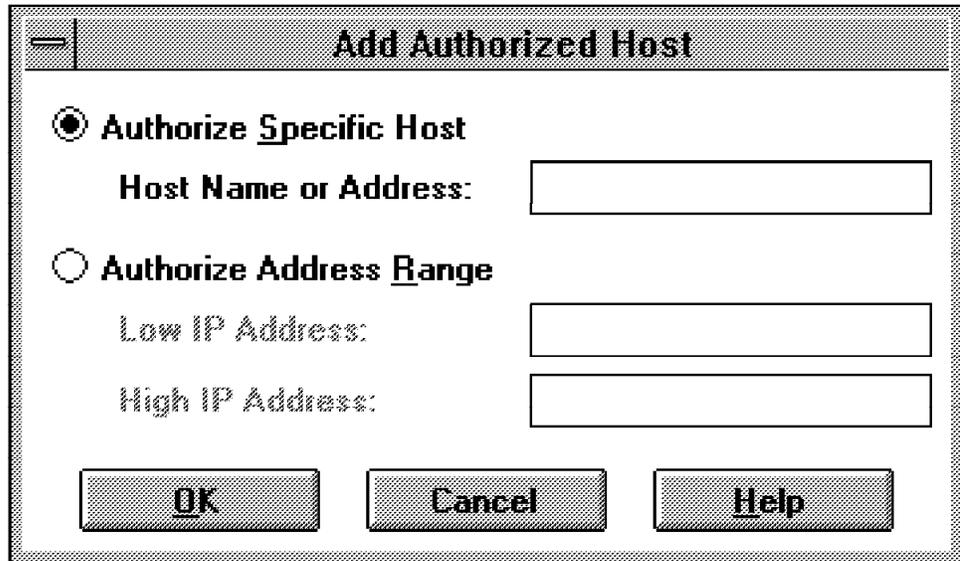
*Figure 32. Add Authorized Host*

You can either specify a single remote host with its hostname or IP address or you can give access to a range of IP addresses as shown in Figure 33.
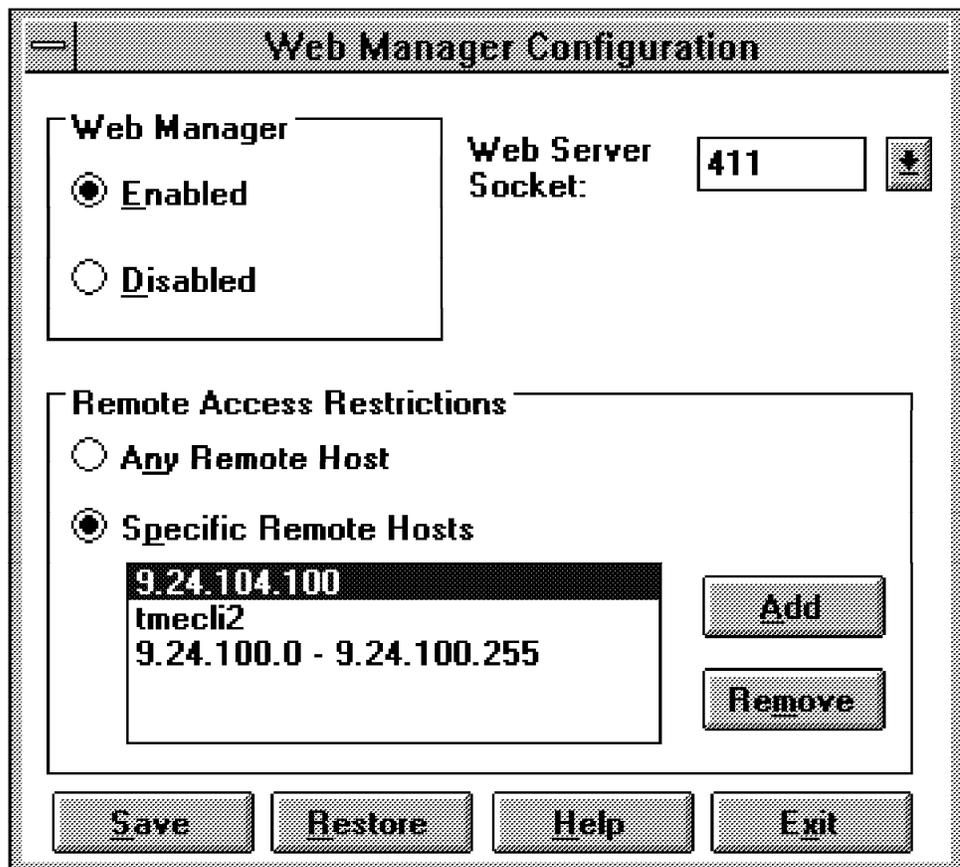


*Figure 33. Different Types of Remote Host Specifications*

After this customization is done, you can now access this client from any host in the list using a Web browser that supports Java (Figure 35 on page 47). We

used the IBM Web Explorer V1.1H and Netscape V3.0 in the project. We also tried some of the functions with the Microsoft Internet Explorer V3.01. The HTTP address of a client is represented by http://<IP-Address or hostname>:<IP-Port>/main.

## 3.2 How NetFinity 5.0 Functions Work from a Browser

Although most of the Web versions of the NetFinity 5.0 functions look very similar to the GUI, there are some differences. Therefore, this section discusses not only the NetFinity 5.0 functions, but the different ways those functions are implemented using the Webability feature.

### 3.2.1 Main Page

The main Web page shows all the available NetFinity 5.0 functions as icons. They represent hyperlinks to the server's pages. There is one function that only shows up in Webability: the Web Guru. For a description of its functions, see 3.2.6, "Web Guru" on page 74.



*Figure 34. Lock Symbol for the Default User ID <PUBLIC>*

> **Note**
>
> Assuming you remembered to change the default user ID <PUBLIC> to something else, you can click on the lock symbol that will appear under the icons. This will allow you to log on to NetFinity 5.0 with your correct user ID and password. Once you are logged on, the log symbol will disappear.
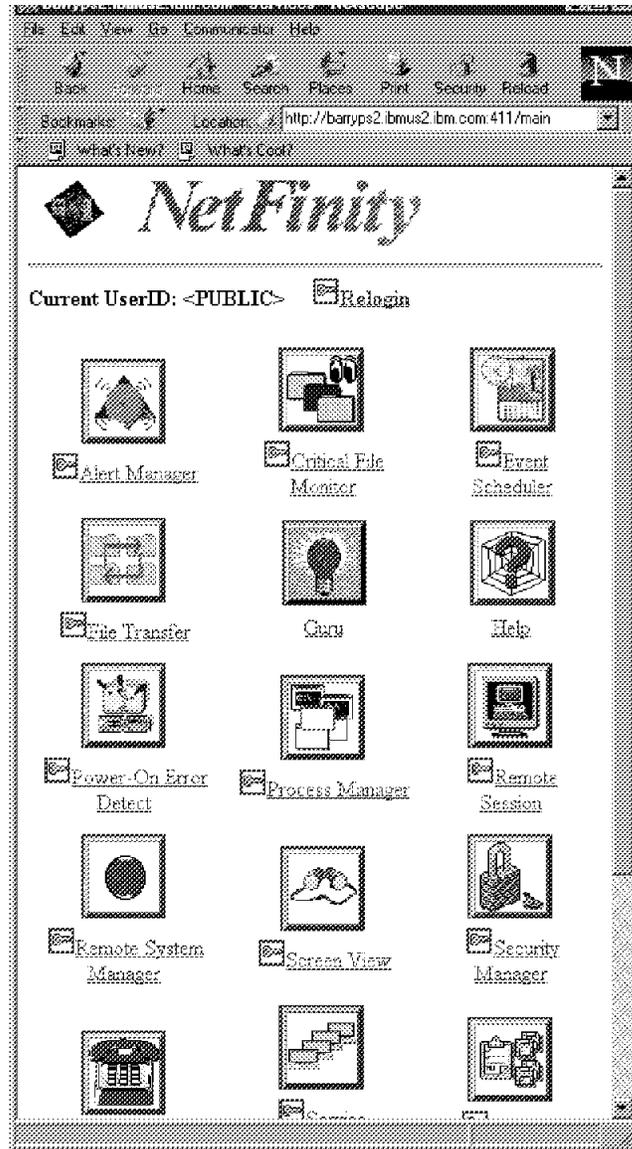
*Figure 35. Main Screen of Webability*

## 3.2.2 Alert Manager

The Alert Manager functions are arranged slightly different than they are in the NetFinity 5.0 GUI. First of all, the log is not part of the main screen. To help with the navigation you find six icons that lead you deeper into the Alert Manager pages.
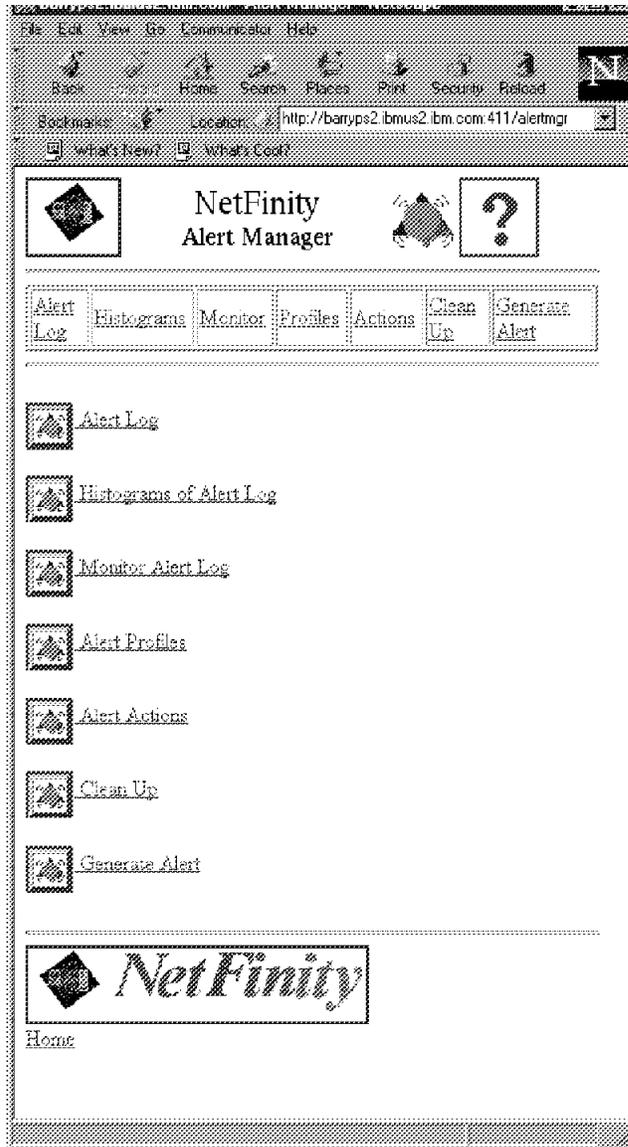
*Figure 36. Alert Manager*

The first icon in the list is for the Alert Log. You can select one or more alert profiles and also specify that you only wish to see alerts for a specific time period. For example, you can request to see the last 3 hours of alerts. After you have made a selection, you can click on the **Refresh Alert Log** button to see the alerts in which you are interested. The bottom part of the Web page lists any errors, as shown in Figure 37 on page 49. You can select one or more of them or click on the hyperlink to get more details. Just under the list of alerts there are some action buttons, which let you either return to the Alert Manager main page or delete some of the alerts.
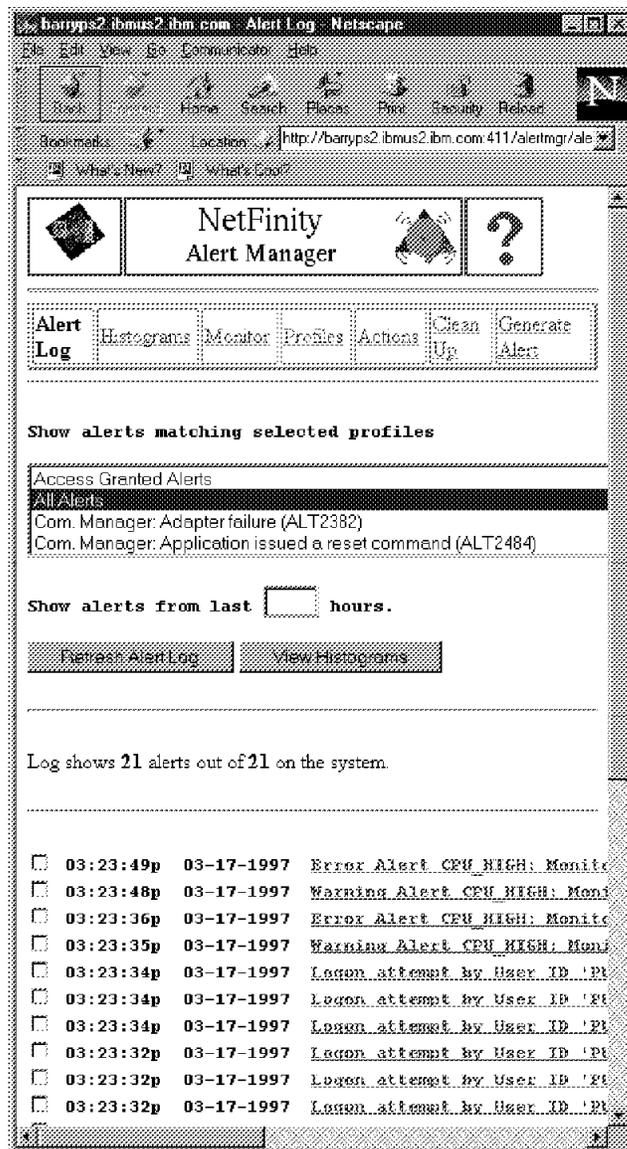
Figure 37. Alert Manager - Alert Log

There is also a button in the middle of the page to view histograms of the alerts. Just click on it to get results sorted by:

- Severity
- Alert class types
- Alert type ID
- Application ID
- Sender ID
- Alert Time History
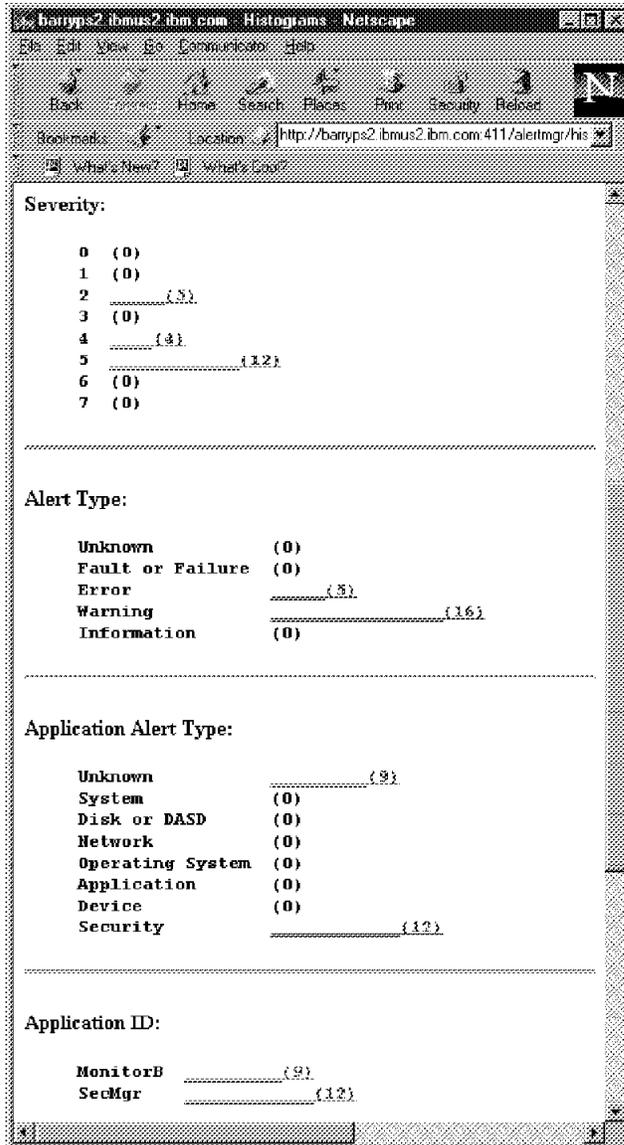
*Figure 38. Alert Manager - Alert Histograms Sample*

The Monitor Alert Log button is for the alert console, which lists every incoming alert since the time the page was started.

*Figure 39. Alert Manager - Monitor Alert Log*

On the Alert Profile page you can either edit or delete existing profile definitions. The NetFinity 5.0 profiles are predefined alert classes, characterized by alert type, severity, application ID, application alert type and sender ID, just as they are using the regular NetFinity 5.0 GUI.
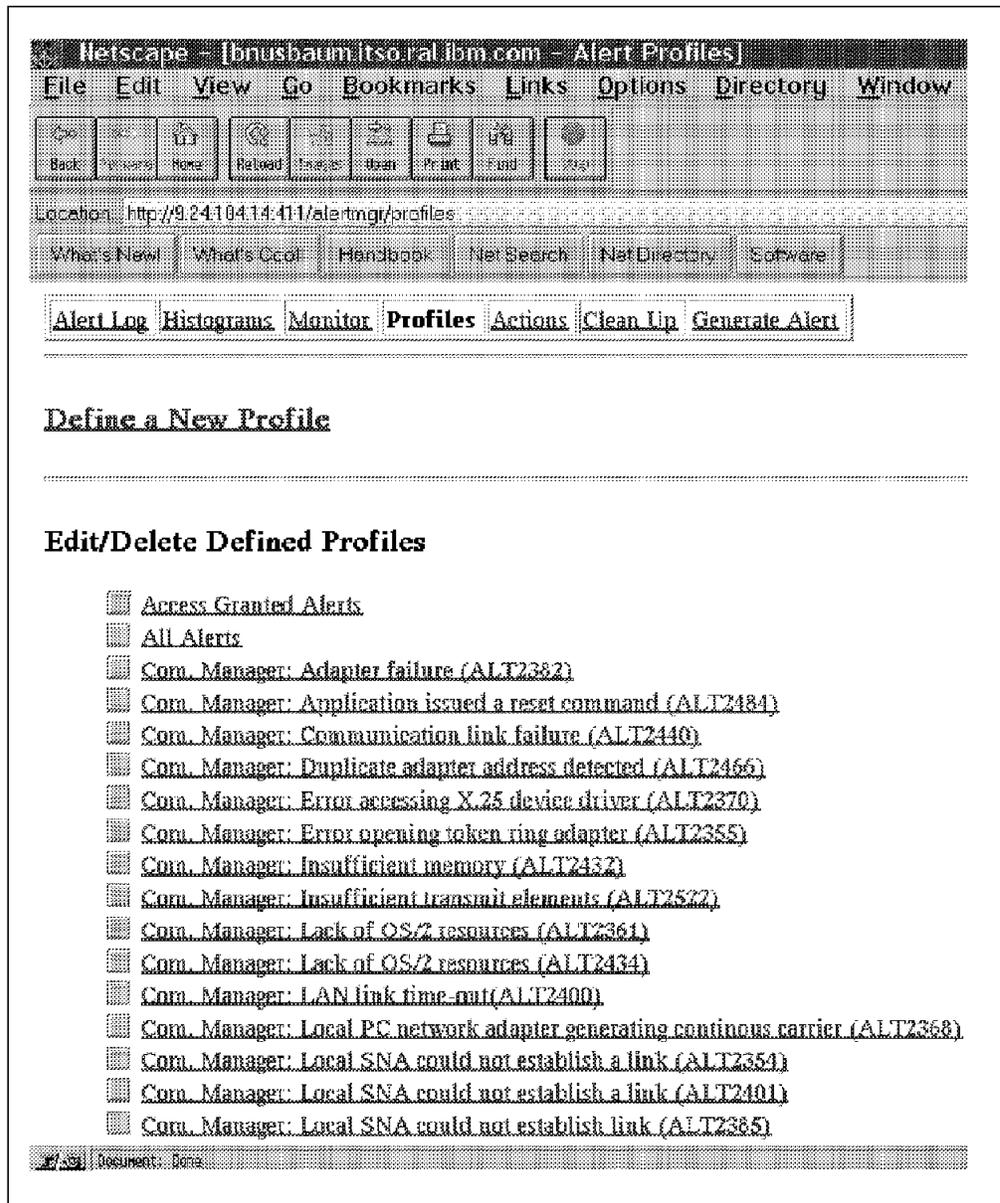
*Figure 40. Alert Manager - Alert Profiles*

If you go back to the Alert Manager main Web page, you can click on the keyword **Profiles**. This takes you to a list of alert actions available on the machine where the Webability server function is running. The list is based upon the operating system on which it is executing.

- Access Granted Alerts
- All Alerts
- File Changed Alerts
- File Created Alerts
- File Deleted Alerts
- Logical RAID Device Critical Alerts
- Logical RAID Device Off-line Alerts
- Logical RAID Device On-line Alerts
- Physical RAID Drive Dead Alerts
- Physical RAID Drive On-line Alerts
- Physical RAID Drive PFA Alerts

- Physical RAID Drive Standby Alerts
- Power-On Error Detect Error Alerts
- Power-On Error Detect Info Alerts
- Predictive Failure Analysis Alerts
- Process Failed to Start Alerts
- Process Started Alerts
- Process Terminated Alerts
- Public Access Granted Alerts
- Service Start Rejected Alerts
- Service Start Request Alerts
- System Access Denied Alerts
- System Off-line Alerts
- System On-line Alerts
- System Restart Initiated Alerts
- System Restart Rejected Alerts
- Threshold Error Alerts
- Threshold Return to Normal Alerts
- Threshold Warning Alerts
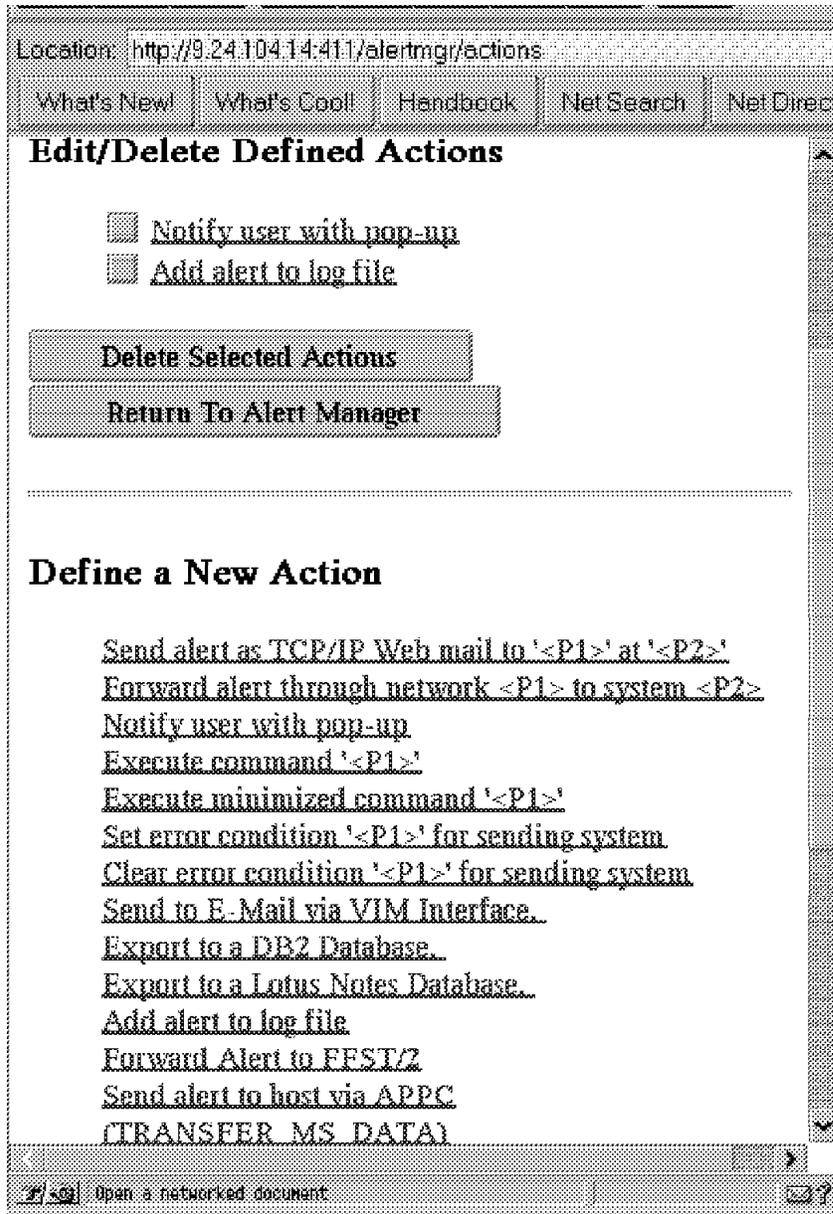
*Figure 41. Alert Manager - Alert Actions (NetFinity 5.0 for OS/2)*

If you click on an alert action, it takes you to the alert definition window, which shows you the same choices as if you were at their regular NetFinity 5.0 GUI interface.
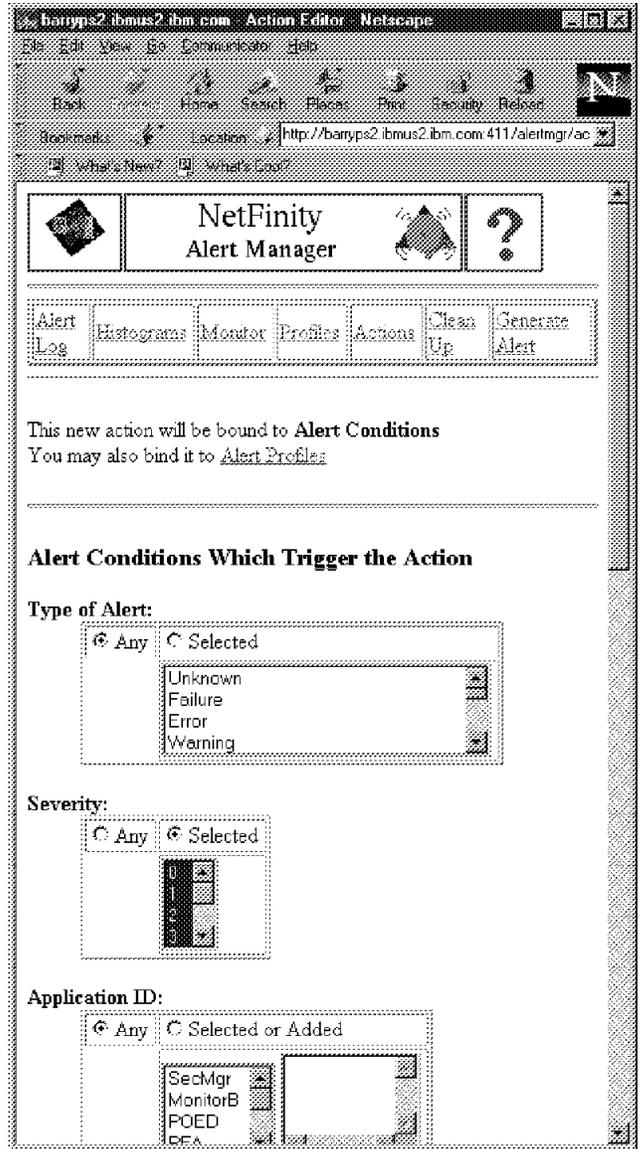
Figure 42. Alert Manager - Alert Actions

The bottom half of the Alert Actions window follows:
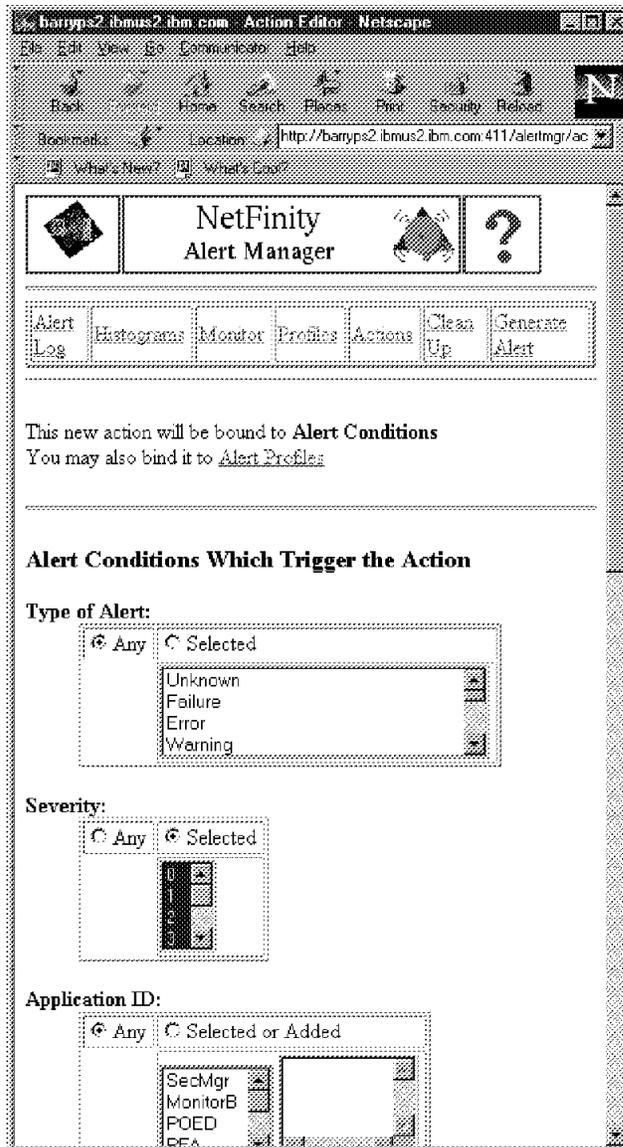
*Figure 43. Alert Manager - Alert Actions*

Another function that is not available in the regular NetFinity 5.0 interface, but is available using Webability, is the Generate Alert page. From here you can create an alert using the genalert function. When using a regular NetFinity 5.0 window, you can type in genalert to find out what all the parameters are, but you still need to retype the command after that. With the Webability function, you can fill in a form to tailor the genalert for your needs.
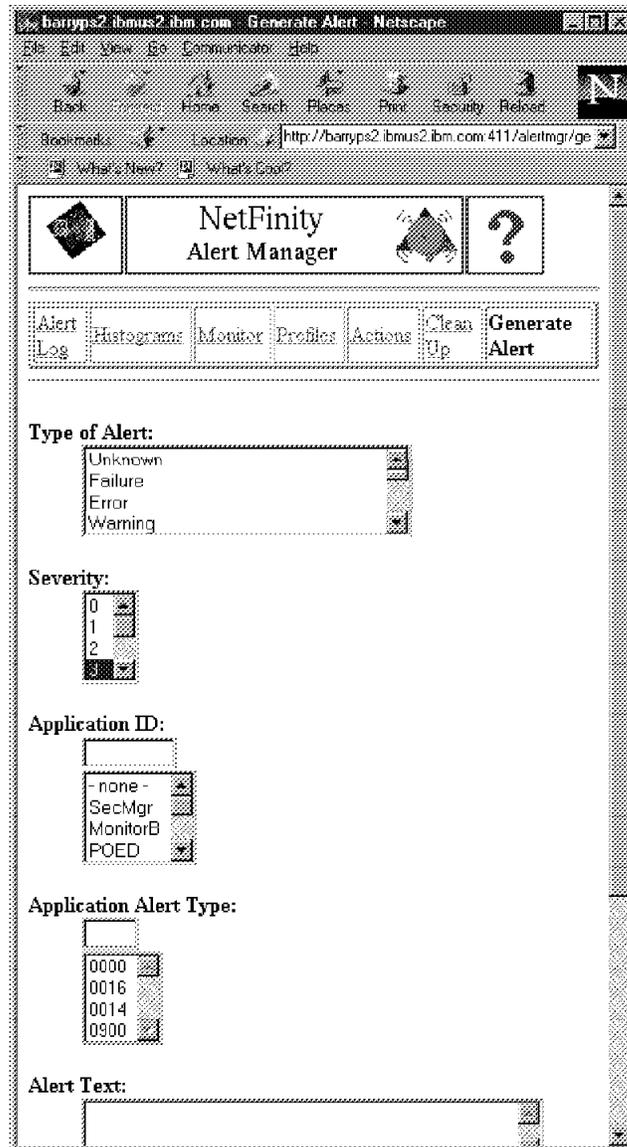
*Figure 44. Alert Manager - Generate an Alert*

### 3.2.3 Critical File Monitor

Just like the GUI, the Web page for the Critical File Monitor lets you select the most important system files to monitor by clicking on a check box. The files that are in the system files list are operating system-dependant. Therefore, a file name that you see on Windows NT may not appear in an OS/2 view.
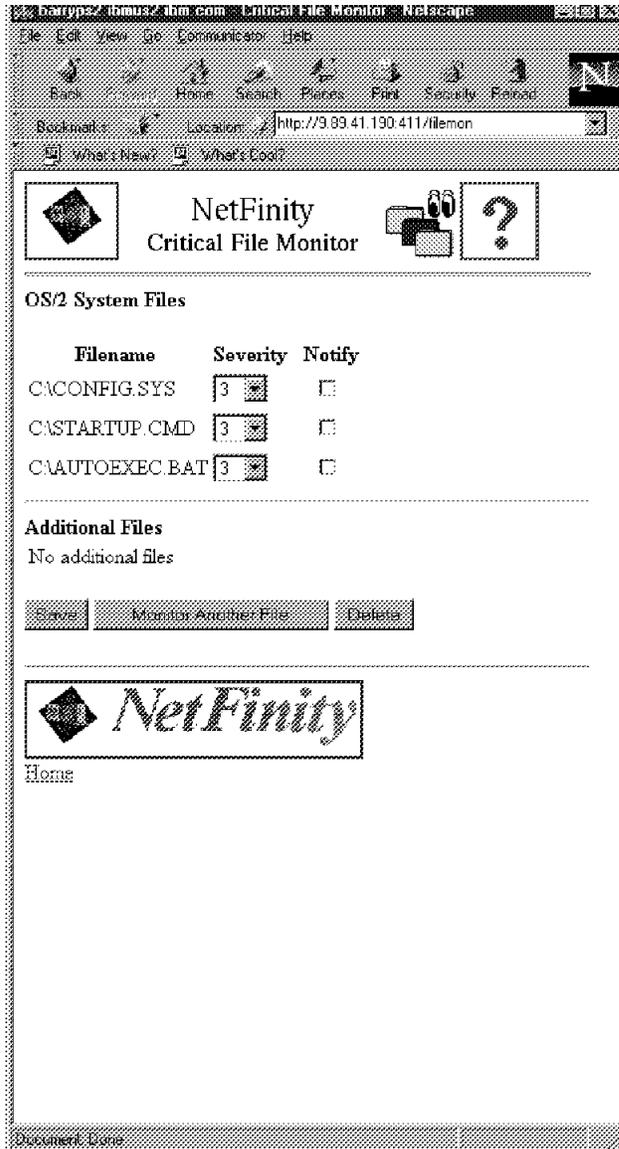
*Figure 45. Critical File Monitor - OS/2 System Files*
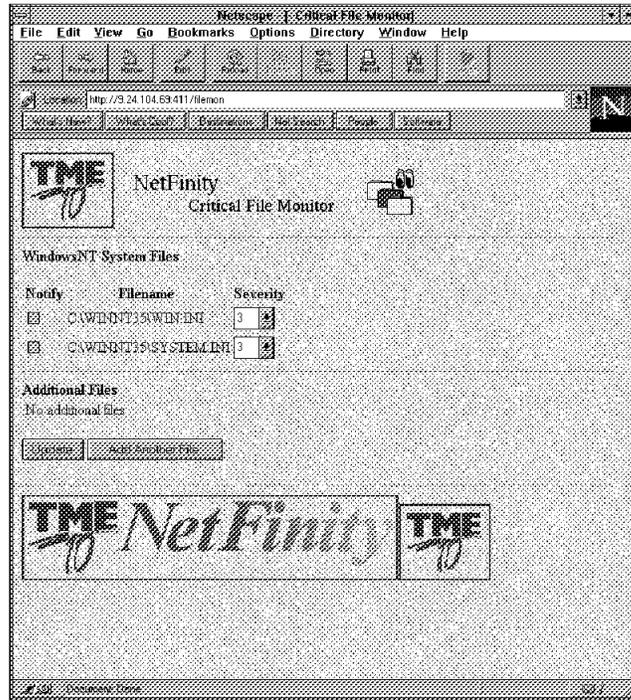
*Figure 46. Critical File Monitor - Windows NT 3.51 System Files*

Clicking on **Add Another File** moves you to a file selection page where you can browse through any of the hard drives and directories on the managed system.

---
**Note**

The file browser only shows you physical drives on which to monitor files. There is no way to monitor LAN-attached, NFS, or NetWare-attached drives.

---
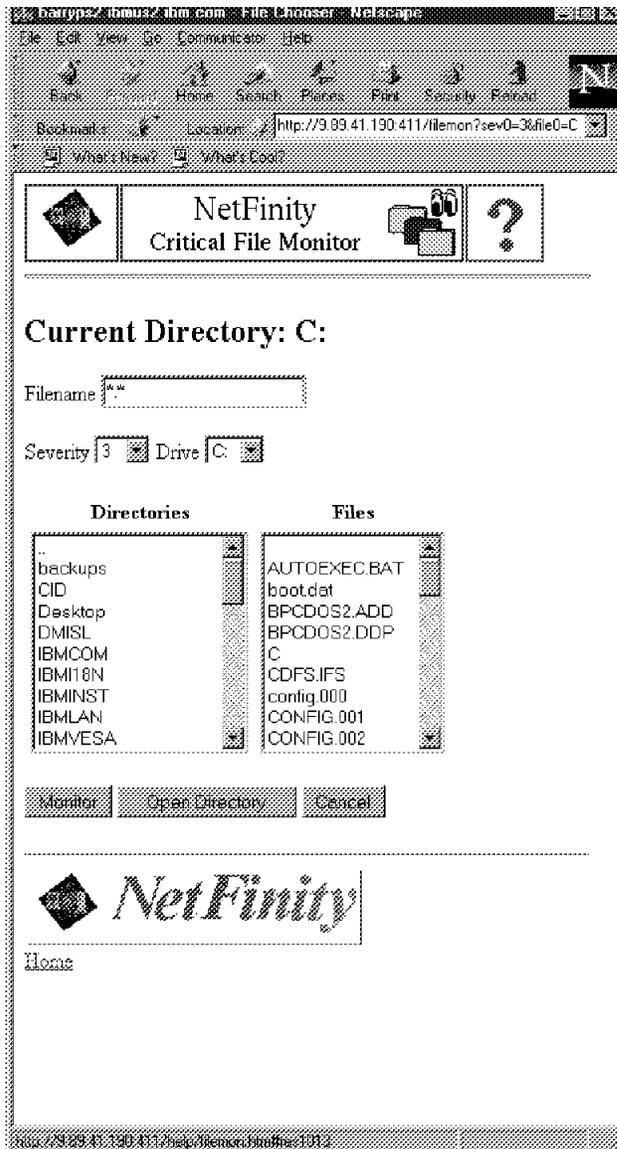
*Figure 47. Critical File Monitor - Add File Monitor*

Select the drive that has the file you are planning on monitoring and click on **Open Directory**. Then select the specific directory and again click on **Open Directory**. Finally, select the specific file that you want to monitor and select the severity. After clicking on **Monitor** and **Save**, monitoring is set up for that file.
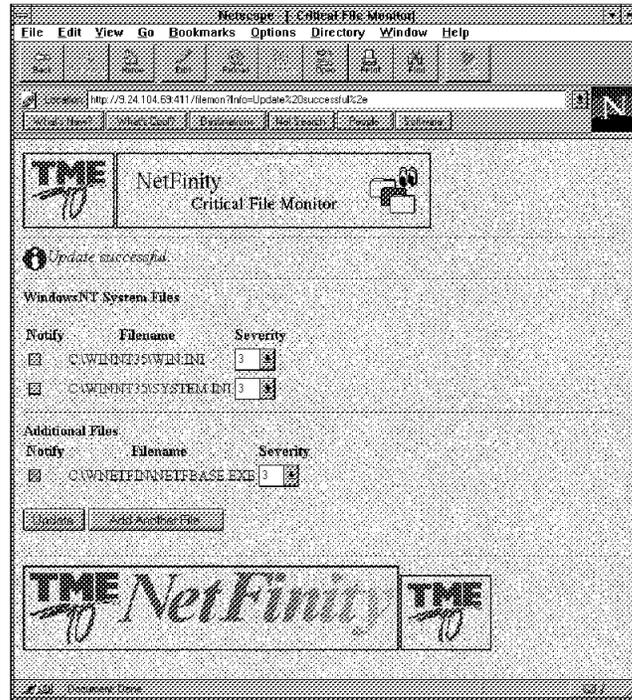
*Figure 48. Critical File Monitor - Additional File Monitor Added*

### 3.2.4 Event Scheduler

The Event Scheduler allows you to schedule actions to take place at a predefined time, or on a regular basis (for example, every Monday at 6 a.m.). Clicking on the hyperlink for the different tasks takes you to the definitions to set them up. Every task has a different entry Web page.
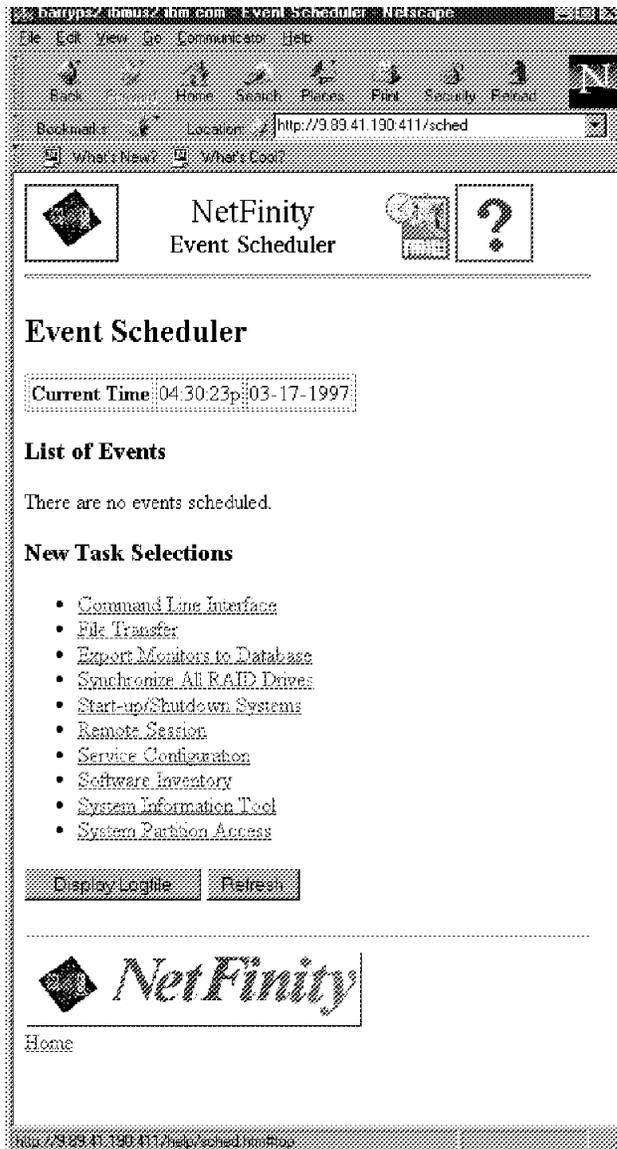
*Figure 49. Event Scheduler*

We investigated some of the tasks. The first one in the list in Figure 49 is to schedule a file transfer. You can copy files or directories either from the local machine to a remote system or vice versa. Another option would be to delete files on the remote system.
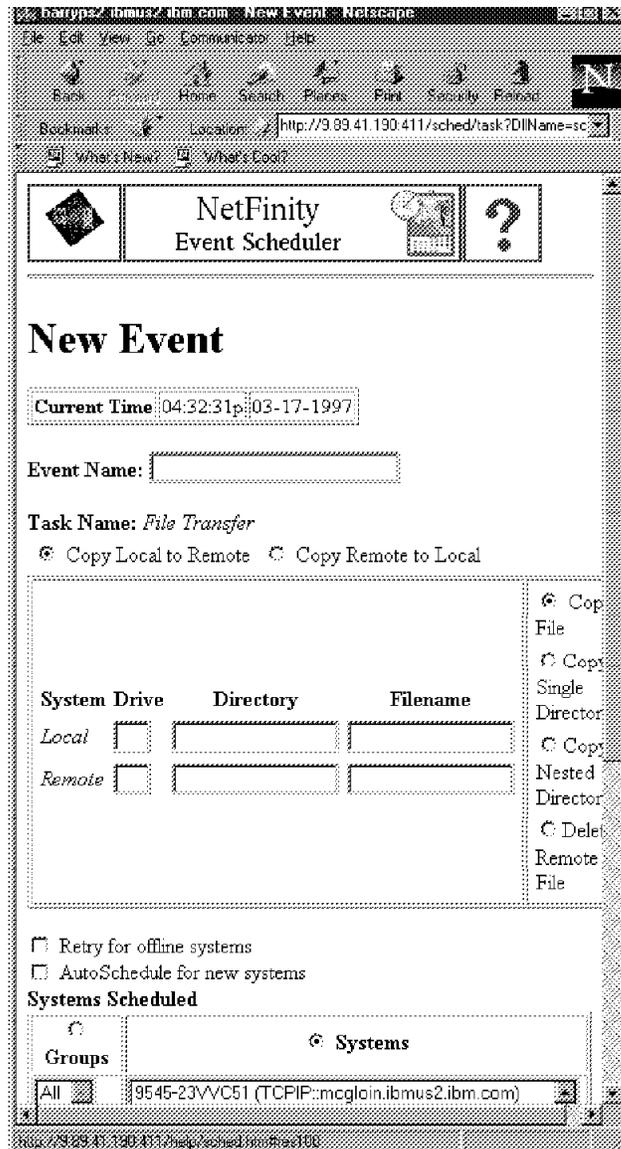
*Figure 50. Event Scheduler - File Transfer*

Synchronizing all RAID Drives does not need any additional parameters, so you only need to provide a name for the event and when you want the scheduled task to take place.
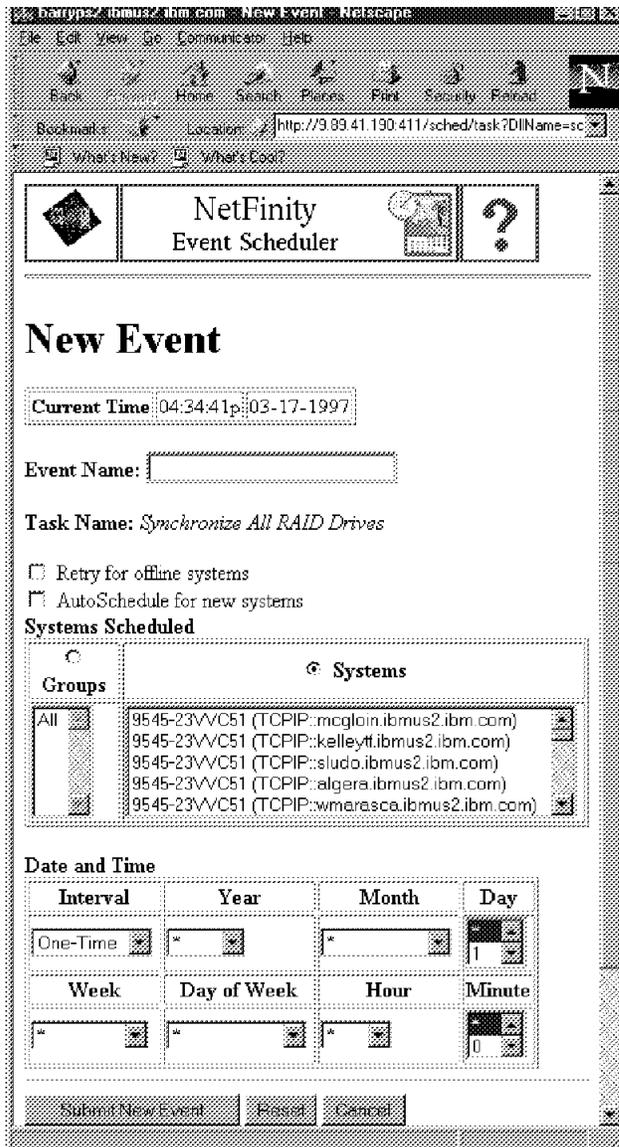
*Figure 51. Event Scheduler - Synchronize All RAID Drives*

You have four options to choose from when scheduling the startup or shutdown of a system or a group of systems. You can also specify the transport protocol on which the command will be issued. That can be helpful when there are scheduled network outages. For example, if the IP network is down, you may still be able to schedule the task by using NetBIOS or SNA.

1. Attempt System Restart

2. Attempt System Shutdown

3. Attempt System Wake-up

4. Attempt System Power-down

*Figure 52. Event Scheduler - Startup/Shutdown Systems*

If you want to schedule a command to execute on a remote system, just select
the **Remote Session** link and provide the command.
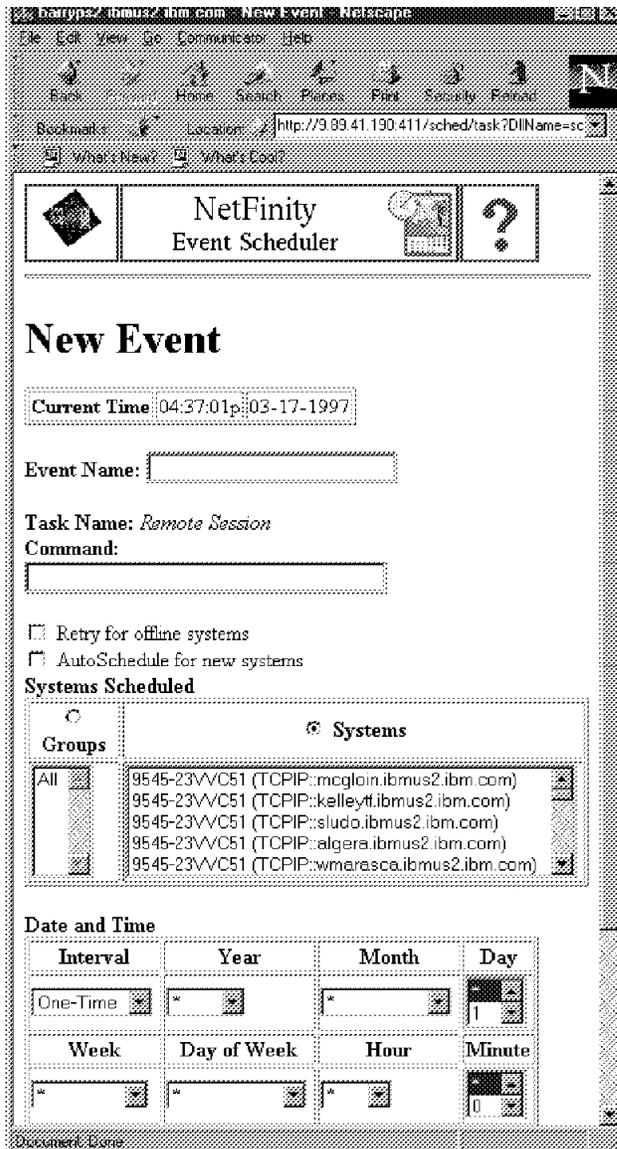
*Figure 53. Event Scheduler - Remote Session*

The Service Configuration Event is new since it belongs to the new Service Configuration Manager. This event can distribute a NetFinity 5.0 configuration file that was created using the Service Configuration Manager to selected NetFinity 5.0 machines. You can choose to either add the file to an existing configuration or to overwrite the old configuration file.
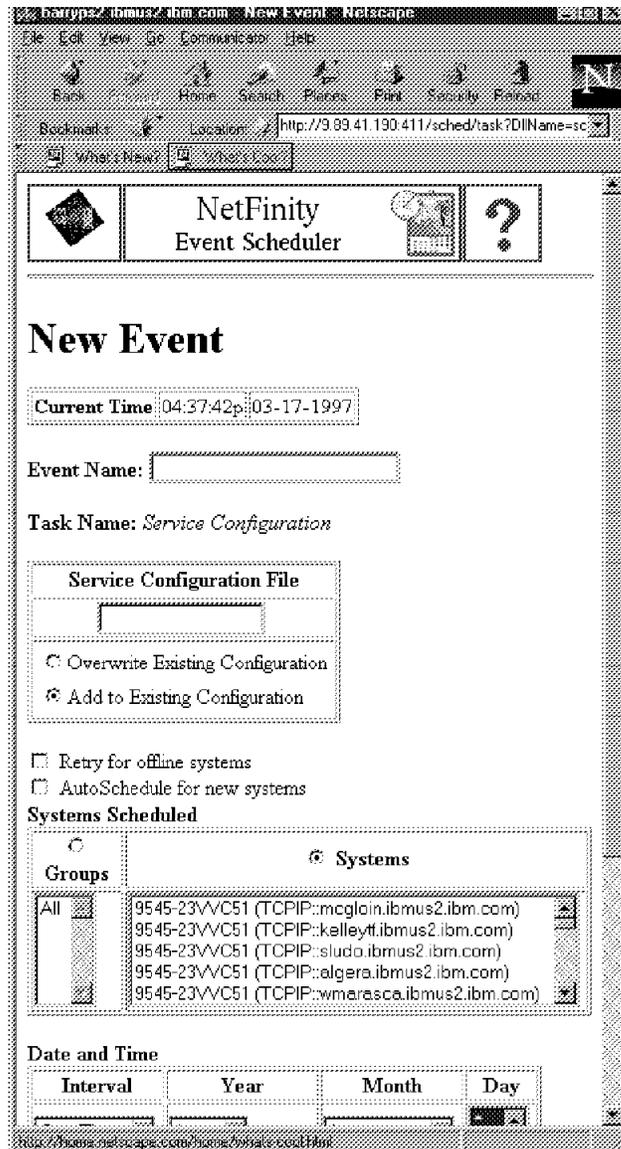
*Figure 54. Event Scheduler - Service Configuration*

To manage your installed software inventory it might be useful to schedule a software inventory task. You can specify a dictionary file that defines the products for which the software inventory task should search. There are several different reports that can be generated:

1. Generate Status Report: Specify the directory where the report is to be written.

2. Generate Summary Report by Product Name: Specify the filename for the report file.

3. Generate Summary Report by Product Version: Specify the filename for the report file.

4. Generate Summary Report by Product Revision: Specify the filename for the report file.

5. Update NetView DM inventory.

6. Update application keywords on client.

7. Update saved inventory on client.
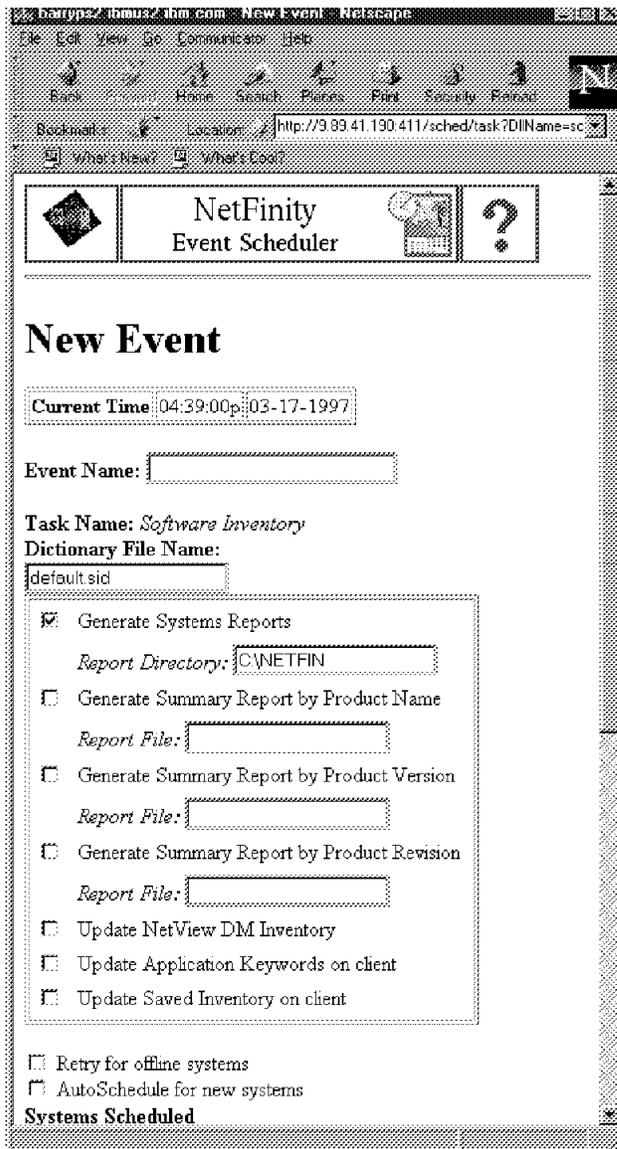


Figure 55. Event Scheduler - Software Inventory

When scheduling the system information tool you need a place to export the information. This can either be a NetFinity 5.0 history file, an ASCII file that gets sent to a printer, or a file name. A new feature of NetFinity 5.0 is that you can schedule a retry attempt for systems that are offline. In addition, you can automatically add newly discovered systems to the schedule as shown in Figure 59 on page 72.

*Figure 56. Event Scheduler - System Information Tool*

The System Partition is a section on the hard disk of some IBM systems, where system services such as Power-On Self Test or the BIOS are stored. With the NetFinity 5.0 System Partition Access feature you can back up, restore and manipulate this part of the disk. Actions you can schedule are as follows:

- Delete the partition file.

- Back up the partition.

- Restore the partition.

- Copy a file from the local system to the system partition.

- Copy a file from the system partition to the local system.
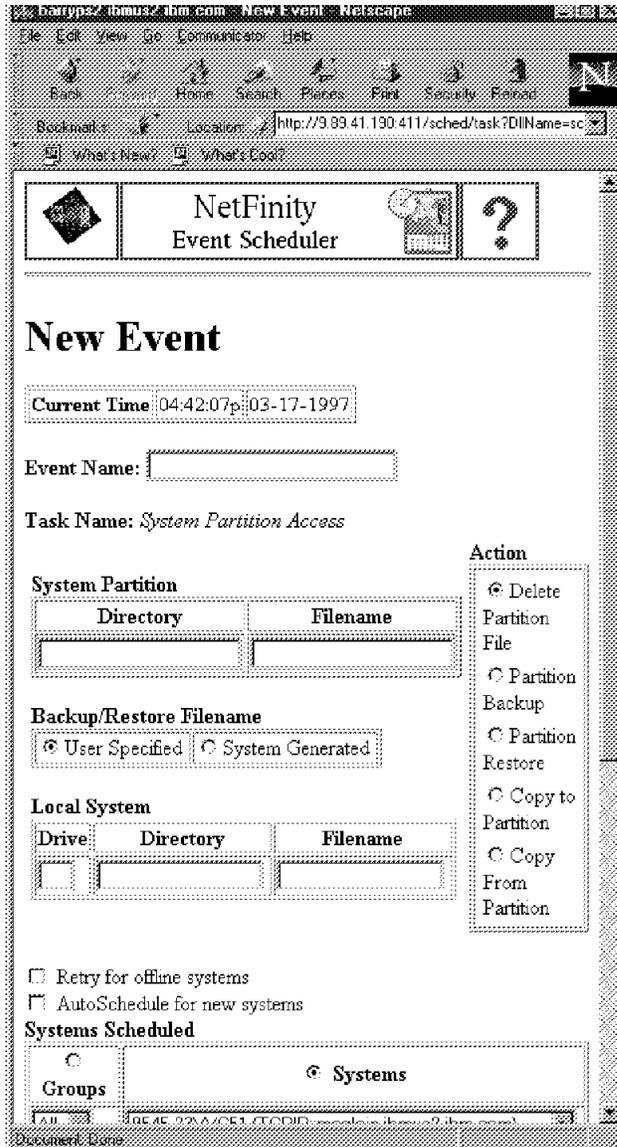
*Figure 57. Event Scheduler - System Partition Access*

Webability also lets you set up an event to issue a NetFinity 5.0 command using its command line interface. You could either use this to run a single command, or you could alternatively use the remote command event to run a script of commands:
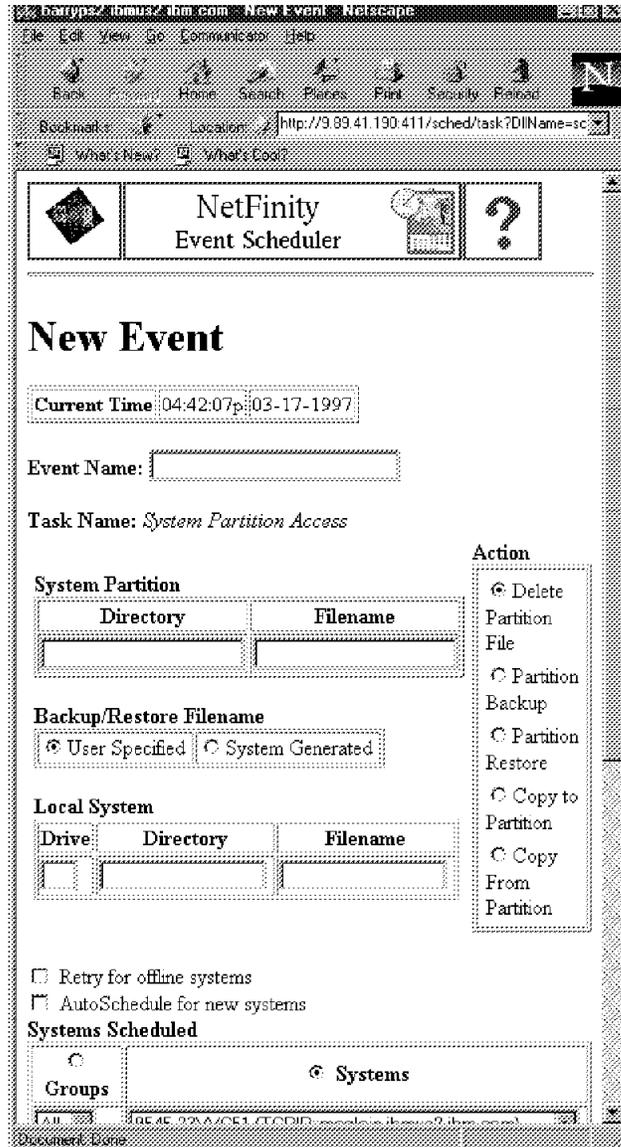
*Figure 58. Event Scheduler - Command Line Interface*

The last figure shows the section that is located directly under all the different event definitions shown in Figure 50 on page 63 to Figure 57 on page 70. It is divided into two parts. In the first part you can specify on which NetFinity 5.0 systems or groups the event shall run. The second part gives you the chance to specify how often and at what time the event should start on the selected machines.
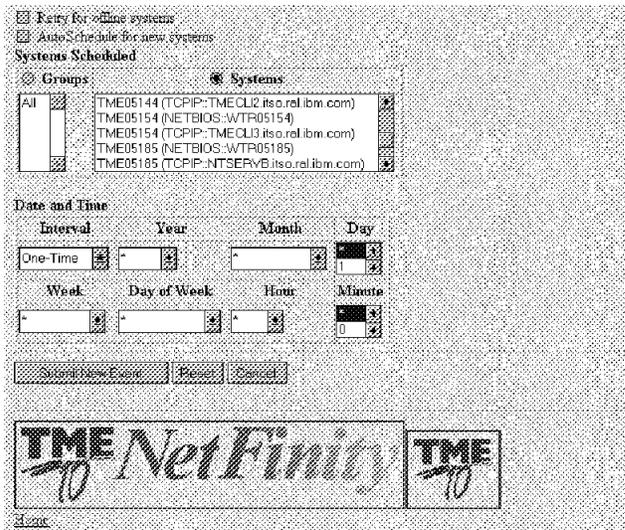
*Figure 59. Event Scheduler - Scheduler Section*

## 3.2.5 File Transfer

File transfer lets you send or receive files to or from the remote machine.
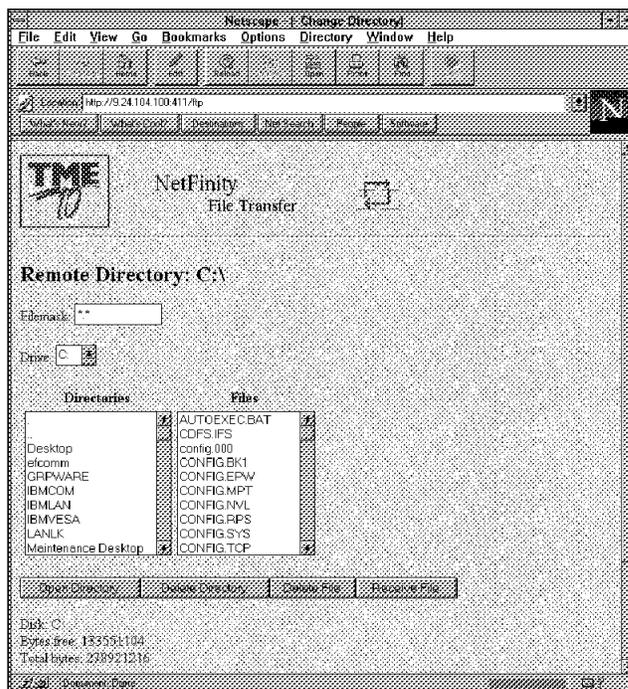


*Figure 60. File Transfer*

Select the file you want to receive on your local machine and click on **Receive File**.
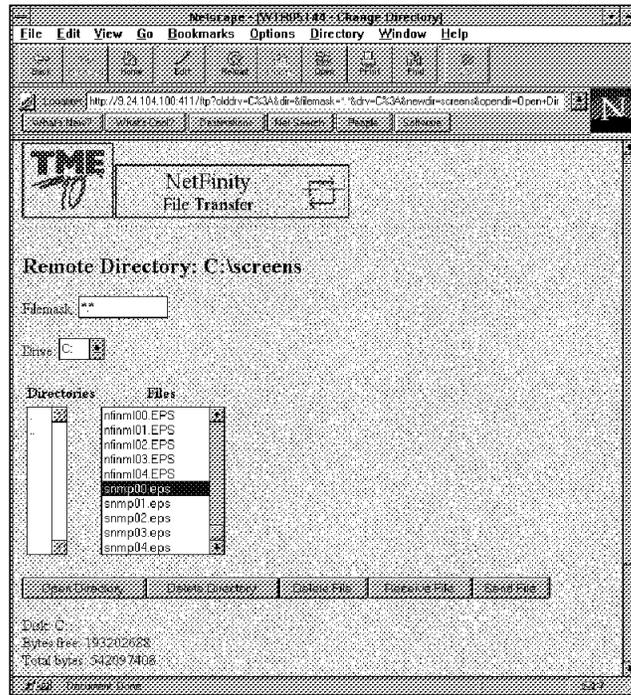
*Figure 61. File Transfer - Select File to Receive*

If Netscape doesn't have an application that is bound to this function, select **Save File**.



*Figure 62. File Transfer - Netscape Warning*

Select a location where to put the file and click on **OK**.
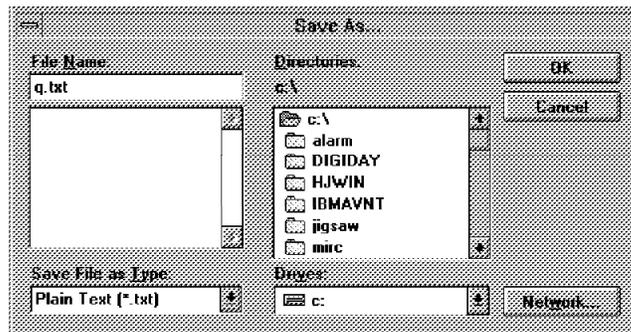


*Figure 63. File Transfer - Save As*

If you want to send a file to the remote machine, click on **Send File**. To select the file that you want to send, select **Browse**.
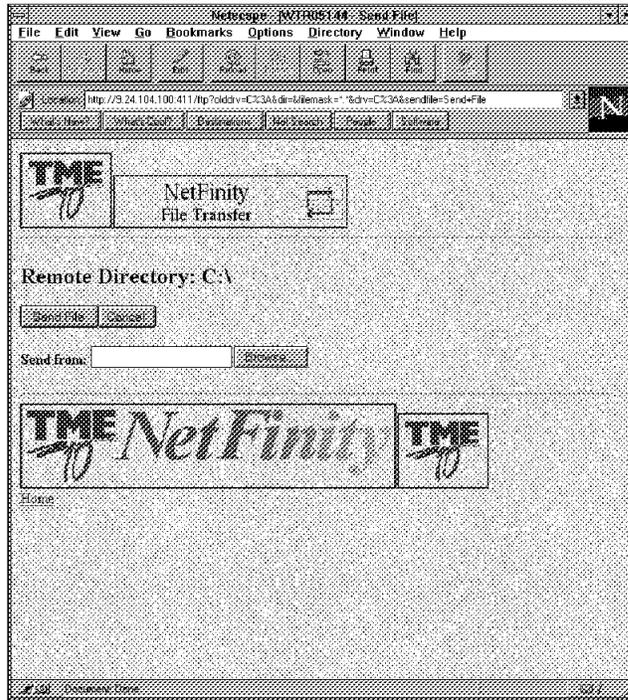
*Figure  64.  File Transfer - Send File*

In the File Upload window shown in Figure 65, select the files you wish to upload and then click on **OK**.
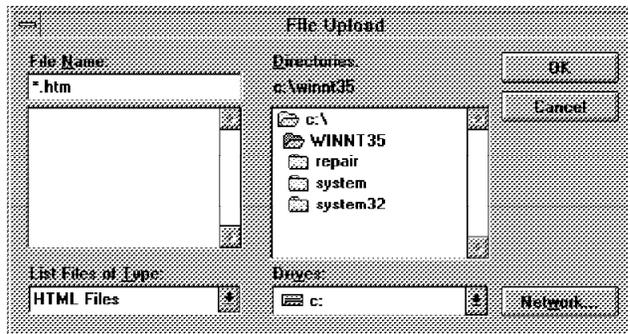


*Figure  65.  File Transfer - Send File*

### 3.2.6  Web Guru

A new Webability feature in NetFinity 5.0 is the Web Guru.  It is a service that helps guide you through the steps necessary to use the Webability features.  It requires a Web browser that supports frames.  The icon for the Guru function can be found on the main page of your NetFinity 5.0 Webability interface.  The Guru service does not show up on the regular GUI desktop interface.
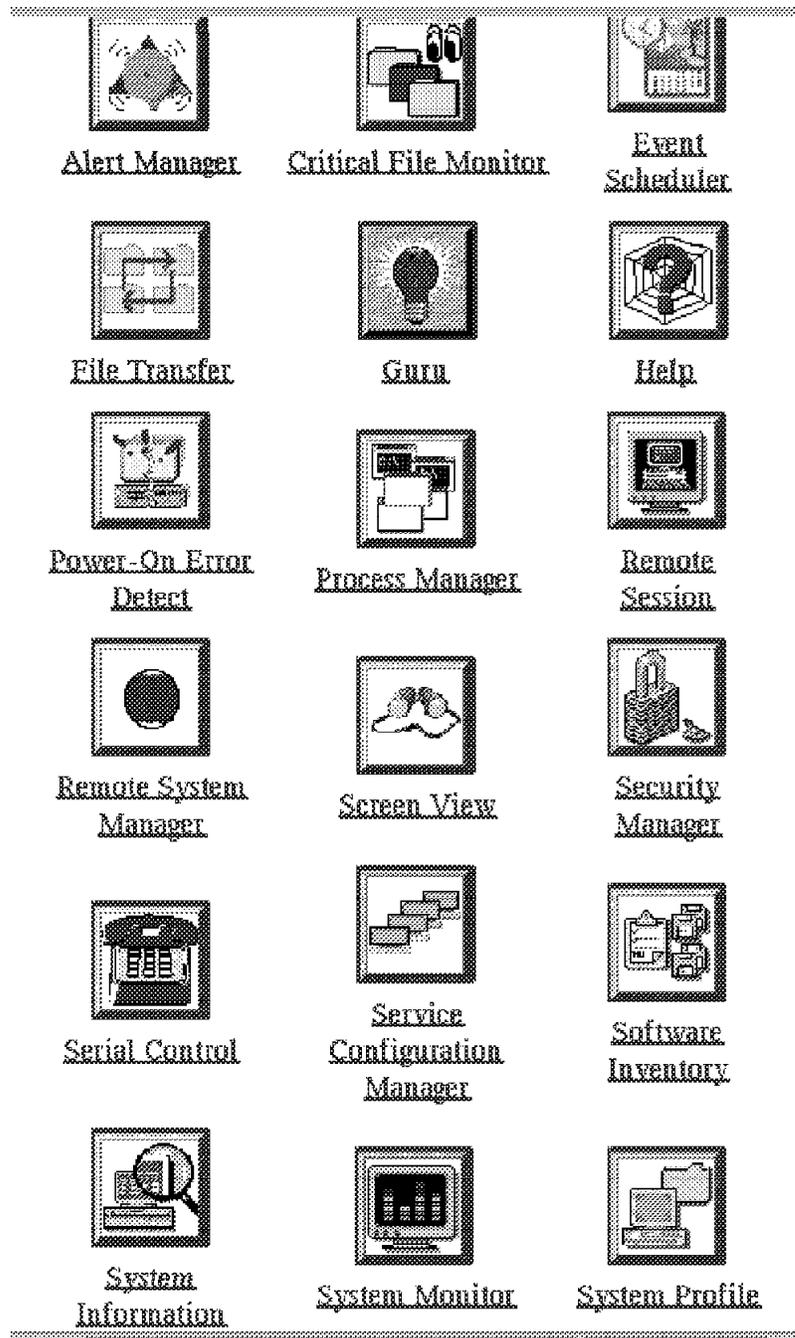
*Figure 66. Web Guru on Netscape Browser*

There are certain predefined procedures that the Guru function can walk you through. At the point that we were using beta code we were able to see the following services:

- Systems Monitor Guru

- Critical File Monitor Guru

- Process Manager Guru

- Service Configuration Guru

- Outgoing Security Guru
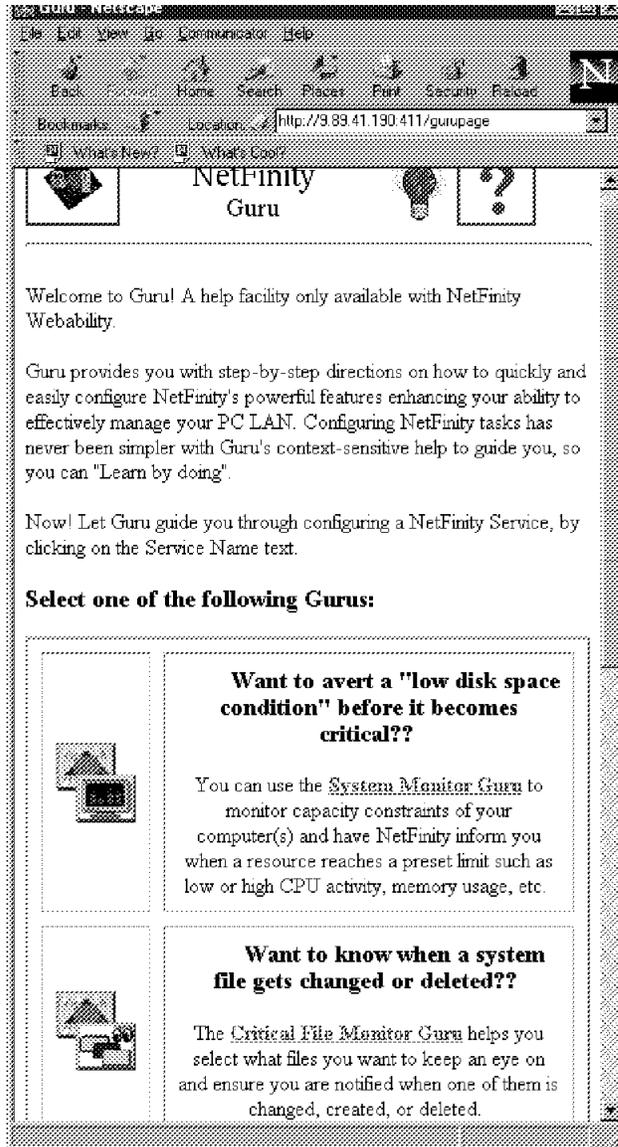
*Figure 67. Web Guru List of Functions*

Clicking on the **Systems Monitor Guru** link brings you to its Guru function which shows you all the monitors and their current values. It also provides documentation on what you can do.

**System Monitor Threshold Guru**

This guru helps you define an action for a threshold. You may use an existing threshold or create a new one.

To make a new threshold, pick the monitor that you would like to set a threshold on.

Click on one of the hypertext links.

NetFinity
System Monitor

**Monitors** Thresholds

| | |
|---|---|
| CPU Utilization | 65.9097 Percent |
| Process Count | 66 processes |
| Thread Count | 226 threads |
| Integer Instructions Rate | 11.7523 MIPS |
| Floating Point Operation Rate | 0.0159841 MFLOPS |
| Interrupt Rate | 383.992 Ints/Sec |
| Port I/O Rate | 29500 IOs/Sec |
| Memory I/O Rate | 29659.7 K IOs/Sec |
| CPU Cache Hit Rate | 98.1996 Percent |
| Drive C: Space Used | 85.4277 Megabytes Used |
| Drive D: Space Used | 177.151 Megabytes Used |
| Drive E: Space Used | 320.399 Megabytes Used |
| Drive F: Space Used | 43.1157 Megabytes Used |
| Drive C: Space Remaining | 14.4102 Megabytes Free |

*Figure 68. System Monitor Guru*

You can easily set up thresholds using the Guru.
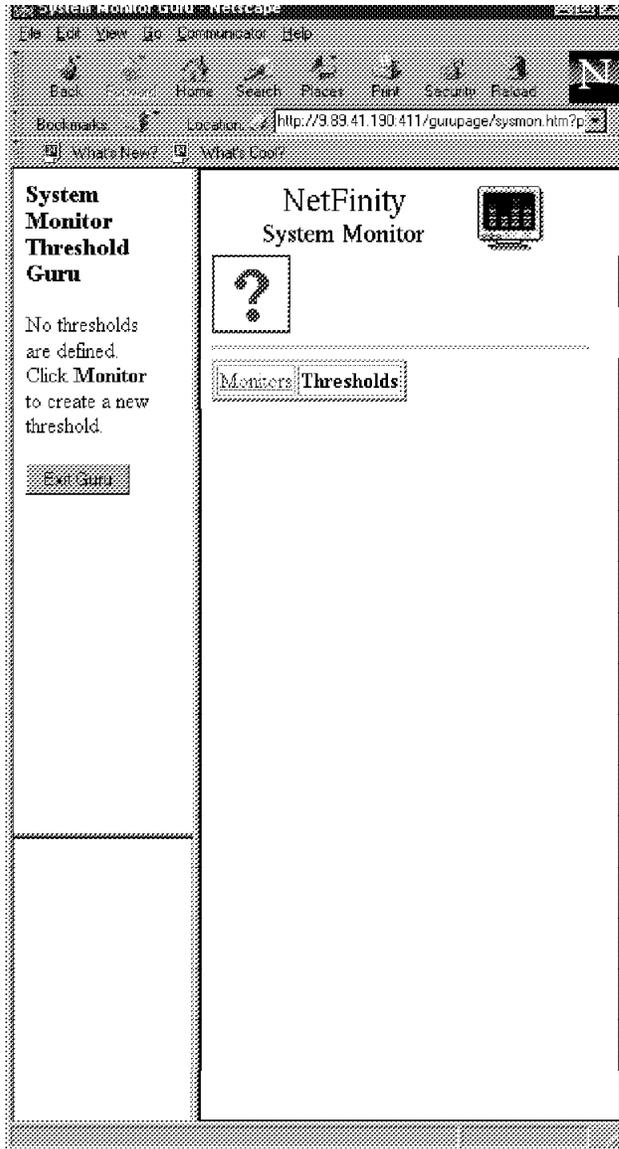
*Figure 69. System Monitor Thresholds*

Another service on which you can get guidance is the Critical File Monitor service. Just like the System Monitor Guru, it too will provide easy customization as well as text to guide you through the process.
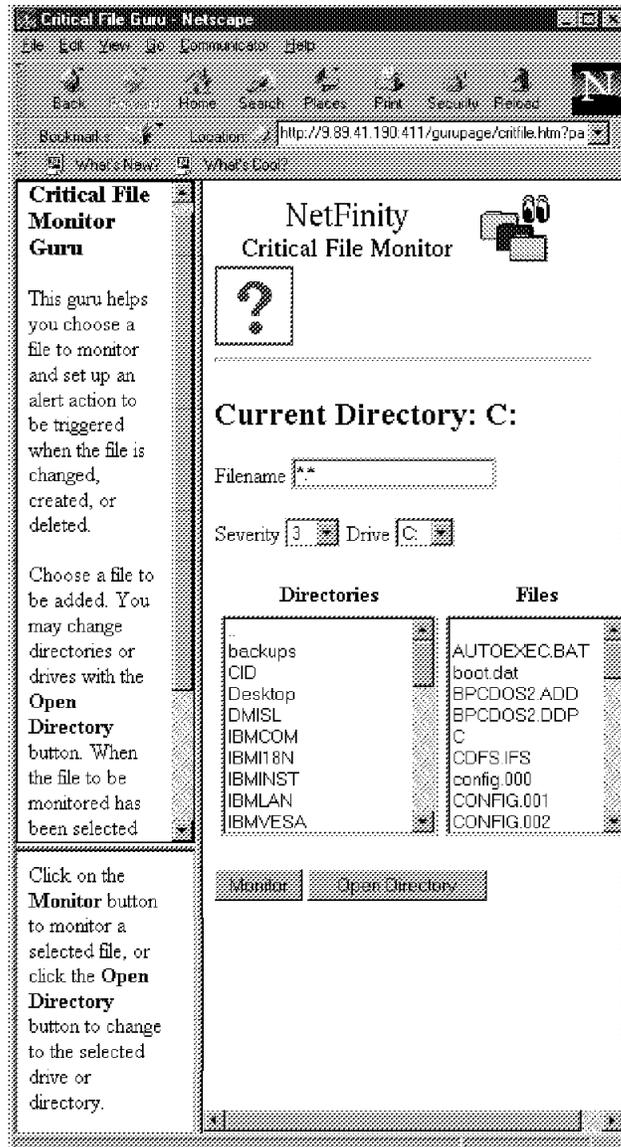
Figure 70. Critical File Monitor Guru

The Process Manager Service can also be customized using the Guru function.

**Process Manager Alert Guru**

This guru helps you define an action for a Process Alert. You may use an existing process alert or create a new one.

To make a new process alert, pick the process that you would like to set an alert

Click on the link in the above or the side frame.

NetFinity
Process Manager

**?**

IBM OS/2 Warp, Version: 3.0

**Process List** Process Alerts

| Type | Program Name |
|------|-------------|
| | D:\CMLIB\ACSRASP.EXE |
| | D:\CMLIB\CMKFMSML.EXE |
| | C:\MUGLIB\MUGLROST.EXE |
| | C:\IBMLAN\SERVICES\PEER.E |
| | C:\IBMLAN\SERVICES\PEER.E |
| | C:\IBMLAN\SERVICES\MSRV.I |
| | C:\IBMLAN\SERVICES\WKSTA |
| | C:\IBMLAN\SERVICES\WKSTA |
| | C:\OS2\EPWMUX.EXE |
| | C:\OS2\EPWMUX.EXE |
| | C:\OS2\EPWPS.EXE |

*Figure 71. Process Manager Guru*

Clicking on any process provides you with a Web page that lets you customize what alerts should be sent if the status of this program changes.

**Process Manager Alert Guru**

To add a new Process Alert:

1. Make sure **Program Name** contains the name of the program for which the alert will be generated

Click on **Program Name** to get started.

NetFinity
Process Manager

**Program Name:**
C:\OS2\CMD.EXE

Alert Severity: 0

✖ Generate alert when program runs

✖ Generate alert when program stops

✖ Generate alert if program not started

Execution time-out: 1 min

⟳Notify ✔

Add

*Figure 72. Process Manager Alerts*

In order to set up for mass customization and distribution of profiles you can use the new Service Configuration Manager component. Since it is new, it would be beneficial to use the Guru function so that you can get instructions as you first start using it.
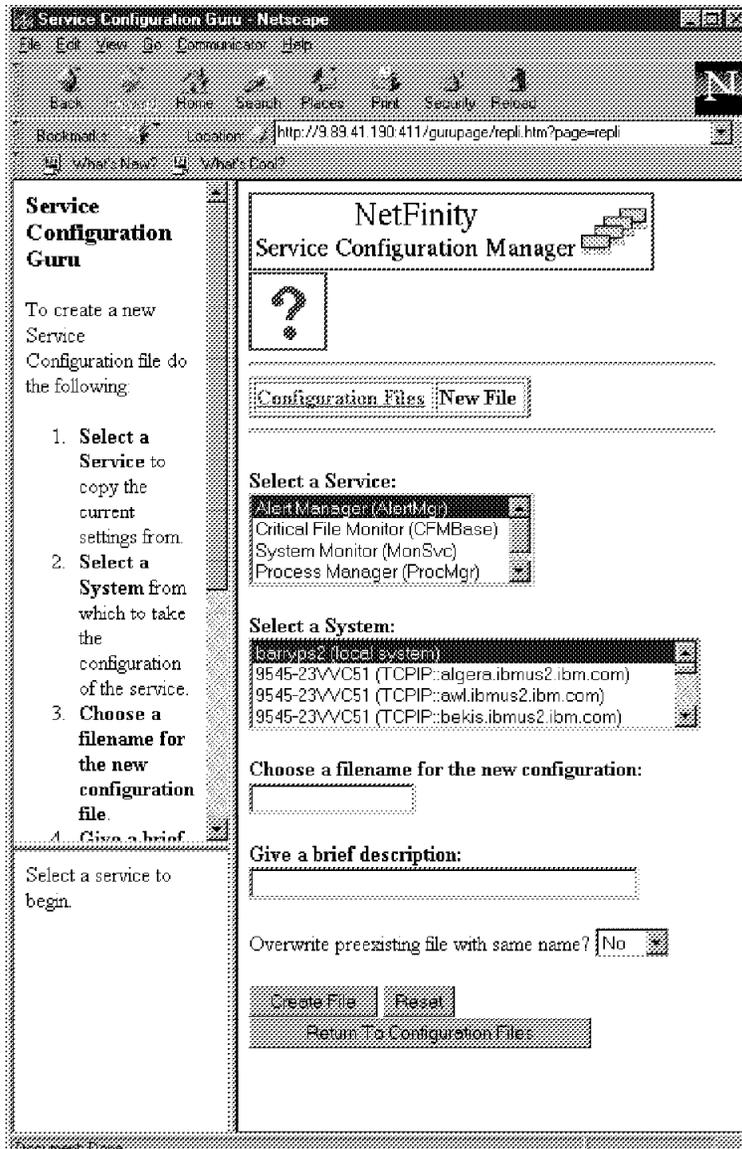
*Figure 73. Service Configuration Manager Guru*

The Outgoing Password Guru is tied to the Remote System Manager service, as that is the application that you would be using to connect to other systems.
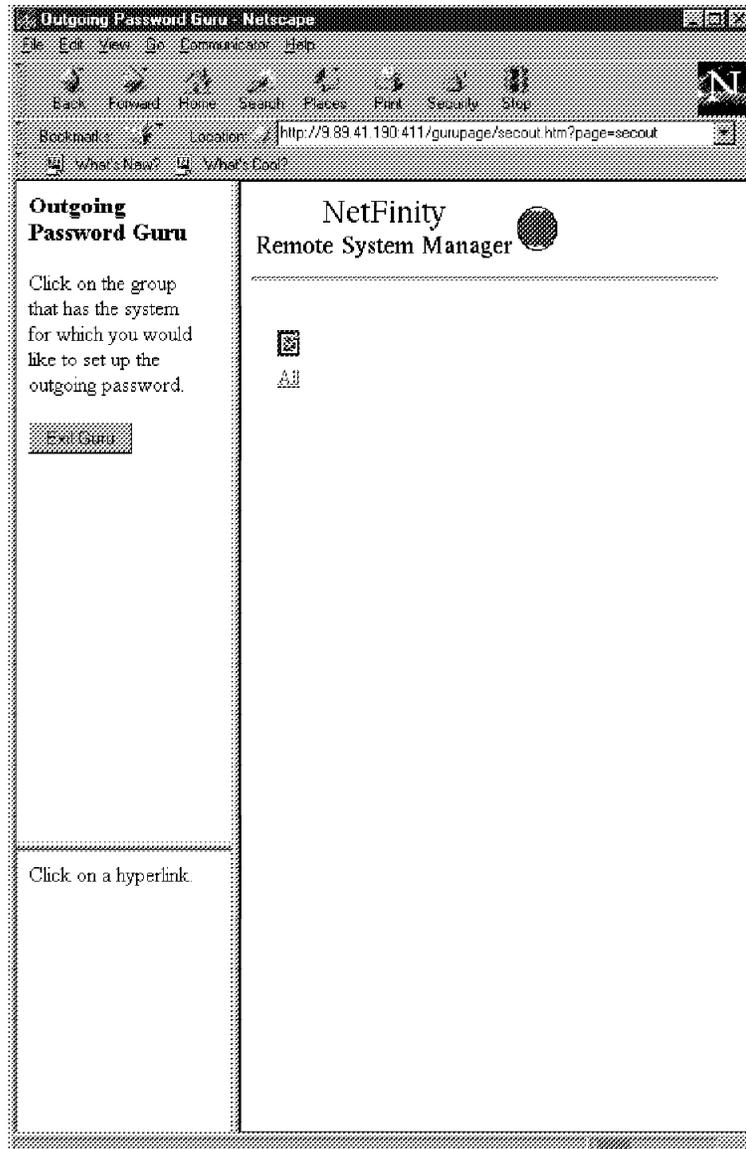
*Figure 74. Outgoing Password Guru*

If you click on one of the groups (in this case **All**), you will get an icon for each of the systems. The icon will be color coded to show its operational status. Green means the system is up and accessible while red means it can not currently be reached.
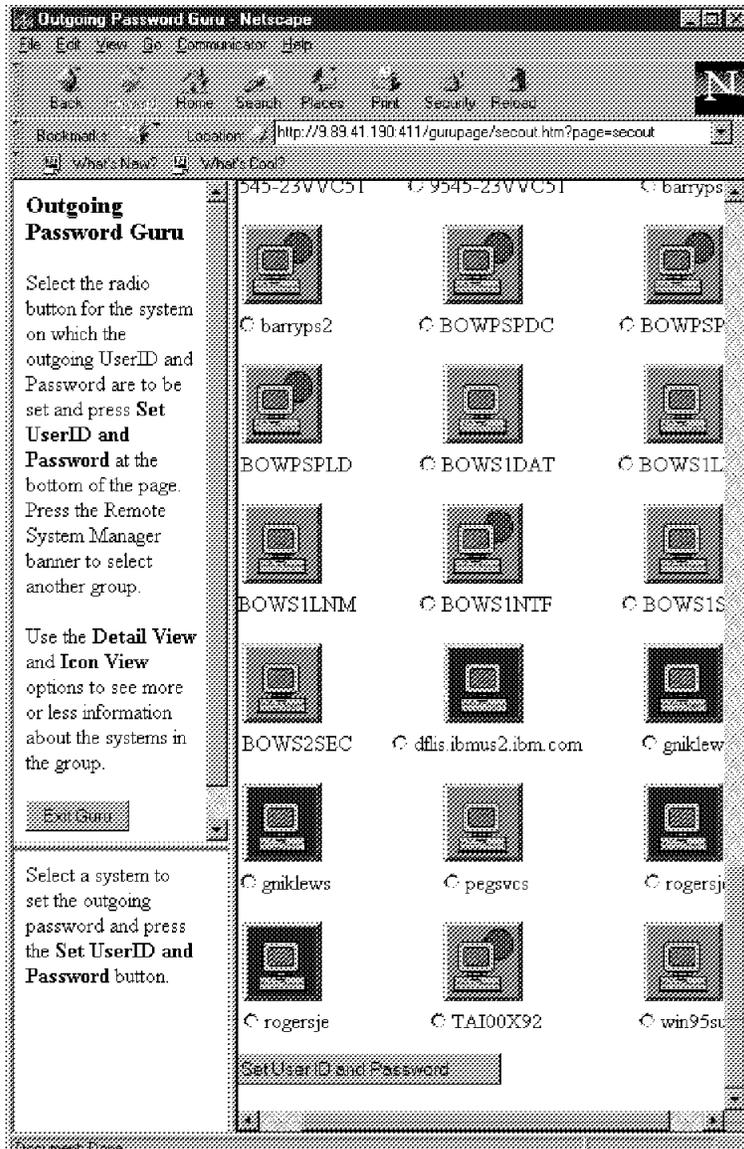
*Figure 75. Remote Systems*

There is also a new online Web help facility. Just like the Guru function, it is available through the Web browser but not through the regular GUI.
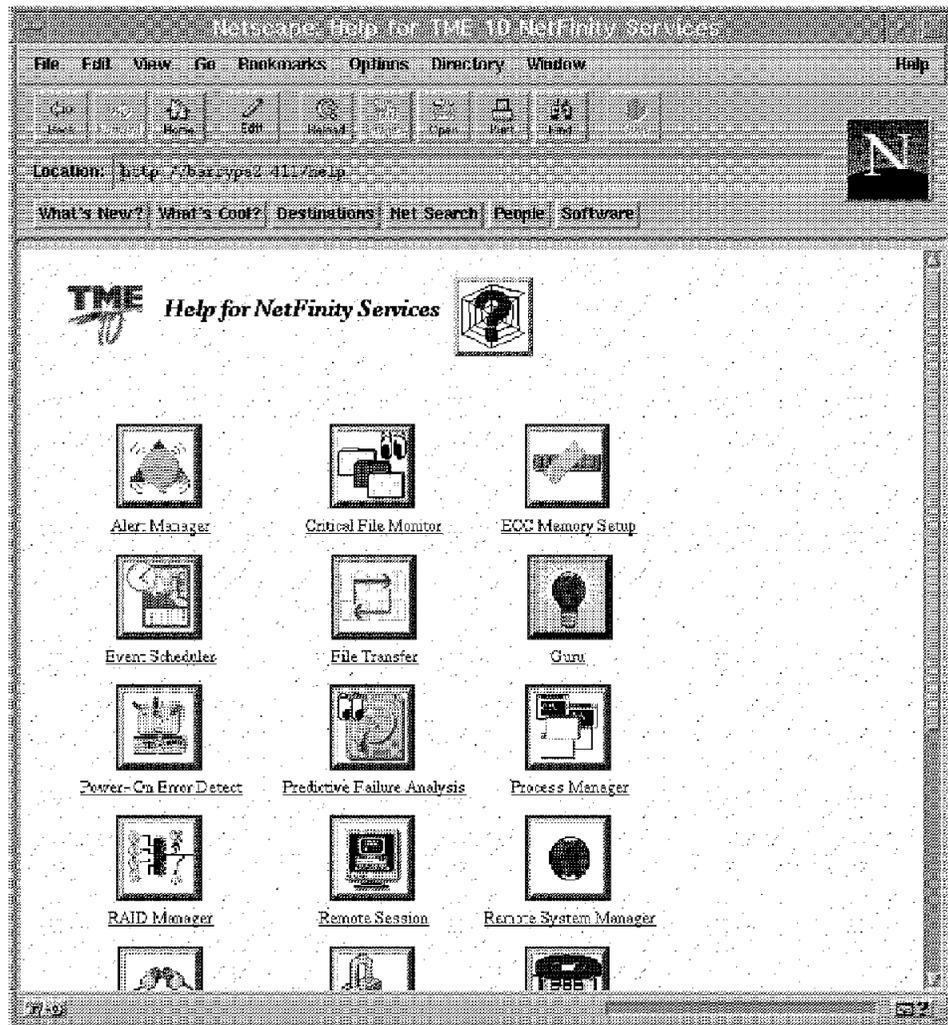
*Figure 76. Web Help*

Each of the services has help associated with it. The following Web page is an example of help for the Alert Manager function.

# Help for Alert Manager

The **Alert Manager** service is an extendable facility that allows receiving and processing of application-generated alerts. A variety of actions can be taken in response to alerts, including alert logging, pop-up user notification, forwarding the alert to another system, program execution, or an application-defined action. Features of this service include:

- Simplifying application use and processing of alerts
- Providing standard responses to alerts, including:
  - Log alert to file
  - Display alert in pop-up window
  - Forward alert to another workstation
  - Execute a program
  - Dial-out to a digital pager
  - Generate an SNMP version of the alert
  - Play a waveform (WAV) file
- Providing extensively configurable alert management and action generation

The Alert Manager's functions are activated from several links in the Alert Manager screen. These links are:

- Alert Log
- Histograms of Alert Log
- Monitor Alert Log
- Alert Profiles
- Alert Actions
- Generate Alert

*Figure 77. Help Examples*

## 3.2.7 Power-On Error Detection

The POED (Power-On Error Detect) function consists of special drivers that are to be installed on the system partition on a Micro Channel system. These drivers become active when the system loads the system programs from the system partition. This is before the operating system boots and before the POST (Power-On Self Test). If the system detects an error in the POST, it sends some special messages to the NetBIOS network which shows the POST error. Since this happens before the operating system is loaded you do not need NetFinity 5.0 installed on the system to do the actual broadcast of the alert. However, if you wish to view the alert, you will need to have NetFinity 5.0 manager code

installed.  If you want to use POED you have to have a Micro Channel machine
attached to a LAN, the POED drivers installed on the system partition, a
supported LAN adapter and the NetBIOS protocol enabled.  Here is a partial list
of supported adapters:

- IBM Token-Ring Adapter

- IBM Ethernet Adapter

- LAN Streamer 32 Adapter

- LAN Streamer 16 Adapter

- 3Com EtherLink/MC Adapter

- SMC Ethernet Elite Plus/A Adapter

- Madge Smart 16/4 Ringnode Adapter

- Ether Streamer Adapter
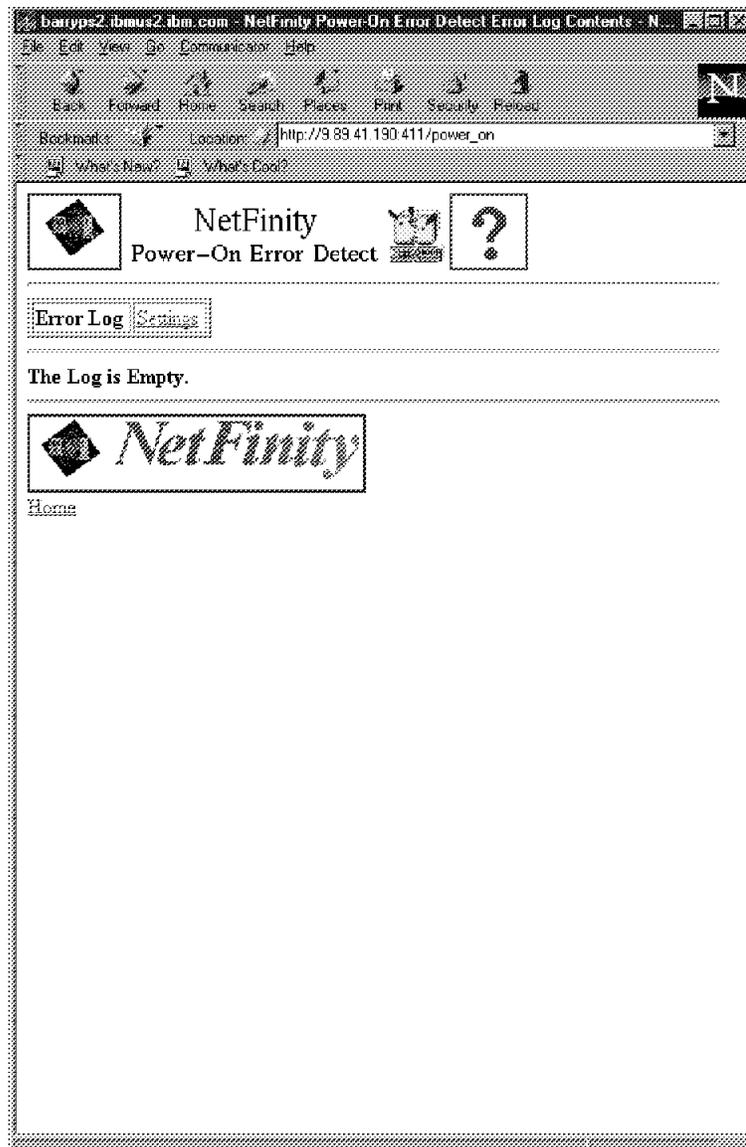


*Figure  78.  Power-On Error Detection (POED)*

On the POED settings screen you must enable at least one of the settings by clicking on **Yes**. Then click on **Save** to activate POED.
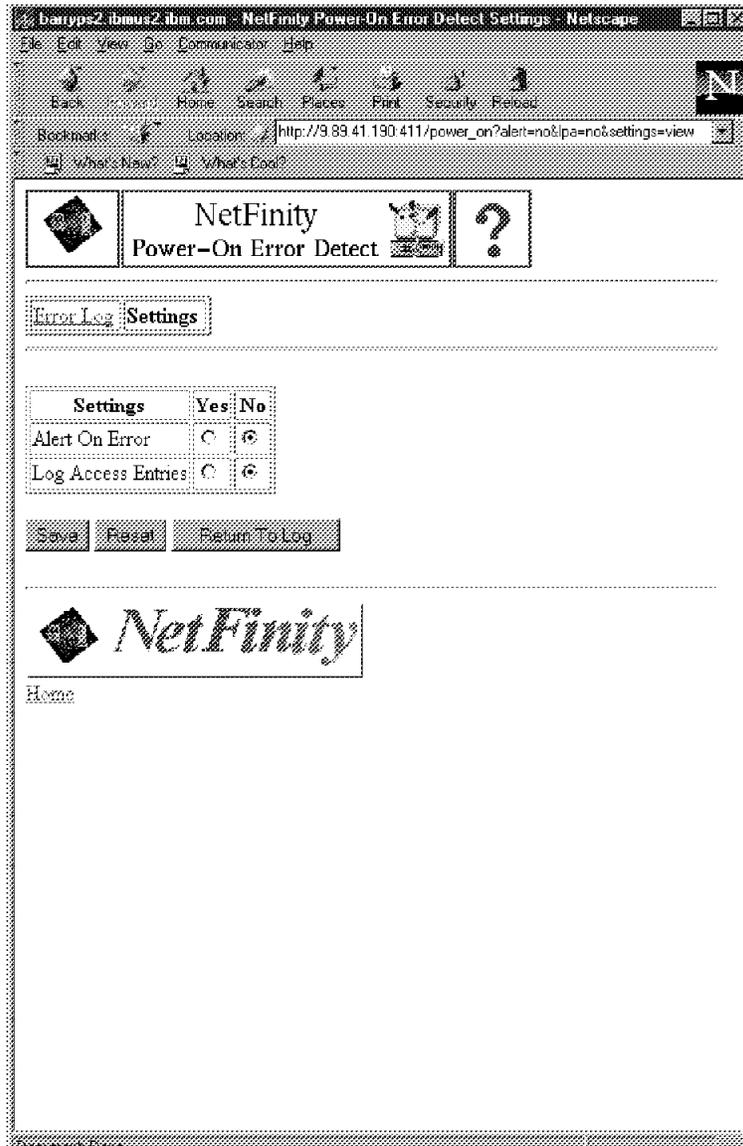


*Figure 79. POED - Enable Settings*

### 3.2.8 Process Manager

The Process Manager main Web page lists all active processes on the machine that we are monitoring with the Web browser. That can be either the local machine, or it can be a system that is accessed through the Remote System Manager service.

*Figure 80. Process Manager on OS/2*

You can select one or more and send a Ctrl-C or Ctrl-Break to the process or even attempt a kill process. As in the GUI it is also possible to issue a command to start a new process.

The view of NT processes is slightly different, but the concept is the same.

To set an alert for a process, either click on the hyperlink of a running process or click on **Process Alerts**. If you already had a list of process alerts, it would show up in the Web page. In the following figure it was our first process alert.
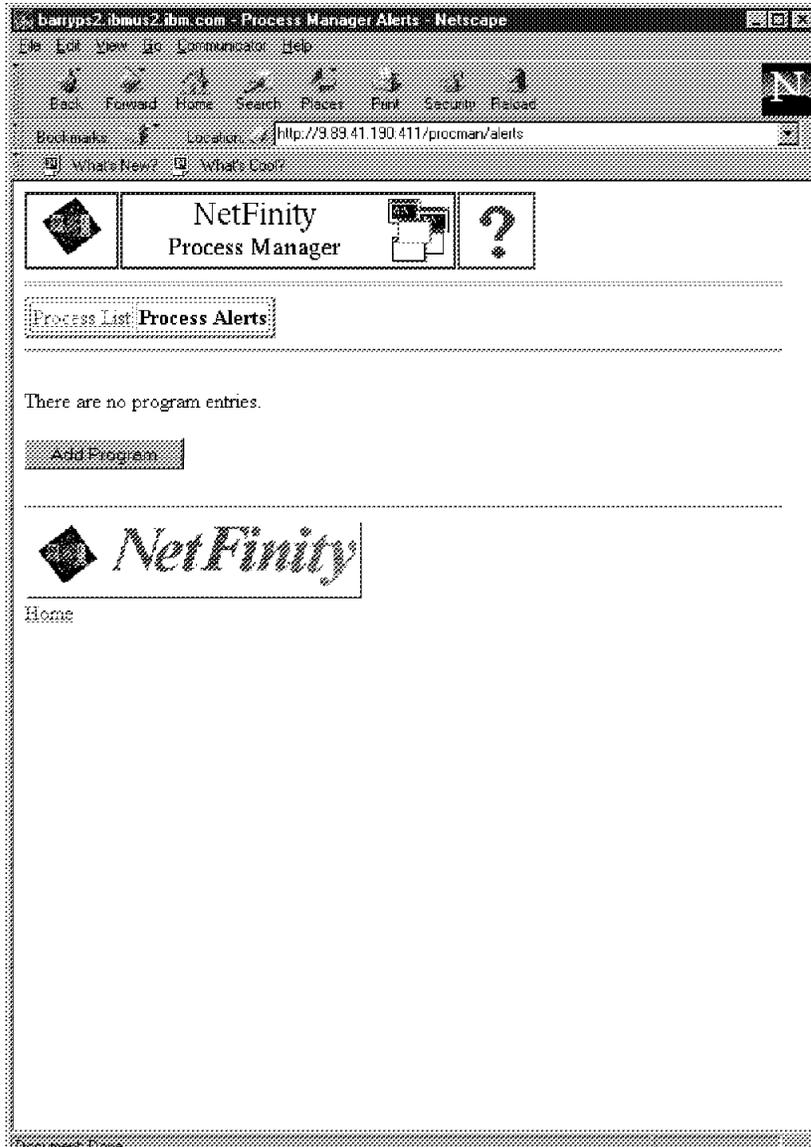
*Figure 81. Process Manager - Process Alerts*

You can click on **Add Program** to get to the alert definition page. You can define an alert to be sent when it starts, when it abends or if it never starts at all. You can also specify the severity. Click on **Add** and you get back to the process alerts page.
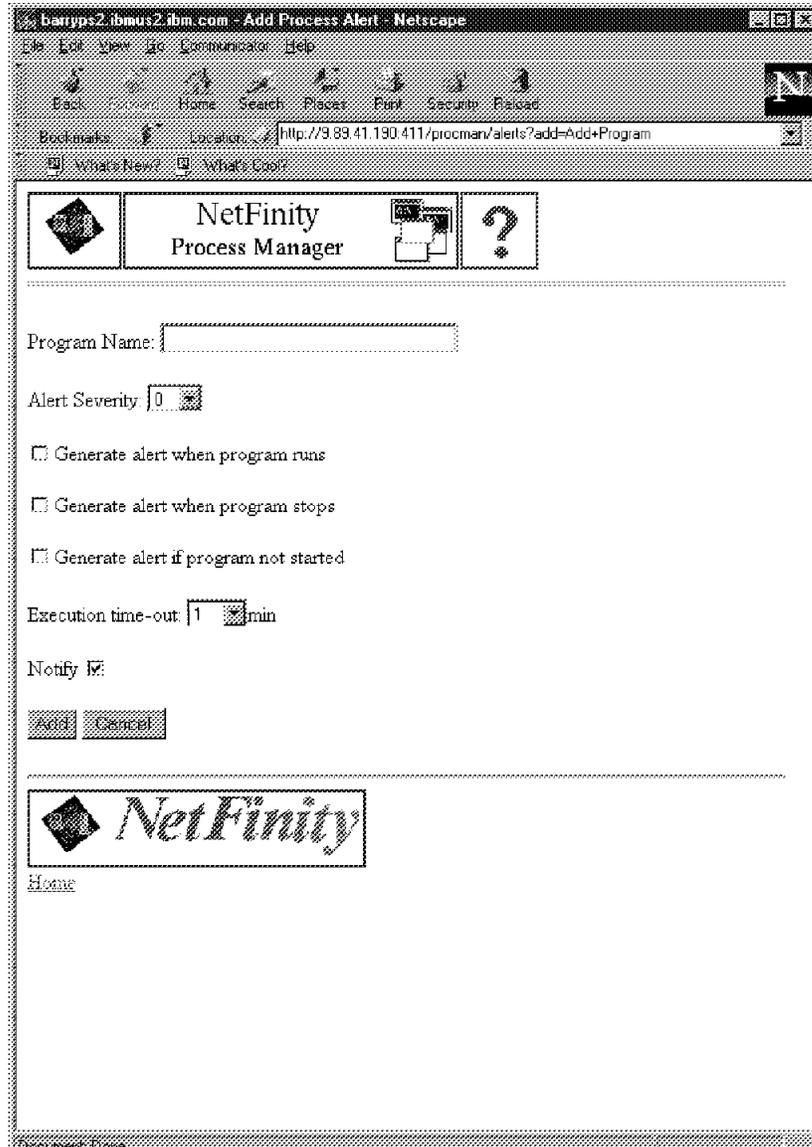
*Figure 82. Process Manager - Add Alert*

Once you have defined some process alerts, you can easily modify them by clicking on the field you wish to update.
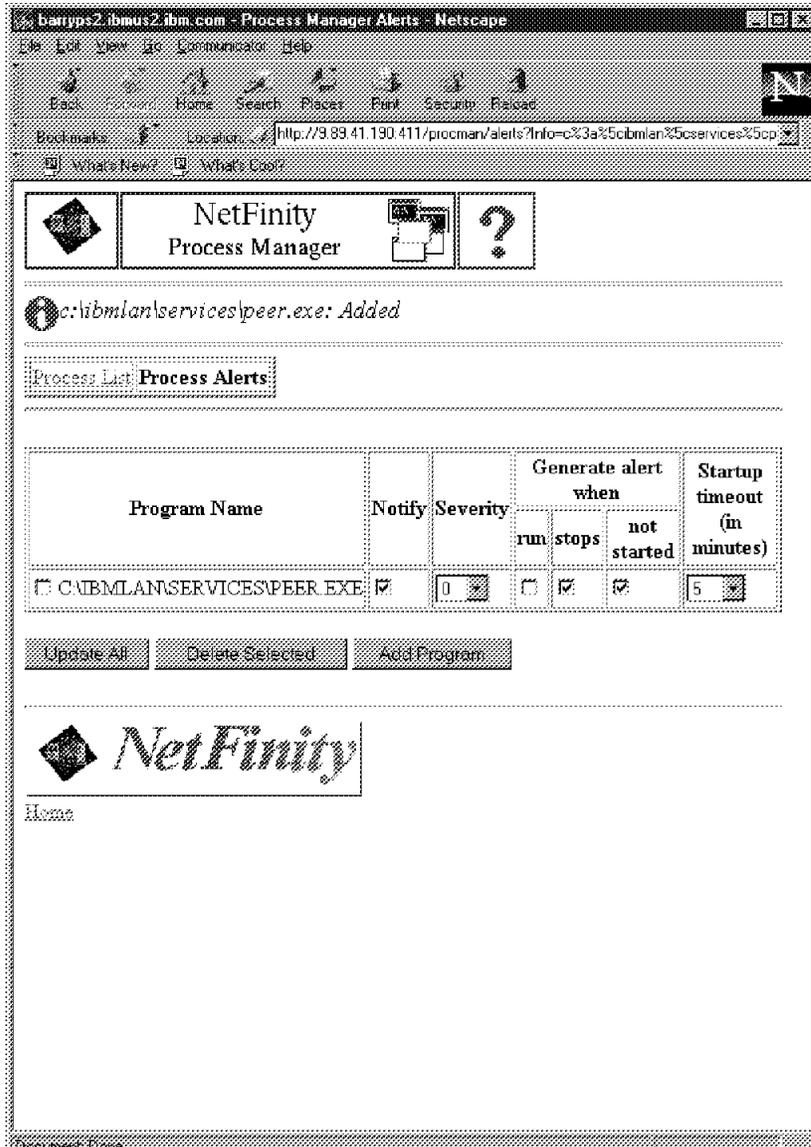
*Figure 83. Process Manager - Process Alert Defined*

## 3.2.9 Remote Session

---
**Note**

In order to use the remote session in Webability, you have to use a Java-enabled browser. We used Netscape 3.01.

---

You can start the remote session using the Web browser. It opens up a Java Applet window emulating a remote session.
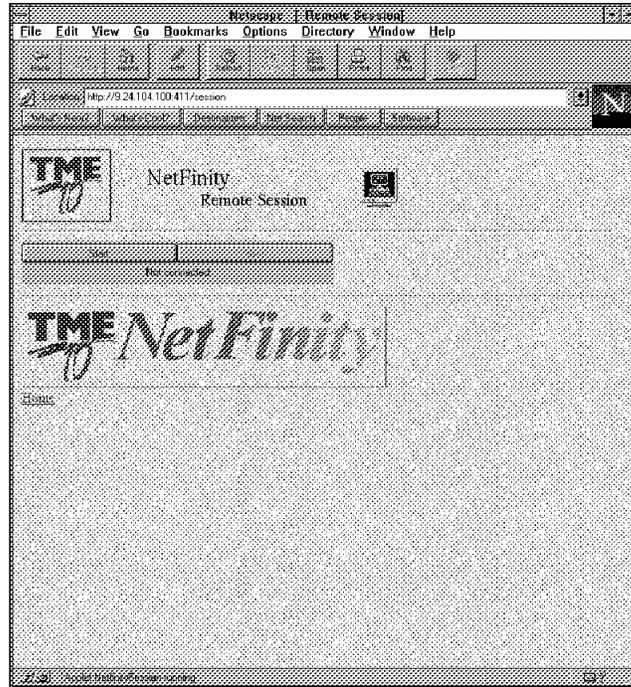
*Figure 84. Remote Session*

You can then issue any line mode command in that window. Anything that you can enter in a regular window can be entered in this Java window. You do not have any access to other windows or the desktop.

### 3.2.10  Remote System Manager

In case you want to manage a remote machine, you can use the NetFinity 5.0 Remote Systems Manager over the Web. As with the GUI, the main page shows the groups defined. From the Remote System Manager Web page you can:

- Add a new group
- Add a rack group
- Set up dynamic addressing
- Initiate discovery in all groups
- Set up system notification defaults
- Edit an existing group
- Delete a group
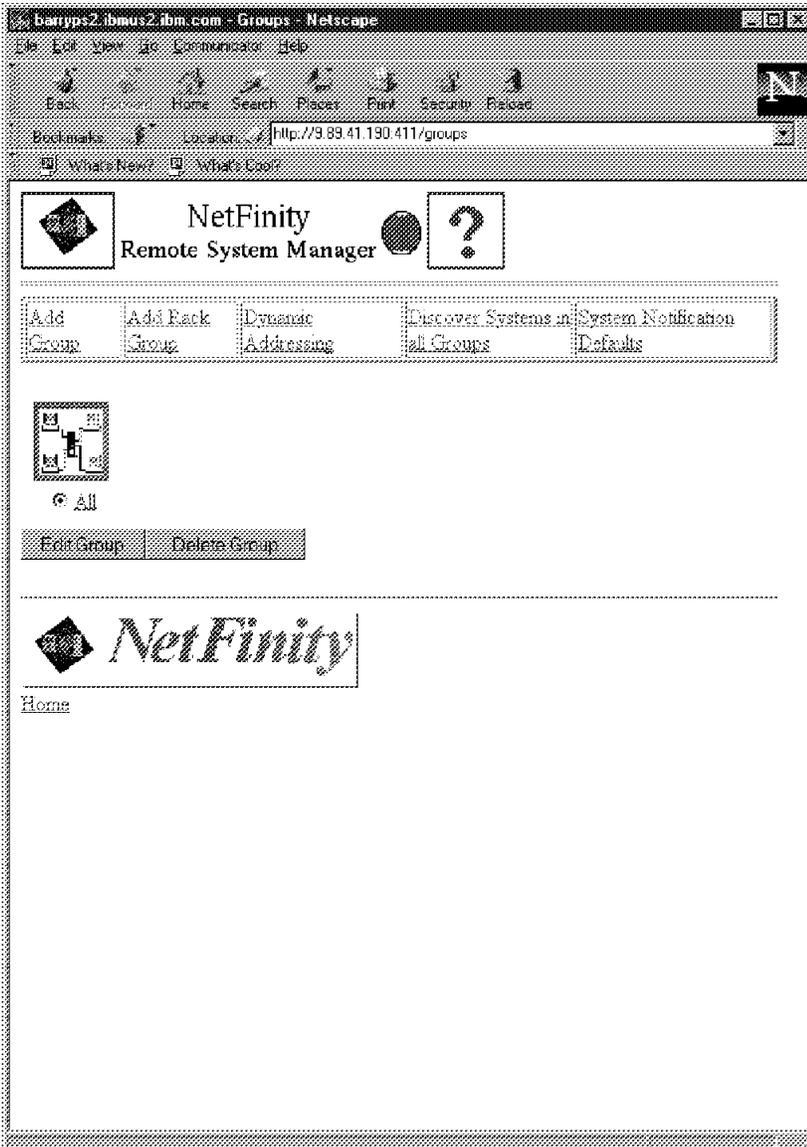
*Figure 85. Remote System Manager*

There are two types of groups you can add.  The first group is the normal NetFinity 5.0 Group.
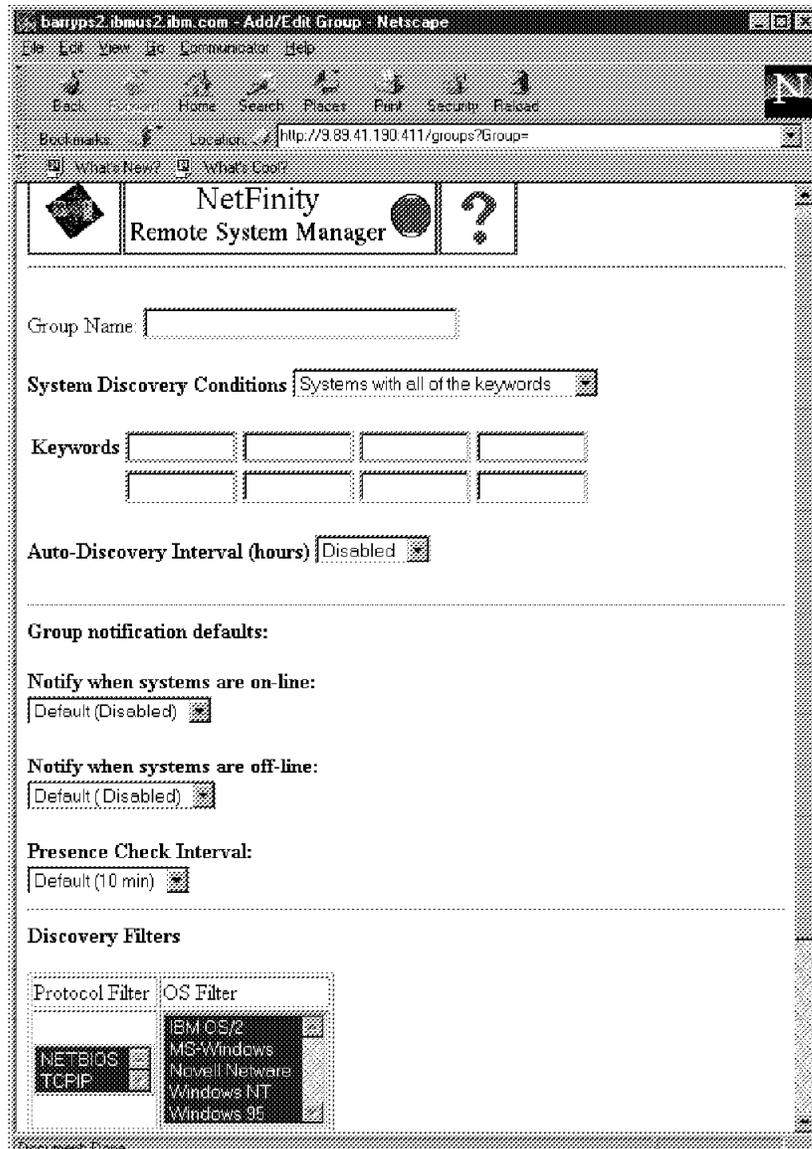
*Figure 86. Remote System Manager - Add a Group*

The new type of group that can be added with this release is called a rack group.
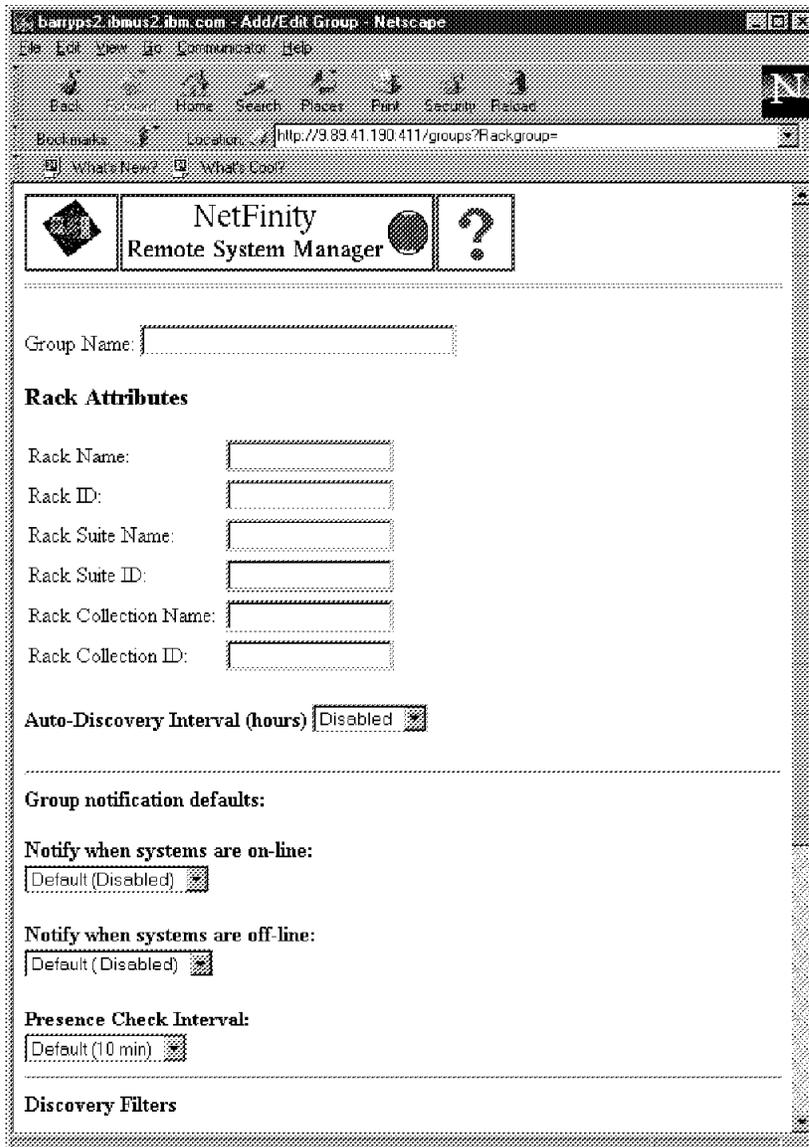
*Figure 87. Remote System Manager - Add a Rackgroup*

Clicking on a group or its hyperlink gets us onto the group page, which shows the systems matching the given criteria for the group keywords.
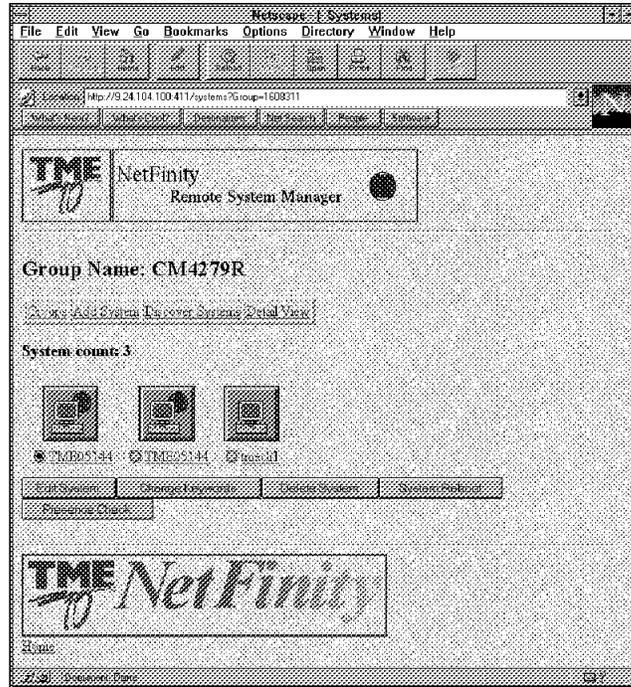
*Figure 88. Remote System Manager - Group Page*

Clicking on the **Discover Systems** link refreshes the screen. If you click on the system name beneath the icon, you will get a choice of different services. Examples of this could be:

- Web Guru

- Help

- System Monitor

---
**Important**

Unfortunately the Login System function available in the GUI is not in the Webability version of NetFinity 5.0 Remote Systems Manager. Therefore, you have to make sure that you have defined an outgoing password for the machine you want to access. This outgoing password *must* match with the incoming password for this user on the remote machine.

---

# NetFinity
## Remote System Manager
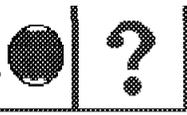
# Group Name: All

| Groups | Add System | Discover Systems | Detail View | Refresh |
|--------|-----------|------------------|-------------|---------|

## System count: 58



| 05303 | aces02 | ARCADIA |
| bethq16f | BFLEMING | BigBad95 |
| Bigmother | bnusbaum | bnusbaum |
| BORENX69 | CLIENT185 | CLIENT298 |
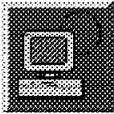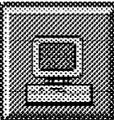
*Figure 89. Remote System Manager - Group Page after Discover Systems*

In addition to discovering systems, you can manually add new systems. This may be because they are outside of the broadcast discovery network or they could be for a system that is not online yet.
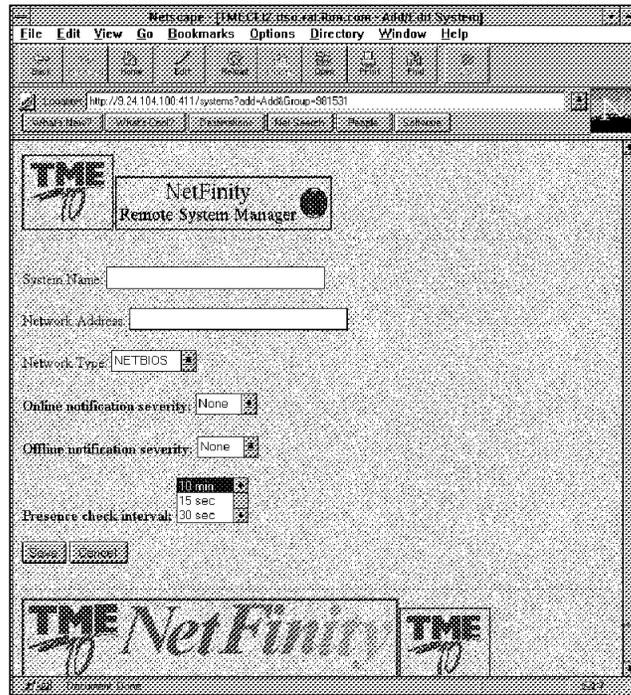
*Figure 90. Remote System Manager - Add System*

Like the GUI, the Web page also gives you a second way to view your group. The detailed view shows a lot of information at first sight.
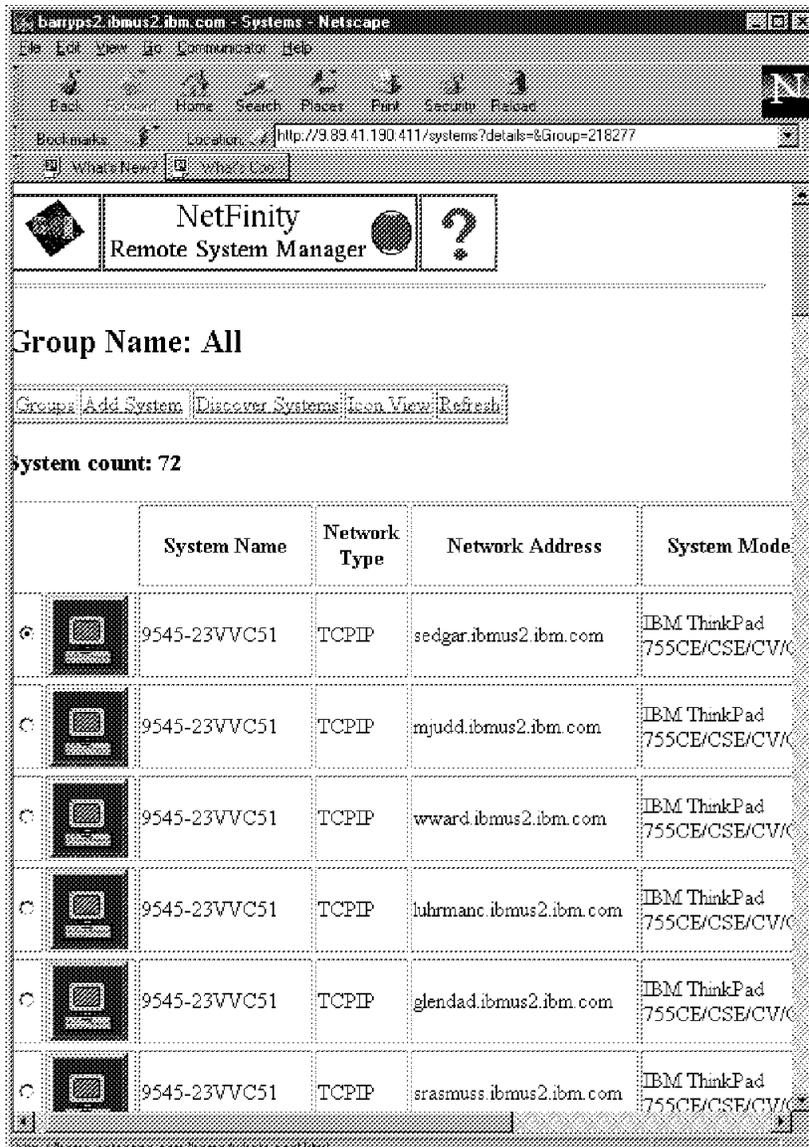
*Figure 91. Remote System Manager - Detail View*

Dynamic addressing can also be set up from the Remote System Manager window:
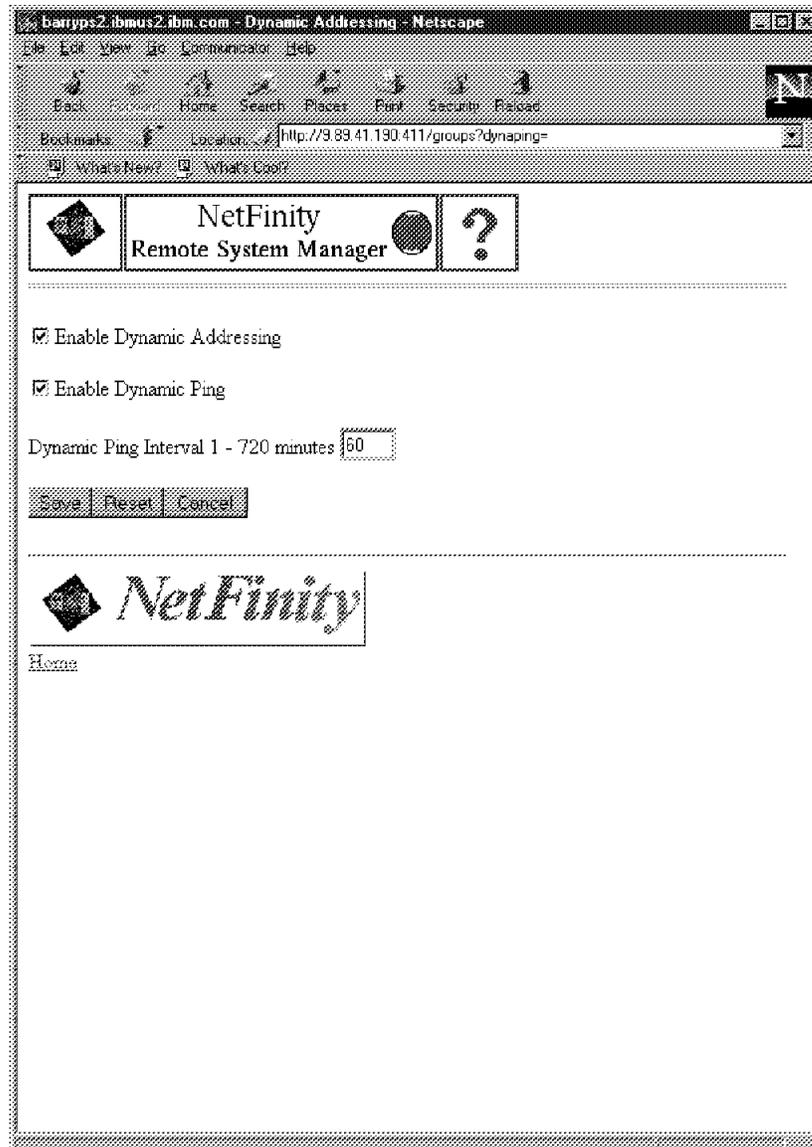
*Figure 92. Remote System Manager - Dynamics*

### 3.2.11  Security Manager

One of the most important features for managing with a Web browser is the
NetFinity 5.0 Security Manager. As we already indicated, the incoming password
of the remote station and the outgoing password of the managing machine *must*
match. There is no way to change the password with the Web browser. You can
add or delete an ID to force a change, but there is no edit or change option for
the passwords.
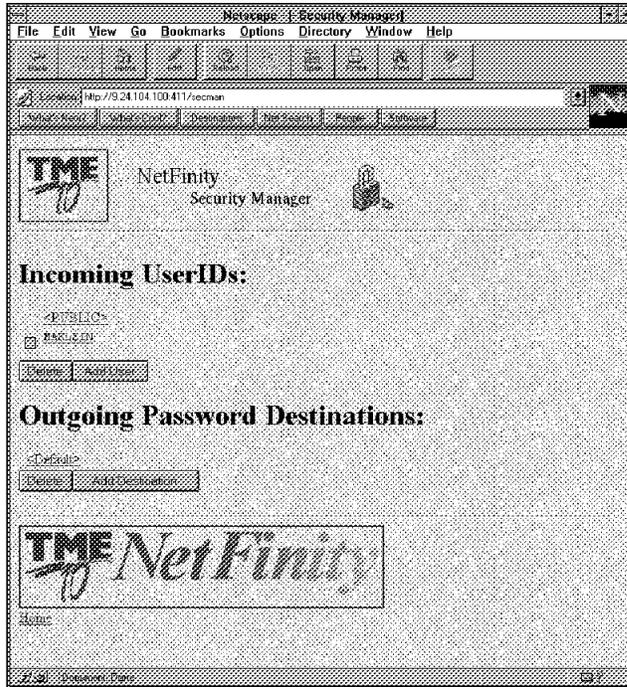
*Figure 93. Security Manager*

For a new incoming password, specify a user ID and a password as well as the services for which you want this user to have access.
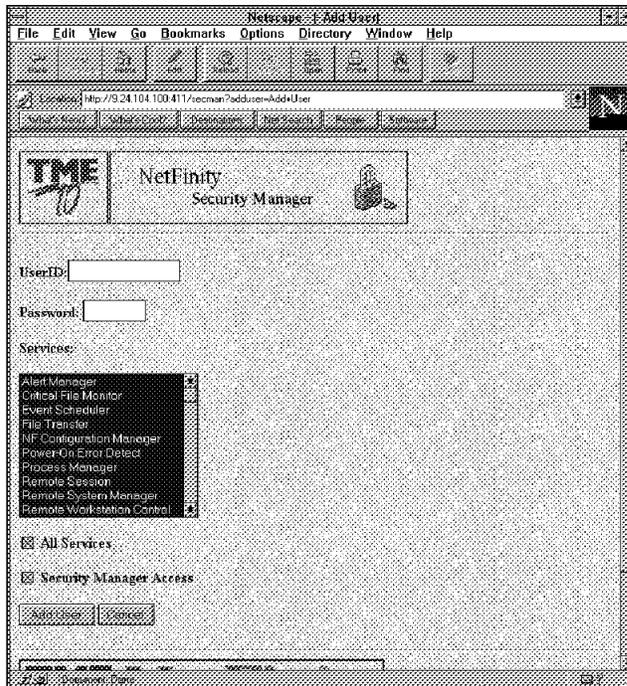


*Figure 94. Security Manager - Add an Incoming Password*

If you define an outgoing password, you have to specify the name of the machine you want to access remotely.
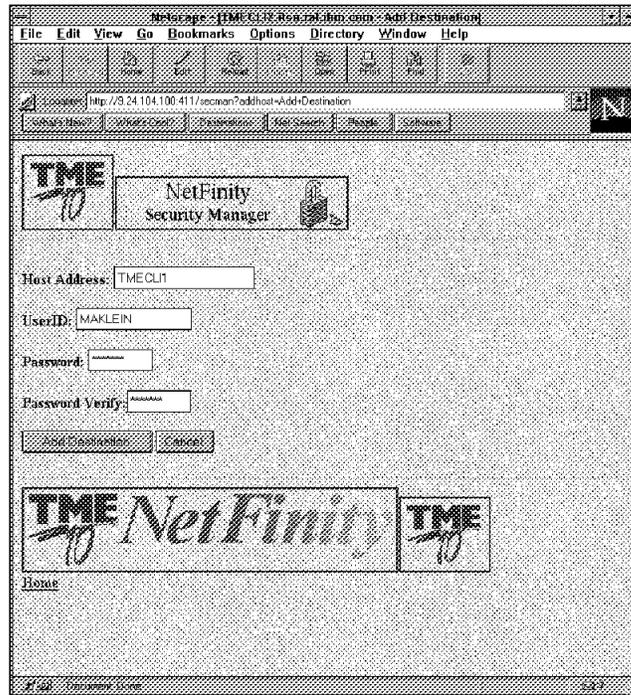
*Figure 95. Security Manager - Add an Outgoing Password*

---

**Important**

Make sure that you do not leave the PUBLIC user ID with access to any services that you want to restrict. Typically, people disable the PUBLIC ID after they have added other user IDs.

---

## 3.2.12 Service Configuration Manager

The Service Configuration Manager is new in NetFinity 5.0 and it also has a Web page. On the main page all of the existing configuration files are listed. In our example, since it was the first time we customized this option, there are no configuration files.
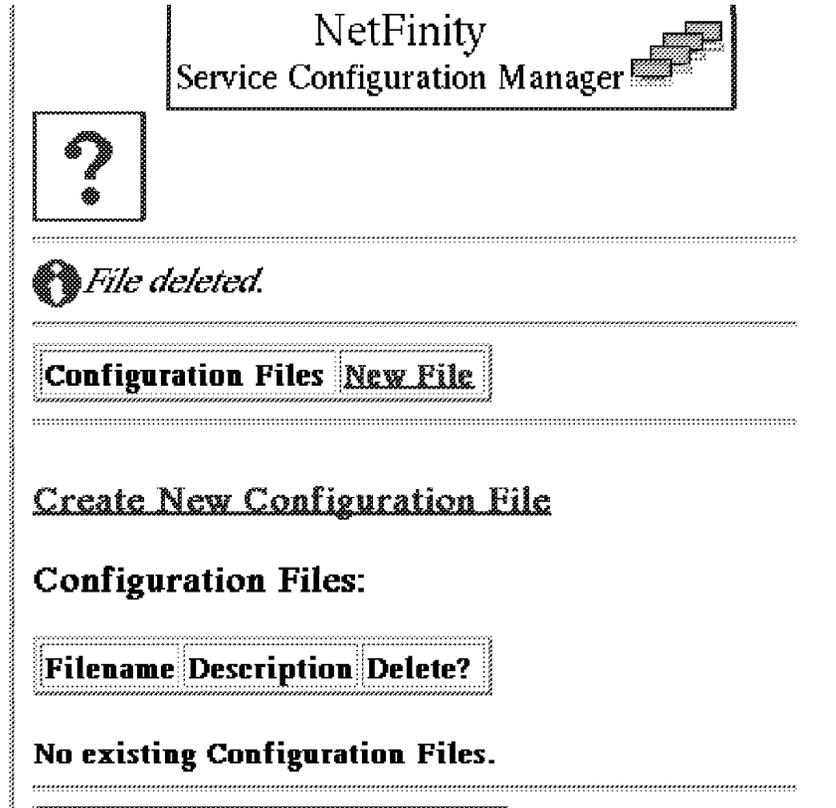
*Figure 96. Service Configuration Manager*

Clicking on **New File** gives us the chance to define a new configuration file. You may get prompted for a user ID and password before you can use a new function. Select the service and the machine from where the configuration shall be taken. All you have to do now is to set the filename and a name for the configuration.

*Figure 97. Service Configuration Manager - New File*

If you have selected to edit the file, the browser brings you to a location where you can edit or delete some of the records from the configuration file. The actual file that gets created will be in the NETFIN\SCF directory.

*Figure 98. Service Configuration Manager - Edit Configuration File (Top)*

## Alert Actions

Notify user with pop-up
Add alert to log file

## Alert Profiles

Service Start Rejected Alerts
Service Start Request Alerts
System Access Denied Alerts
System Offline Alerts
System Online Alerts
System Restart Initiated Alerts
System Restart Rejected Alerts
Database 2: System resource error
Database 2: Resource limit occurred
Database 2: Resource limit occurred during a distributed proce

Delete Selected Records    Reset

Return To Configuration Files

*Figure 99. Service Configuration Manager - Edit Configuration File (Bottom)*

Now we have an entry in the list on the main page.
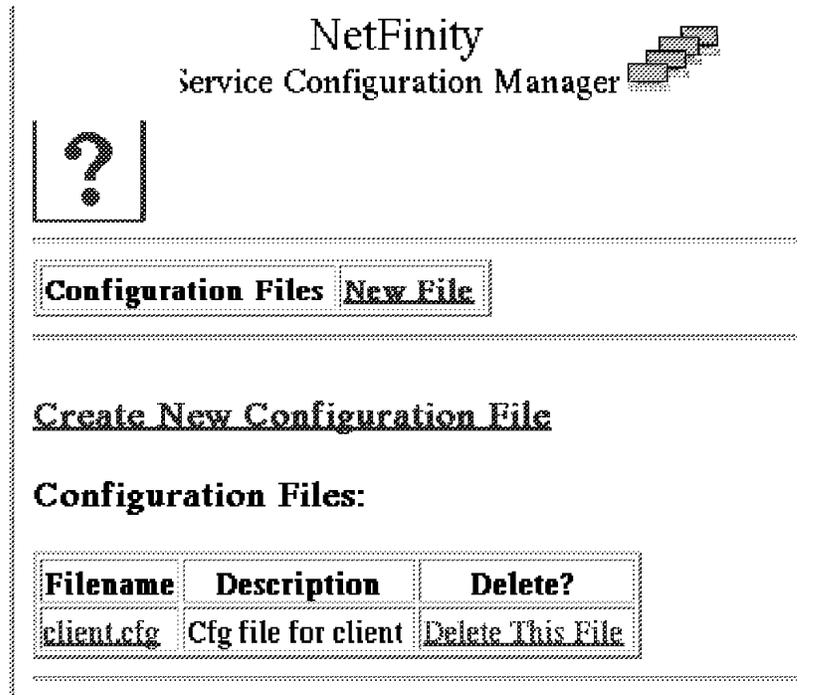
*Figure 100. Service Configuration Manager*

### 3.2.13 Software Inventory

The software inventory on the main page is the equivalent to the software inventory on the GUI. If you click on this icon, the manager tries to figure out what software is installed on the target machine. The display of the output looks like Figure 101 on page 109. Product name, vendor, version and revision and the path are displayed. The manager uses a dictionary to decide which programs to display and which not to show. The name of the directory is default.sid, and it shows up as NetFinity Dictionary in the Web page. If you can create other SID dictionaries, they will show up on this Web page too.

*Figure 101. Software Inventory*

## 3.2.14 System Information

This screen displays hardware and software configuration information about the system. You can scroll down through this information or click on an icon for more details. If you click on an icon, the browser jumps directly to that section. All information about the remote system is on this page.
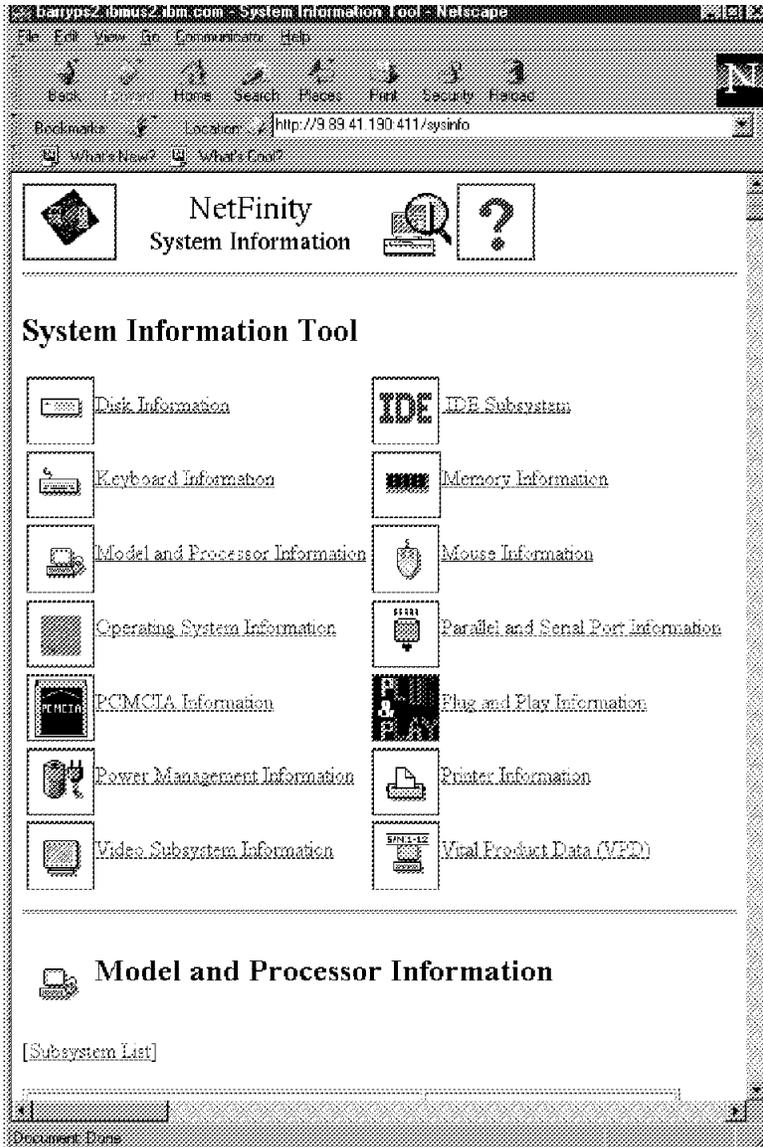
*Figure 102. System Information - High-Level View*

**Model and Processor Information**

[Subsystem List]

| Model Name | IBM PS/2 Thinkpad 755 |
|---|---|
| Machine Type/Model | 9545/L9E |
| Processor | 80486DX4 |
| Processor ID String | |
| Family Number, Model Number, Stepping Level | Family 4, Model 8, Stepping 0 |
| Math Coprocessor | 80387DX4 |
| Number of Processors | 01 |
| Processor Speed | 75 MHz |
| System Board Speed | 25 MHz |
| Expansion Bus Type | ISA or AT bus |
| Internal Processor Cache | Enabled |
| Model | F8 |
| Submodel | 64 |
| BIOS Revision | 18 |
| BIOS ROM Date | 02-15-1996 |
| Dedicated IRQ Levels | 0 1 2 8 9 12 13 14 |
| Shared IRQ Levels | None |

*Figure 103. System Information Tool - Details*

### 3.2.15 System Monitor

The Web page for the system monitor is different from the GUI because it displays all monitors with its actual values when you start the page. In front of each monitor you see a check box.
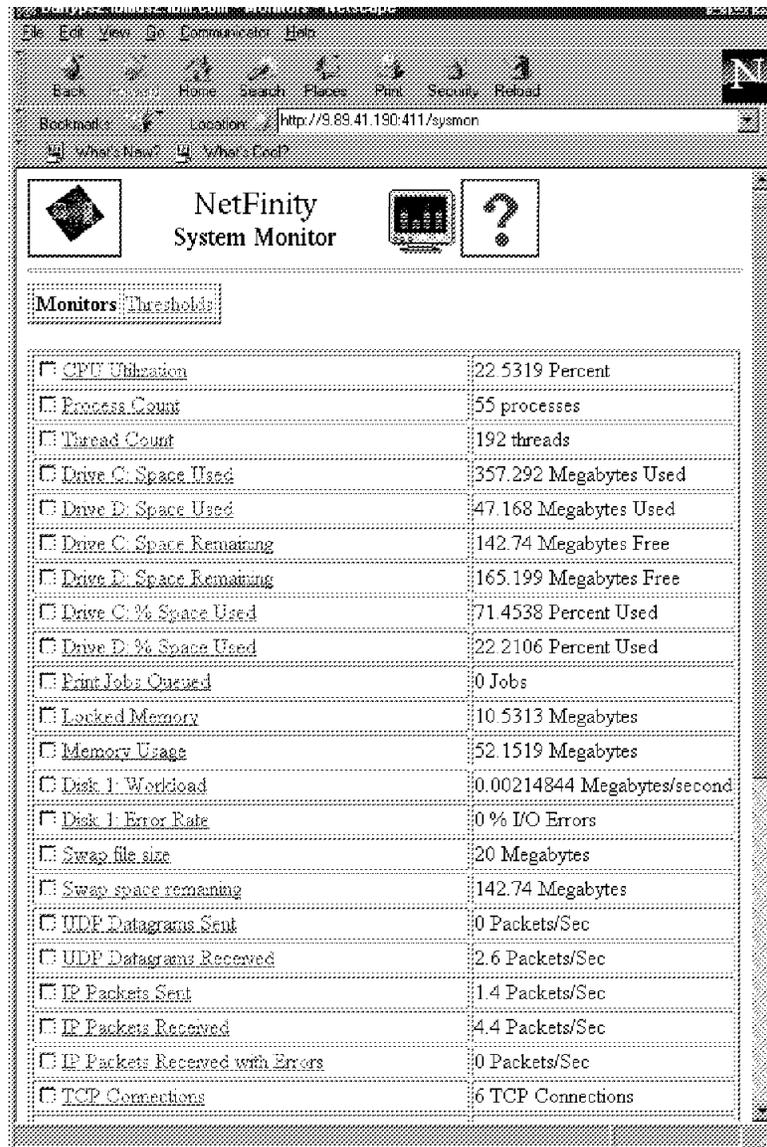
*Figure 104. System Monitor Metrics*

If you check the monitor box, then click on the **Set Refresh** button on the bottom of the page. It will bring up a different page with only those monitors. It will refresh that screen on a regular basis, based upon the value you had entered in the field called Refresh Rate. You can find the entry field for the refresh rate and the button to activate at the bottom of the monitor page.
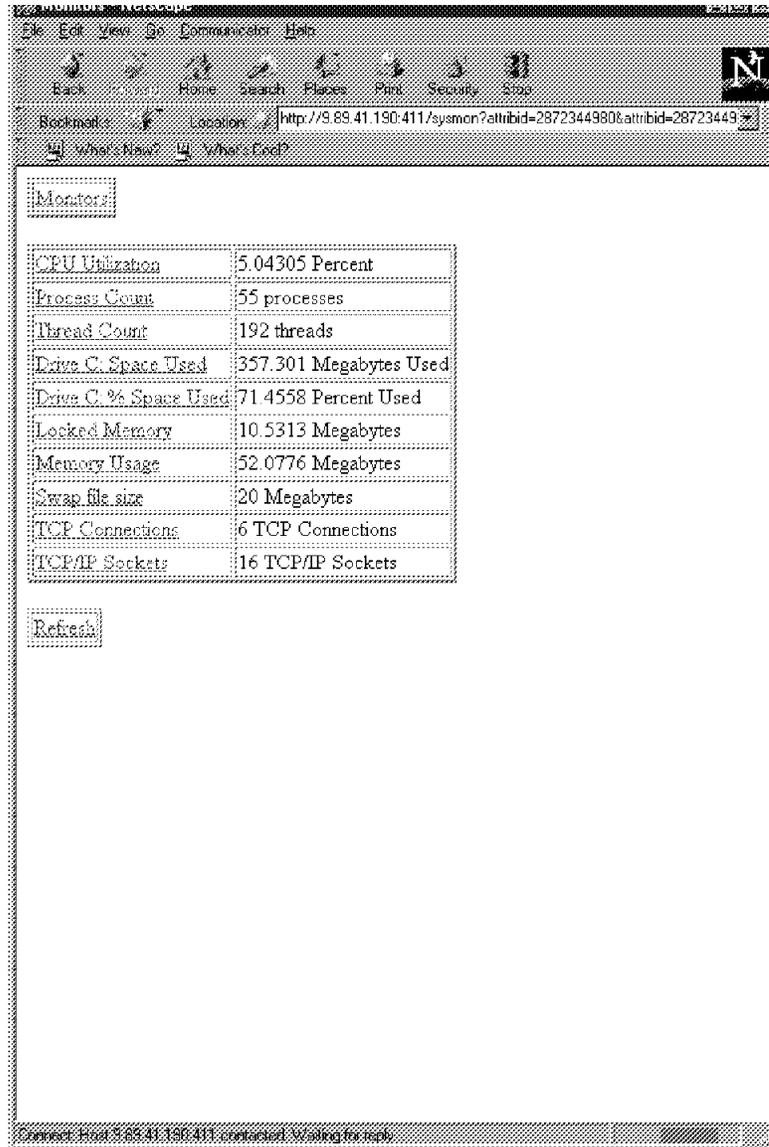
*Figure 105. System Monitor - Selected Monitors*

Clicking on a monitor opens the monitor threshold window where you can define its attributes for alert generation.
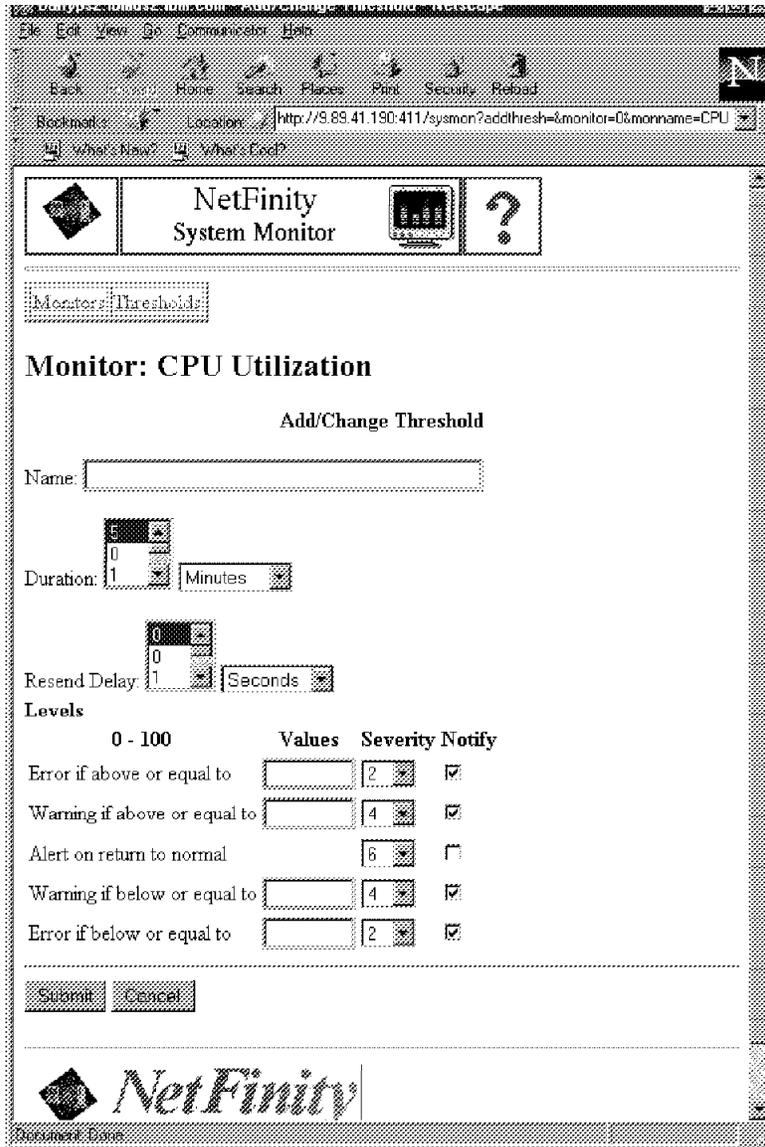
*Figure 106. System Monitor - Threshold Settings*

If you click on **Submit**, the threshold is set in the target machine and you get the next panel which shows all thresholds for this machine. You also get this picture when you click on **Thresholds** on the monitor main page. The check box in front of each threshold is used to select more than one threshold for deletion. As you see in our sample screen the Delete button is at the bottom of the page.
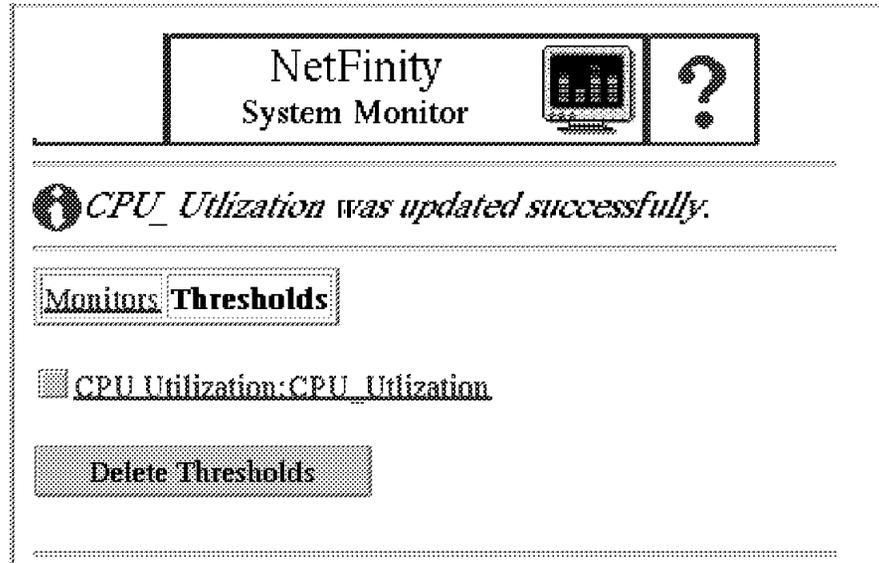
*Figure 107. System Monitor - Threshold Display*

## 3.2.16 System Profile

The system profile page contains system-related data for information such as system type, serial number, display type, user name, location and more. You can also update the information from the Web browser. The next two screens show the system profile main page.
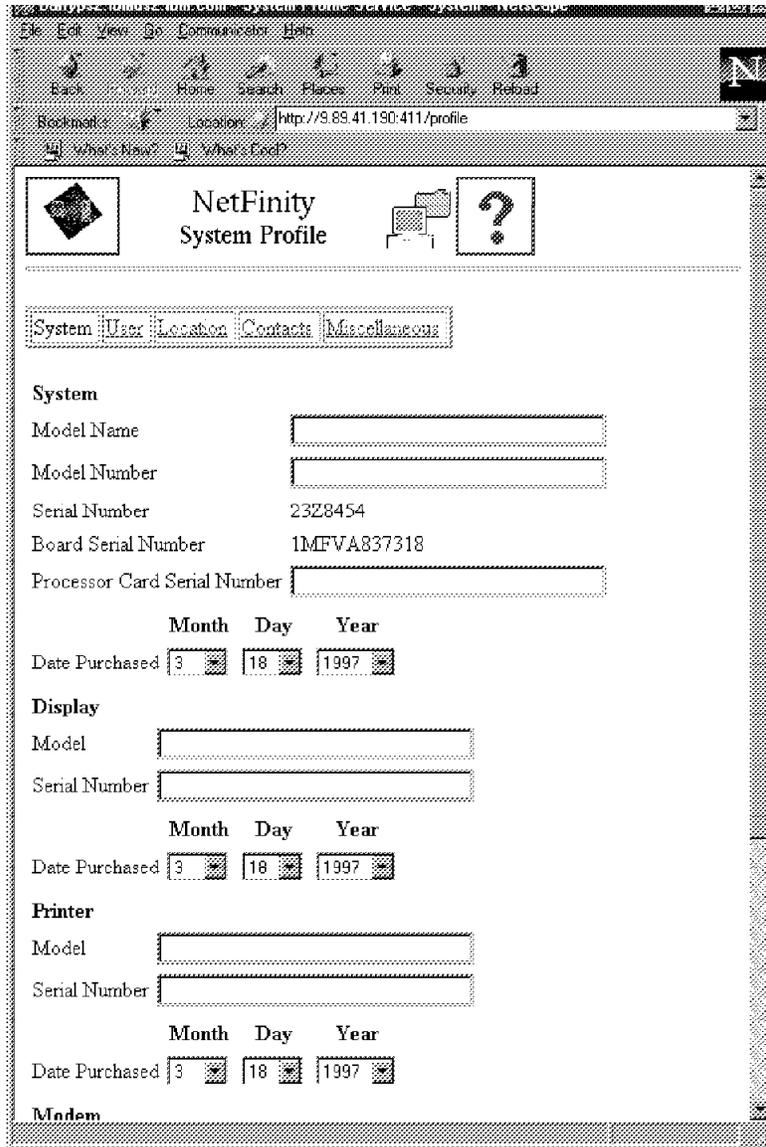
*Figure 108. System Profile (Top)*

If you click on User, Location, Contacts or Miscellaneous, its Web pages are shown. Following is a sample System Profile page:
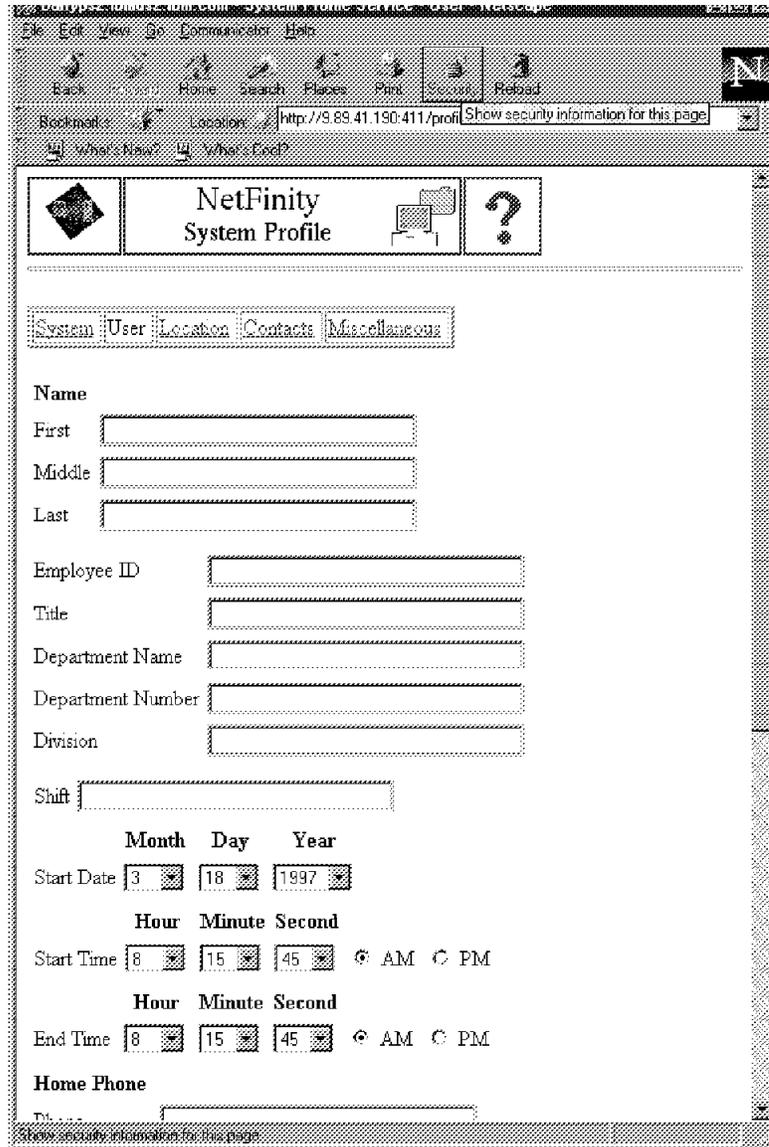
*Figure 109. System Profile - User Page*

## 3.3 Sample Scenario for the Use of Webability

Whenever the free disk space on the monitored system is less than a specific value an alert will be generated. The alert will cause the start up of a Web browser with the web page for the Alert Manager already loaded. In order to help users who are unfamiliar with the Webability function we show a scenario using the regular NetFinity 5.0 GUI, followed by the same scenario using the Webability function.

## 3.3.1 Step 1: Define a Threshold

The first thing we did was define a threshold in the Systems Monitor using the Remaining Disk Space monitor.

### 3.3.1.1 NetFinity 5.0 GUI

Start the System Monitor service by double-clicking on its icon in the NetFinity 5.0 main window. Select **Windows** and **Show Monitors**.
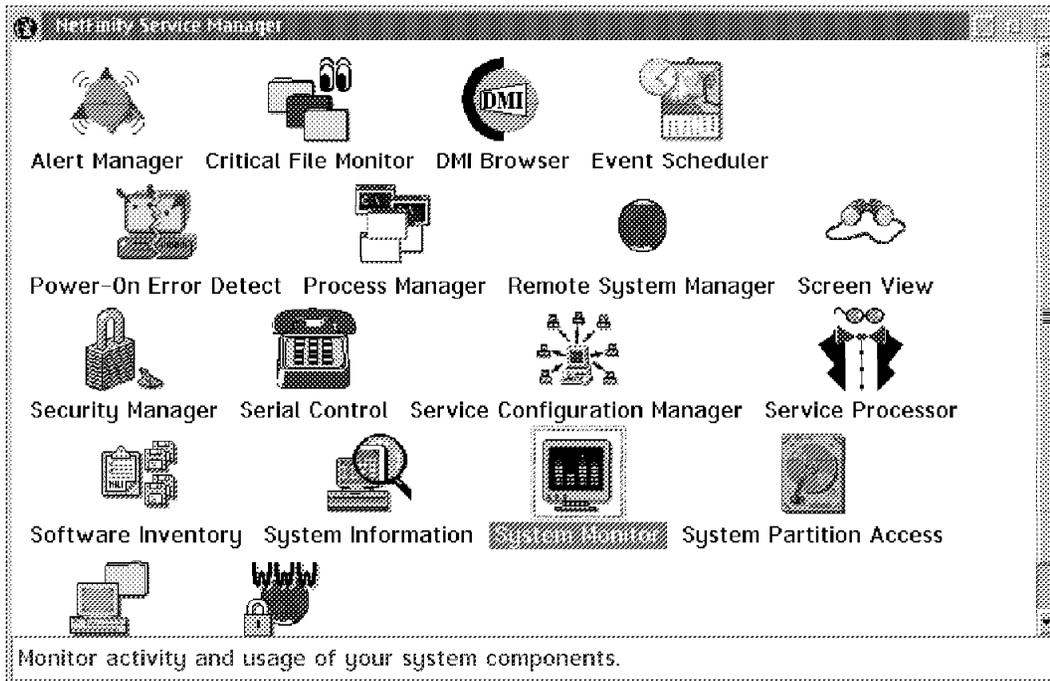


*Figure 110. NetFinity 5.0 Main Window*

Now select **Drive C: Space Remaining** and **Accept**.



*Figure 111. Selecting the Monitor*

You should see the monitor on the screen similar to the one in Figure 112 on page 119. To define the desired threshold we can double-click on the monitor window or click the right mouse button and select **Open -> Threshold**.
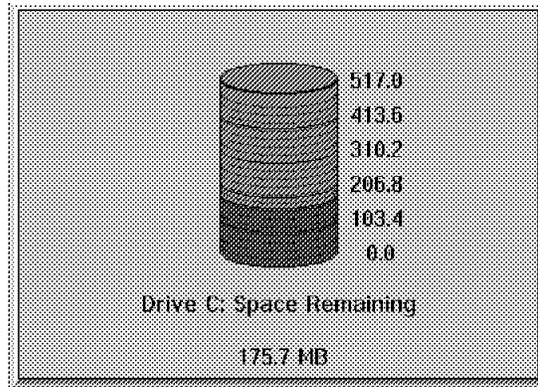
*Figure  112.  Remaining Disk Space Monitor*

This opens the threshold definition screen for the monitor as shown in
Figure  113.  The first thing to do is to give the threshold a name.  This name will
then appear in messages later on when the alert is generated.  It can also be
used in the Service Configuration Manager when you distribute predefined
profiles.
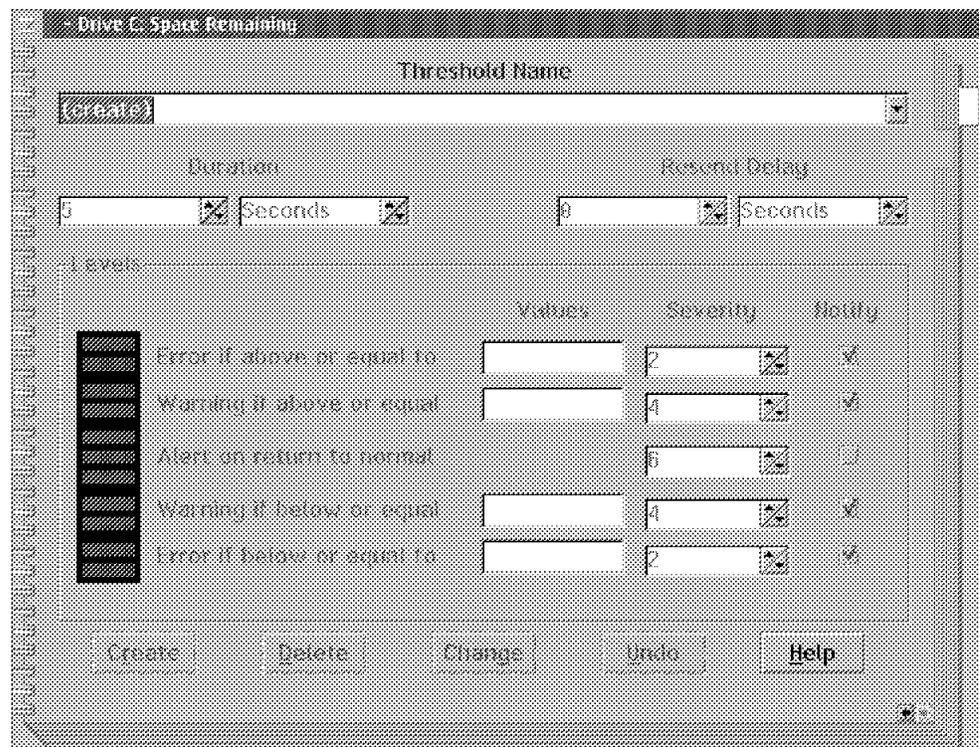


*Figure  113.  Threshold Window*

You can also set a value for how long the error condition must exist before an
alert will be generated and how long NetFinity 5.0 should wait before sending a
new alert.  In the lower-half of the window the actual values for the thresholds
and the severity these alerts will be given can be defined.  For our scenario the
settings are as follows.
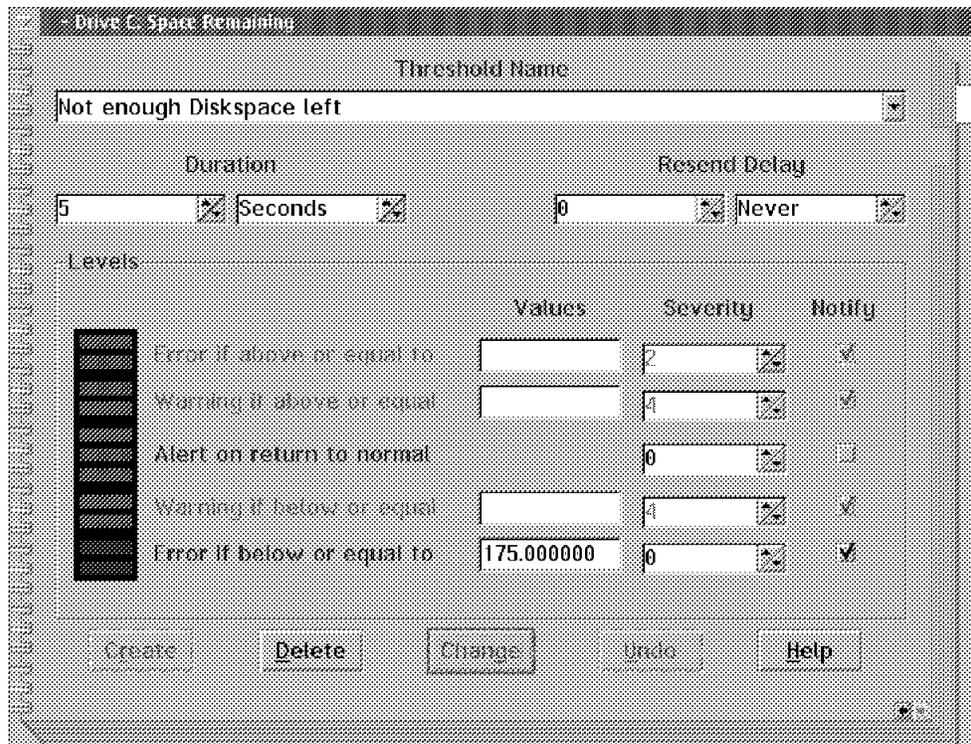
*Figure 114. New Threshold Defined*

If you want to perform the same actions using the Webability service, you can accomplish the task with a few hyperlinks:

- Click on the System Monitor hyperlink on the main Web page

- Click on **Drive C: Space Remaining**

- Tailor the values and click on **Submit**

## 3.3.2 Step 2: Setting Up the Alert in the Alert Manager

Setting up a threshold does not create a pop-up window in Netscape, so we have to define a new alert action that executes a remote command on the managing system to start Netscape (of another browser) and load the Alert Log page.

### 3.3.2.1 NetFinity 5.0 GUI

Enter the NetFinity 5.0 Alert Manager by double-clicking on its icon.
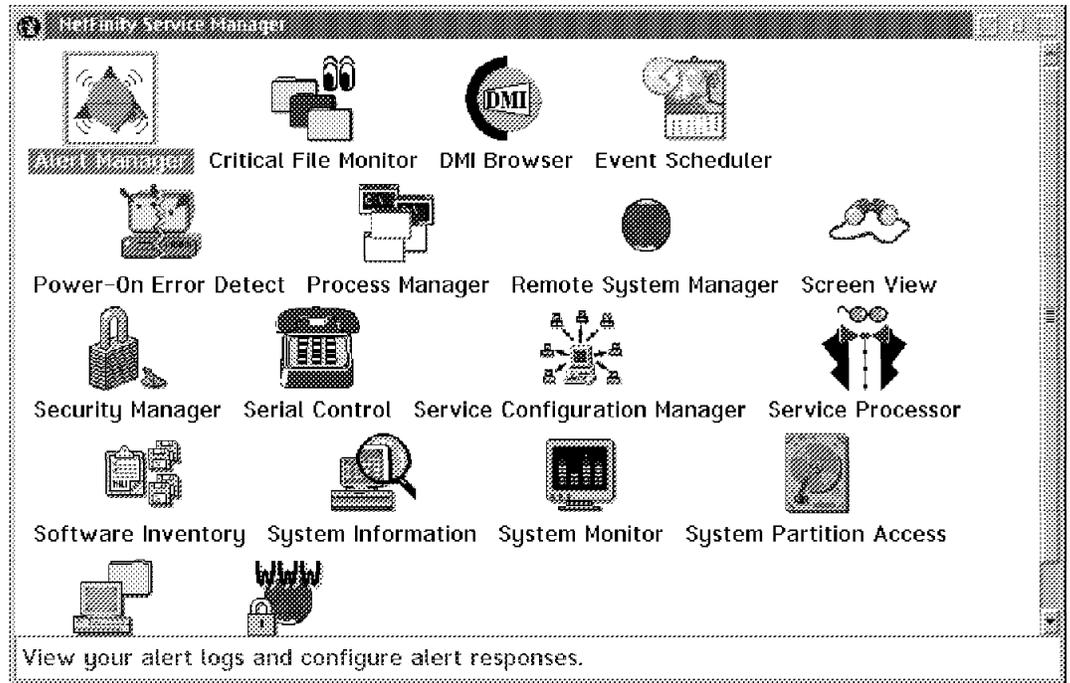
*Figure 115. Start Alert Manager*

From the Alert Log window select **Actions**.
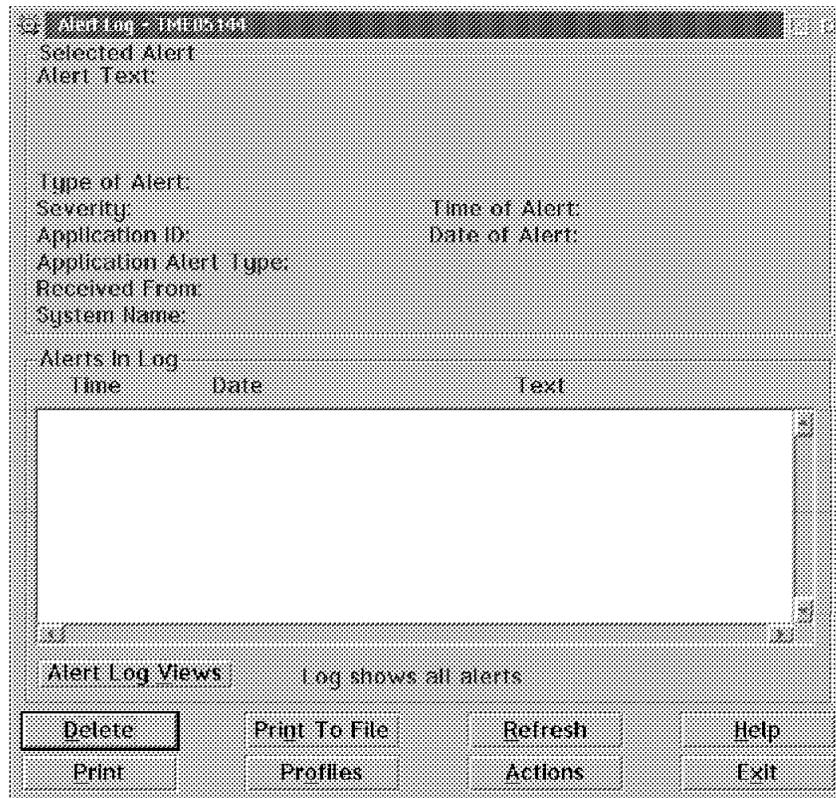


*Figure 116. Alert Log Main Window*

In the Alert Actions window click on **New**.
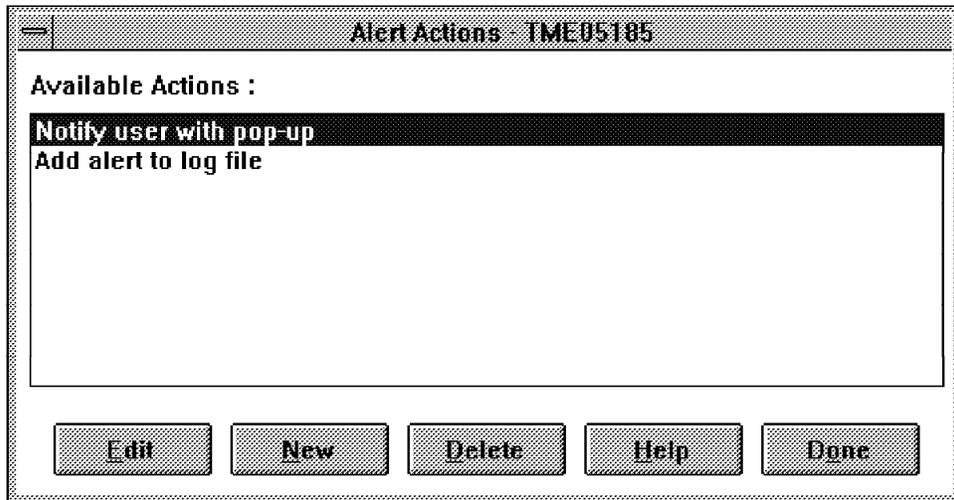
*Figure 117. Alert Actions Overview*

This brings you into the Action Editor, where you can define which actions are taken, when certain types of alerts occur. For our scenario the alert type we need is Error, the severities we chose in the threshold window were 2 and 4, and the Application that will send the alert is MonitorB. We did not specify any Application Alert Type or specific Sender ID, but that can be done as well. The action that should be taken when this alert occurs is to **Execute command** '<**P1**>' in the Action Definition Part of the window. The command that is executed is put into a batch file, because it did not fit into the given box. You can see the exact command in Figure 120 on page 123.
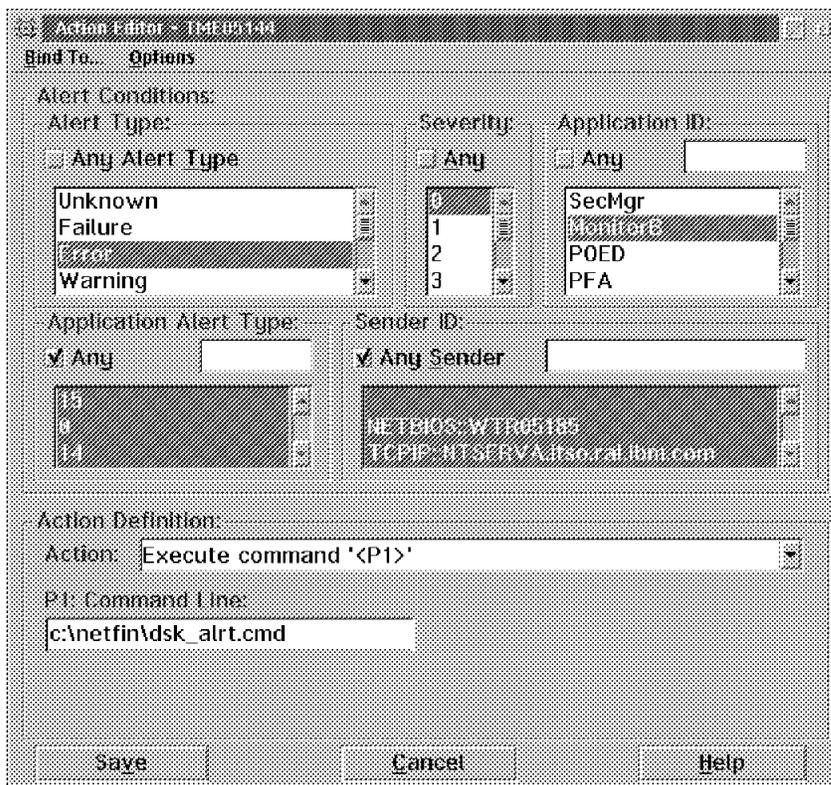


*Figure 118. Definition of a New Alert Action*

Click on **Save** and you get back into the Alert Action window.
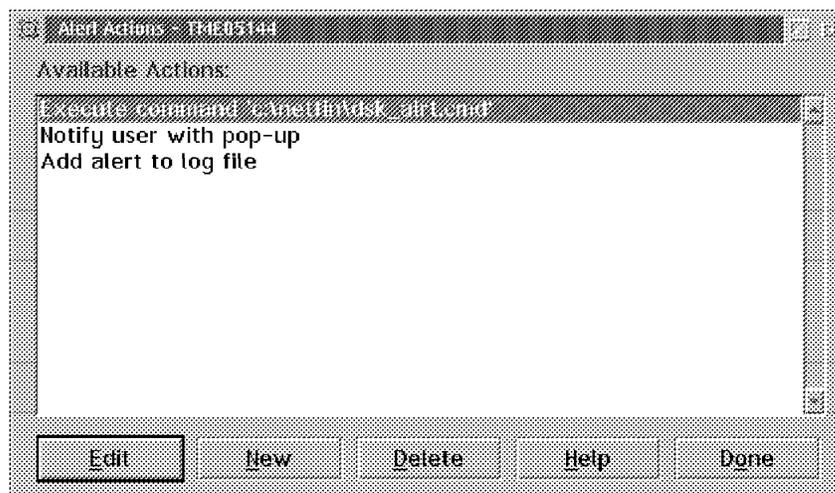


*Figure  119.  Alert Actions Window with New Alert*

After that the definition of the alert is finished.

### 3.3.2.2  Webability
Using the Webability service you follow the same basic procedure.  Start the
Alert Manager by clicking on its icon on the main Webability page.  On the Alert
Manager page select the alert action we want to use (that is, **Execute command**
′ < **P 1 >** ′).

Defining the conditions for the alert action is the same as in the GUI.  Define
type, severity, application ID, application alert type, sender ID and fill in any
parameters.  Click on **Save Changes** and the action is set up via the Webability
interface.

## 3.3.3  Step 3: Definition of DSK_ALRT.CMD and Test
Create a file x:\netfin\dsk_alrt.cmd which contains the following NetFinity 5.0
command.
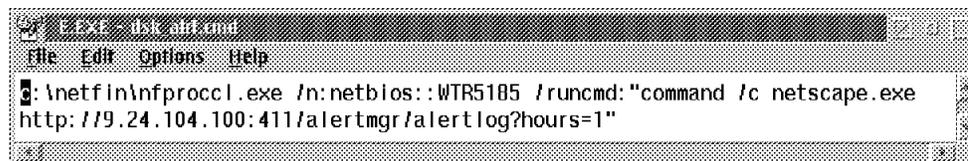


*Figure  120.  Command to Be Executed*

All you need to do is save that file, and you are ready to try it.  NetFinity 5.0
comes with a command called genalert, which is used to generate user alerts.
This can be done manually from the command line or from an application as part
of a script.  There is no GUI interface for genalert, but using the Webabilty
interface you can fill in a form.  That helps guide you through the possible
parameters.
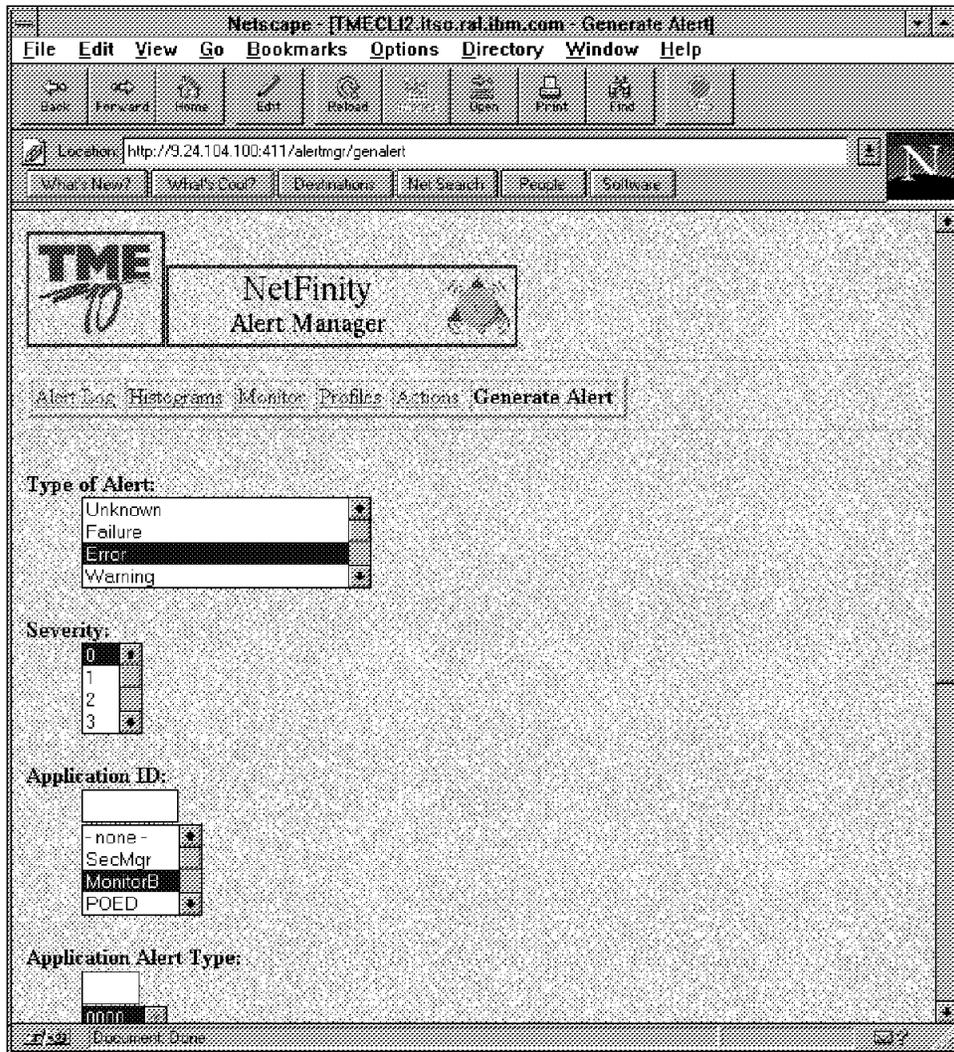
*Figure 121. Generate Alert (Top)*
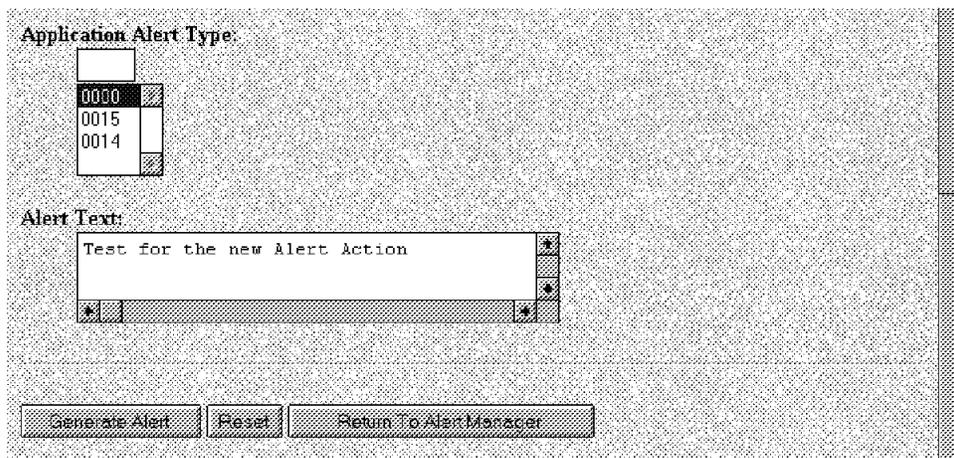


*Figure 122. Generate Alert (Bottom)*

If the alert was generated, you see the following message in the Webability Generate Alert page.
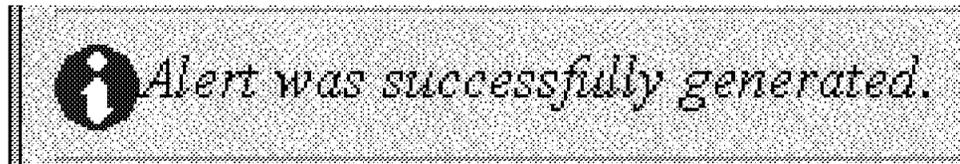
*Figure 123. Success*

If you have not removed the Notify User with Popup alert action, you get the Alert Received window on the machine, where the alert was generated.
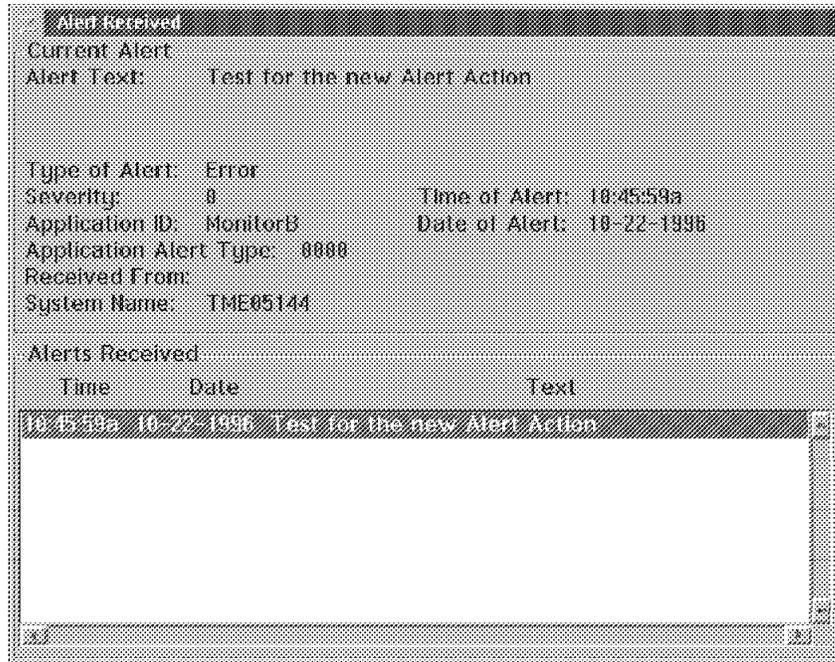


*Figure 124. Alert Received Pop-Up Window*

If you have removed all rights from the <PUBLIC> user, then the Web browser on the managing machine automatically pops up with a logon window.
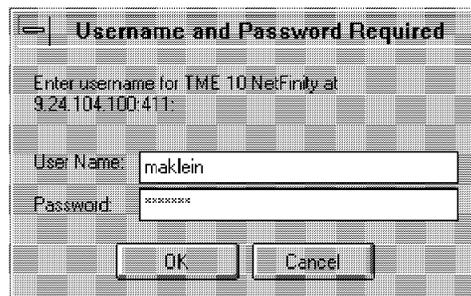


*Figure 125. Logon with Incoming User ID and Password*

This brings you to the Alert Log page.
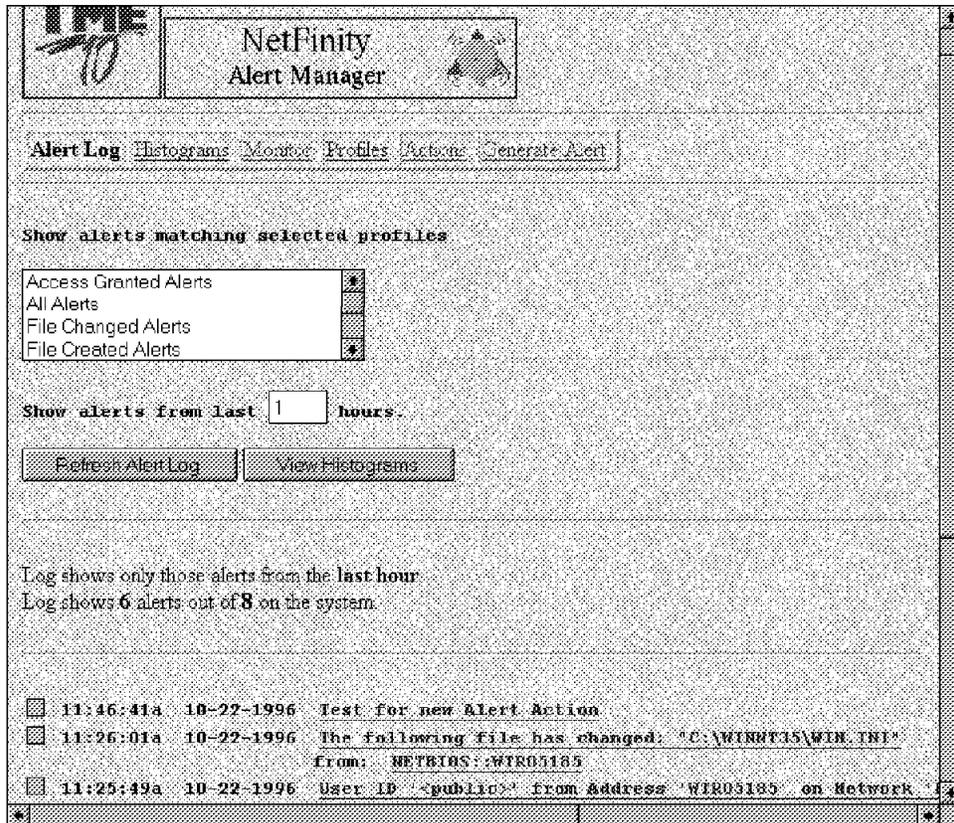
*Figure 126. Alert Log with New Alert on Top of the List*

Anytime the threshold for free disk space on drive C: is reached, your Web browser should start on the managing machine.

# Chapter 4. Command Line Automation

This chapter shows automation examples using the OS/2 platform. It makes extensive use of the new command line interface in NetFinity V5.0. The scenarios that are covered in this chapter have to do with the following:

- Security settings
- Alert manager changing thresholds
- Add/change critical file monitor thresholds
- Process Manager alerts management
- System monitor threshold setting

The sample REXX programs do not check for all possible errors, nor do they cover all the functions. They only provide a framework to get started on using the command line interface.

## 4.1 Environment Description

The environment of these scenarios is as follows:

- OS/2 Warp Connect with NetFinity 5.0 Manager and DB2
- Windows 95 on a ThinkPad
- Windows NT V3.51 on an IBM PC 350

## 4.2 Check Security on Managed Machines

The purpose of this chapter is to describe how you can check if the security settings on the targets are correct or if the machines have no security on them. When you initially install NetFinity 5.0 the default security for public is such that anyone can use it. If the machines get left in their default setting with no security on it, we want to change the setting so that no unauthorized access will be possible. As you can see in Figure 127 on page 128 the default for the <Public> user is all services are enabled as well as access to the Security Manager.
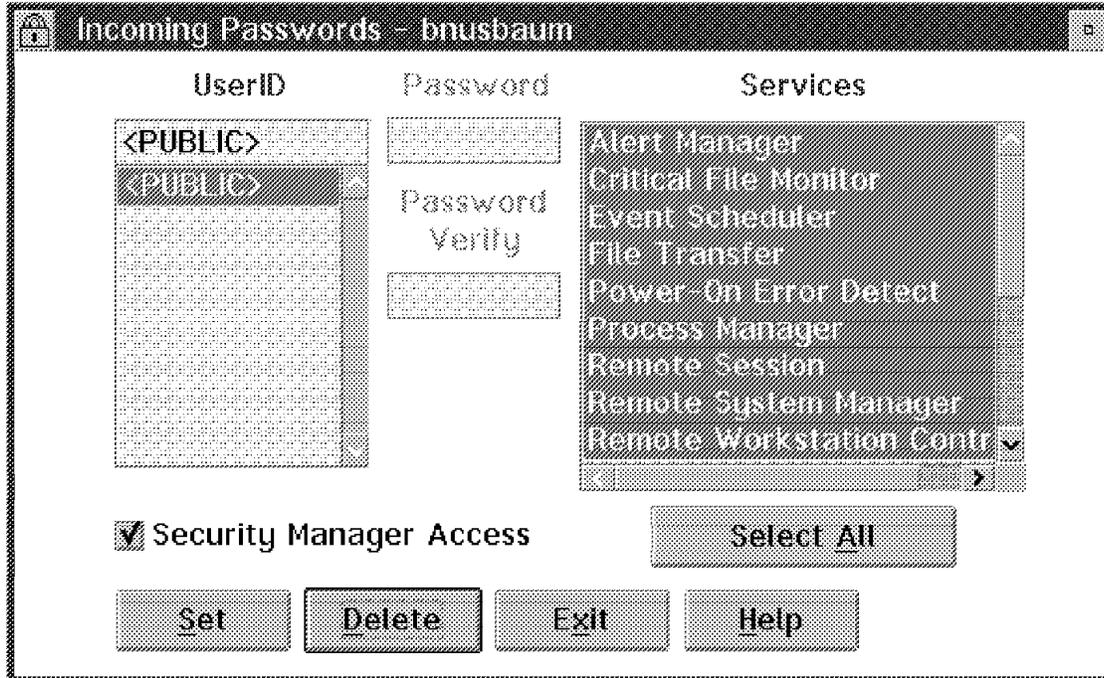
*Figure 127. Incoming Passwords*

The sample program consists of three parts. Part one is the main program
which checks the security and modifies it if necessary.

Part two is an external procedure named GETGROUP. This procedure needs no
parameters. When it is called it displays a list of all defined groups and asks for
the group with which you want to work. It returns the groupname to the calling
procedure. Following is a GUI picture of all defined groups on the manager
machine.
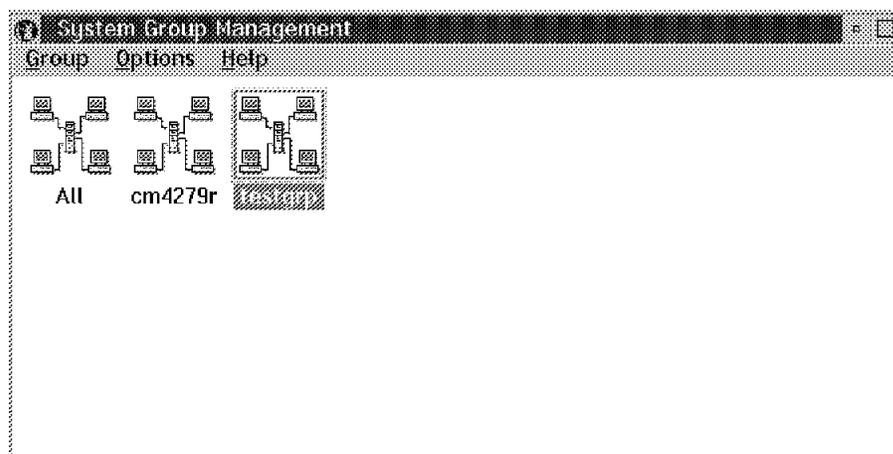


*Figure 128. System Group Management*

For an example of the command line output see 4.2.1.1, "Command Output" on
page 129.

Part three is an external procedure named GETSYS. As the name implies this procedure returns system information. Two parameters are needed when calling this procedure. The first parameter is the groupname and the second is either 1 or 0. You need to specify 1 when you call this procedure the first time. That initializes the routine and places a list of all of the systems into a queue. Specify a value of 0 when you want to get the data for the rest of the systems. The procedure returns a string with the system data or a null string if no more systems are available. The correct syntax is:

```
CALL GETSYS TESTGRP 1
```

The output looks like:

```
03BE86DD "tmecli1" NETBIOS WTR05095 TRUE OS2
```

You normally can't see the output because it is sent directly to the calling procedure.

## 4.2.1 GETGROUP

As mentioned before this procedure requires no parameters but there must be a queue defined called NFQUEUE. In our programs this is done in the main program SETSECU.CMD. We read the groups with the NFRSYSCL /GETGRP command into the NFQUEUE. The syntax is:

```
@NFRSYSCL /GETGRP /ALL 2>nul " rxqueue NFQUEUE
```

The at symbol (@) suppresses the output of the command to the console. The 2>nul part routes the standard error message to a null variable (a trashbin) and the pipe (|) routes the standard output to the function rxqueue. This function puts the data into the queue NFQUEUE. After doing that, the grouptag and the name are read from the queue and put into a stem variable called *group*. We do a loop as often as groups are available in the queue and display each groupname with the index (loopcount). The user enters a number from 1 to the number of groups and the procedure returns the name of the group to the caller. If the number entered is not in the list, the procedure returns a null string and the program stops.

### 4.2.1.1 Command Output
Below you see the output from the command line. We need to parse this information to work with it.

```
[C:\]nfrsyscl /getgrp /all
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ GRPTAG=0x00020022, GRPNAME="All", COMBO=ANY,
  KWD={  },
  OS_MASK={  }, AUTODISC=60,
  PROTO_MASK={  },
  DEFONLN=DISABLED, DEFOFFLN=DISABLED, DEFPING=NONE
}
{ GRPTAG=0x001504F9, GRPNAME="cm4279r", COMBO=ANY,
  KWD={ "cm4279r", "CM4279R" },
  OS_MASK={  }, AUTODISC=DISABLED,
  PROTO_MASK={  },
  DEFONLN=NONE, DEFOFFLN=NONE, DEFPING=30
}
{ GRPTAG=0x1903303D, GRPNAME="testgrp", COMBO=ANY,
  KWD={ "testgrp", "TESTGRP" },
  OS_MASK={  }, AUTODISC=60,
  PROTO_MASK={  },
  DEFONLN=NONE, DEFOFFLN=NONE, DEFPING=NONE
}

[C:\]
```

## 4.2.1.2 GETGROUP.CMD
Following is the sample program to get the group information:

```
/* Get all defined Groups */
'@NFRSYSCL /GETGRP /ALL 2>nul ] rxqueue NFQUEUE'          /* get the groups */

/* Get the Groupnames from the Q */
row=0
do while queued()>0
   parse pull t1 t2 t3 t4 rest
   if pos("GRPTAG",t2) > 0 then do                        /* check corr.tag */
      row=row+1                                           /* put into array */
      GROUP.row.1="GRPTAG:"]]substr(t2,10,length(t2)-10)
      GROUP.row.2=substr(t3,10,length(t3)-11)
      end
end /* do */

NoOfGroups = row                                          /* keep no of groups */
say 'Select one of the following groups'
do i = 1 to row                                           /* display gr.names */
   say i' -> 'group.i.2
end /* do */
say ''
rc=charout(,'1 - 'row' -> ')                              /* select the group */
selgrp=linein()
if selgrp < 1 ] selgrp > row then do
   say 'Your selection is not in the list'
   say 'Program ends'
   return ""
end

return group.selgrp.2
```

## 4.2.2  GETSYS

The GETSYS.CMD program gets all of the relevant data for each system in one call and returns the data as a string.  It needs two parameters: group name and an init value:

CALL GETSYS TESTGRP 1

The group name defines the group on which to work.  The init parameter is to tell the procedure to either get the systems into the queue or to work on the existing data in the queue.  At the beginning we define a new queue in which to put the data.  The next step discovers all systems in that group.  REXX is able to handle more than one queue but you can only work with one queue at a time. That's why we must activate the right queue to read data from it. RXqueue("SET",...) does that.  Then we get the data into the system queue with the NFRSYSCL /GETSYS command.  Since the command output is in more than one line we have to continually loop through it until we are done.  This loop runs until it finds a } in the first token t1.  The data is read into variables and returned as a string to the caller who can parse this data again into variables.  We return the SYSTAG SYSNAME SYSPROTO SYSADDR SYSONLINE SYSOS values.

03BE86DD "tmecli1" NETBIOS WTR05095 TRUE OS2

If you need additional information from this procedure, just modify the select statement for your needs and add the variable to the return string.  Don't forget to modify the calling procedure in order to parse the additional variables.  See 4.2.2.2, "GETSYS.CMD" on page 132 for the select statement and 4.2.3, "SETSECU.CMD" on page 134 for the parse statement.

### 4.2.2.1 Command Output

The NFRSYSCL /GETSYS command returns several lines for each protocol on each system. Here is a sample for just one system.

```
[C:\]nfrsyscl /getsys /sys&colontme05188
NetFinity Remote System Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ SYSTAG=0x0003B6DA, SYSNAME="tme05188", PROTO=TCPIP,
  ADDR="NTSERVA.itso.ral.ibm.com",
  ONLINE=TRUE, ONLN=DISABLED, OFFLN=DISABLED, PING=30,
  GRPLIST={ 0x00020022, 0x001504F9 },
  ERRORCOND={  },
  OS=OS2, OS_VER=3.00, MANAGER, SERVER, MAC=08005ACF1BA3
}
{ SYSTAG=0x0039C750, SYSNAME="tme05188", PROTO=NETBIOS,
  ADDR="WTR05188",
  ONLINE=TRUE, ONLN=DISABLED, OFFLN=DISABLED, PING=30,
  GRPLIST={ 0x00020022, 0x001504F9 },
  ERRORCOND={  },
  OS=OS2, OS_VER=3.00, MANAGER, SERVER, MAC=08005ACF1BA3
}

[C:\]
```

### 4.2.2.2 GETSYS.CMD

The getsys sample follows:

```
/* get systems in group */
/* trace ?r */
parse arg selgrp init

if init then do
   /* define a systen q for data exchange */
   QName = RXQUEUE("create", "SYSQUEUE")                    /* system q */
      if QName \= "SYSQUEUE" then do
          call RXQUEUE "DELETE", "SYSQUEUE"
          call RXQUEUE "DELETE", QName
          call RXQUEUE "create", "SYSQUEUE"
          end
   rc='@NFRSYSCL /DODISC /GRP:'selgrp' 2>nul'               /* discover systems */
   QName = RXqueue("SET", "SYSQUEUE")                       /* make SYS q active */
   '@NFRSYSCL /GETSYS /GRP:'selgrp,
   ' 2>nul ] rxqueue SYSQUEUE' /* put sys in q */
   end
else
   QName = RXqueue("SET", "SYSQUEUE")                       /* make SYS q active */
if queued() > 0 then do                                     /* put systems in */
   t1=""
   do while t1<>"}"
      parse pull t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 .           /* sys. stem */
      select
      when t1="{" then do
         systag=substr(t2,10,8)                             /* systag */
         sysname=substr(t3,9,length(t3)-9)                  /* sysname */
         sysproto=substr(t4,7,length(t4)-7)                 /* protocol */
         end
      when pos("ADDR=",t1)<>0 then do
         sysaddr=substr(t1,7,length(t1)-8)                  /* address */
         end
      when pos("ONLINE=",t1)<>0 then do
         sysonline=substr(t1,8,length(t1)-8)                /* online */
         end
      when pos("GRPLIST=",t1)<>0 then do                    /* grouplist line */
         nop
         end
      when pos("ERRORCOND=",t1)<>0 then do                  /* error line */
         nop
         end
      when pos("OS=",t1)<>0 then do                         /* os line */
         sysos=substr(t1,4,length(t1)-4)
         end
      otherwise do
         nop
         end
      end /* select */
   end /* do */
   return systag sysname sysproto sysaddr sysonline sysos
end
else
   return ""
```

## 4.2.3 SETSECU.CMD

The main program defines a log file and sets a variable called no_success. This is used at the end to see if some of the actions did not complete successfully or if some systems were offline. The NFQUEUE we define is the default queue for all actions. Please remember that if you want to read from a different queue you have to make it the new default. We store our default user ID for later use in a variable called DUID and the default password in DPW. Therefore, we have to define a default user ID with the GUI. This is done with the Security Manager - Edit/Display Outgoing Passwords. Here is a sample.
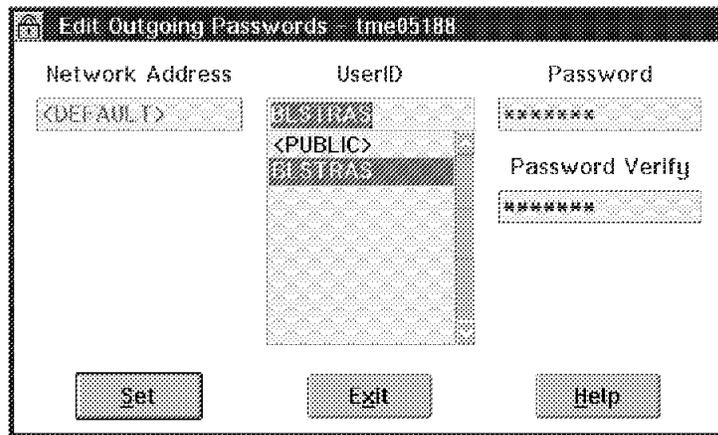


*Figure 129. Edit Outgoing Password*

To get this information using the command line interface we use the NFSECCL /LISTOUT command. Then we call GETGROUP to see what groups we can work on. GETSYS is the next action. Because it's the first time we call this procedure we set init=1. We will have to loop through the data at this point. This loop runs as often as the GETSYS returns data. Within the loop we parse the variable that contains the system data. Then we point to the correct queue and if the online variable contains the value TRUE, we call NFSECCL to get all user IDs and passwords. Another loop runs as long as user IDs are available in the queue. If the user ID is <Public>, the password is PWD="*" and the SECMGR is active. We add a new user ID. This user ID has the same ID and password we set at the beginning of the program stored in DUID and DPW. Changing the default <Public> user so it has no right at all is done using the NFSECCL /EDITIN command. We write a log record to our log file to document what has happened. As pointed out before these steps are executed for as long as there is data in the variable returned from the GETSYS procedure. The last action is to close the log file. Below you see some sample log file entries for setsecu.cmd.

```
16 Oct 1996 17:48:02 USER:"BLSTRAS"  PWD:"CM4279R" added on SYSTEM "tme05188" NETBIOS WTR05188
16 Oct 1996 17:48:03 USER <PUBLIC>  changed on SYSTEM "tme05188" NETBIOS WTR05188
16 Oct 1996 17:48:03 System:"tmecli1" via:TCPIP addr:8235tr04.itso.ral.ibm.com is offline
16 Oct 1996 17:48:04 USER:"BLSTRAS"  PWD:"CM4279R" added on SYSTEM "tmecli1" TCPIP TMECLI1.itso.ral.ibm.com
16 Oct 1996 17:48:05 USER <PUBLIC>  changed on SYSTEM "tmecli1" TCPIP TMECLI1.itso.ral.ibm.com
```

*Figure 130. Setsecu Log File*

The program is started simply by entering setsecu at the C:\ prompt.

```
Select one of the following groups
1 -> All
2 -> cm4279r
3 -> testgrp

1 - 3 -> 3
Error occurred or some Systems are offline. See log file

[D:\ITSO]
```

### 4.2.3.1  SETSECU.CMD

The following is the listing for the program we described above.

```
/**/
'CLS'
logfile="NETFINCL.LOG"
no_success=0
QName = RXQUEUE("create", "NFQUEUE")                        /* define q */
   if QName \= "NFQUEUE" then do
       call RXQUEUE "DELETE", "NFQUEUE"
       call RXQUEUE "DELETE", QName
       call RXQUEUE "create", "NFQUEUE"
       end
QName = RXqueue("SET", "NFQUEUE")                           /* make q active */


/* get own uid and pw for later use */
'@NFSECCL /LISTOUT /ALL 2>nul ] rxqueue NFQUEUE'           /* write data to q */
do while queued()>0                                        /* read data from q */
   parse pull t1 t2 t3 t4 t5
   if pos("DEFAULT",t2)<>0 then do                         /* search default host */
      DUID=substr(t3,8,length(t3)-8)                       /* default user */
      DPW=substr(t4,5,length(t4)-4)                        /* default PW */
   end
end

call getgroup                                              /* get the groupname */
if result="" then exit
selgrp=result

init=1
call getsys selgrp init                                    /* get sys into vars */
init=0

/* check security and change if neseccary */
do while length(result)>0
   parse var result systag sysname sysproto sysaddr sysonline sysos
   QName = RXqueue("SET", "NFQUEUE")                       /* make q active */
   if sysonline="TRUE" then do                             /* is sys online */
      '@NFSECCL /N:'sysproto'::'sysaddr' /LISTIN /ALL',    /* get uid's & pw's */
      '2>nul ] rxqueue nfqueue'                            /* for system */
      do while queued()>0                                  /* as long as uid's */
         parse pull . uid pw secmgr .
         if uiD='USERID="<PUBLIC>",',                      /* check acc. allowed */
            & pw='PWD="*",',
            & secmgr='SECMGR,' then do                     /* sys is open */
               '@NFSECCL /N:'sysproto'::'sysaddr,          /* add default */
               ' /ADDIN:'duid,                             /* uid & pw ... */
               ' /PWD:'dpw,
               ' /ADDSECMGR',
               ' /ADDSVC:"ALL" 2>nul'
            if rc=0 then                                   /* log record */
               wrc=lineout(logfile,date() time()" USER:"duid,
                     " PWD:"dpw" added on SYSTEM "sysname,
                        sysproto sysaddr)
            else do
               wrc=lineout(logfile,date() time()" Add of USER "DUID,
                   " on SYSTEM "sysname sysproto sysaddr,
                   "not successfull rc="rc)
               no_success=1
               end
            if rc=0 then do
```

```
                          '@NFSECCL /N:'sysproto'::'sysaddr,      /* close <PUBLIC> */
                          ' /EDITIN /USERID:"<PUBLIC>"',
                          ' /PWD:"*" /DELSECMGR /DELSVC:"ALL" 2>nul'
                          if rc=0 then                                       /* log record */
                             wrc=lineout(logfile,date() time()" USER <PUBLIC>",
                                  " changed on SYSTEM "sysname,
                                    sysproto sysaddr)
                          else do
                             wrc=lineout(logfile,date() time()" Change of USER",
                                  " <PUBLIC> on SYSTEM "sysname sysproto sysaddr,
                                  "not successfull rc="rc)
                             no_success=1
                             end
                  end
               end
            end
         end
         else do
            wrc=lineout(logfile,date() time()' System:'sysname 'via:'sysproto,
                 'addr:'sysaddr' is offline')
            no_success=1
         end
         call getsys selgrp init                             /* get sys into vars */
      end
      wrc=lineout(logfile)                                   /* close logfile */
      if no_success then
         say "Error occured or some Systems are offline. See log file"
      exit
```

---

## 4.3 The Alert Manager

The Alert Manager is the service in NetFinity 5.0 that indicates how to process
the alerts. You can start programs, export data to databases, forward alerts to
managing machines, store the alerts to files and a lot more. In this example, we
add another action to the Alert Manager to forward all alerts to the managing
machine. We do this with a REXX procedure for a whole group.

### 4.3.1 SETALERT

In the beginning of the procedure we create the queue NFQUEUE. Then we
define the name of the log file and a variable named no_success. What we also
need is the name of the manager machine. The user has to enter that name and
use the command NFRSYSCL /GETSYS to get the possible protocols and
addresses for this machine. We do this because the manager may not be the
machine where we run the program, even though typically it is. mngr_prot and
mngr_addr are the two string variables that hold the protocols and addresses
separated by a blank. GETGROUP is the next action and is described in 4.2.1,
"GETGROUP" on page 129. The same is true for GETSYS. See 4.2.2, "GETSYS"
on page 131 for more information.

Within the DO WHILE loop that begins after the comment /* check security, we
get all alert actions defined in the target machine. NFALRTCL /LISTACT /ALL is
the command that gets that information. We check if there is an action
ALERTPOPUP and we check to be sure that the manager the user defined is able
to communicate with the client. If so, we delete this action using the NFALRTCL
/DELACT command and log that action to a file. If the manager can
communicate with the client we use the NFALRTCL /ADDACT command to add a

new action that forwards every alert to the manager. After writing the log record we go to the next system with a call to GETSYS and the loop starts again.

Below you see an example of the log file entries.

```
16 Oct 1996 17:54:03 POPUP ACTION  deleted on SYSTEM "tme05188" NETBIOS WTR05188
16 Oct 1996 17:54:03 ALERTFORWARD action  added on SYSTEM "tme05188" NETBIOS WTR05188
16 Oct 1996 17:54:03 System:"tmecli1" TCPIP 8235tr04.itso.ral.ibm.com  Addr:8235tr04.itso.ral.ibm.com is offl
ine
16 Oct 1996 17:54:04 POPUP ACTION  deleted on SYSTEM "tmecli1" TCPIP TMECLI1.itso.ral.ibm.com
16 Oct 1996 17:54:04 ALERTFORWARD action  added on SYSTEM "tmecli1" TCPIP TMECLI1.itso.ral.ibm.com
16 Oct 1996 17:54:06 ALERTFORWARD action  added on SYSTEM "tmecli1" NETBIOS WTR05095
16 Oct 1996 17:54:07 ALERTFORWARD action  added on SYSTEM "tme05188" TCPIP NTSERVA.itso.ral.ibm.com
```

*Figure 131. Setalert Log File*

Start the program by entering the name setalert. Enter the name of the manager machine. This entry is case-sensitive. Make sure you enter the name of the machine and not the address as they are not always the same.

```
This program changes alert manager settings in all
machines in a selected group
Please enter the system name of the manager machine
Attention this entry is case-sensitive
Press Ctrl+c then Enter for exit
tmecli1
Select one of the following groups
1 -> All
2 -> cm4279r
3 -> testgrp

1 - 3 -> 3
Some Systems are offline. See log file

[D:\ITSO]
```

### 4.3.1.1 Alert Manager GUI

After starting the Alert Manager from the GUI you get this window.

*Figure 132. Alert Manager*

Click on **Actions** to display the Alert Actions window.
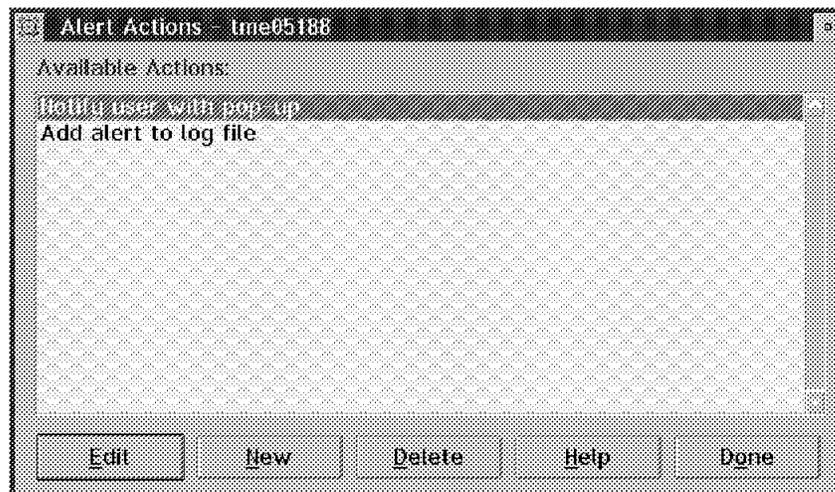


*Figure 133. Alert Actions*

### 4.3.1.2 NFALRTCL /LISTACT Output
The output of the command NFALRTCL /LISTACT looks like this:

```
[C:\]nfalrtcl /n:netbios::WTR05095 /listact /all
NetFinity Alert Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ TAG=0x0048FAD5, HANDLER="ALERTFORWARD",
  PARM0="NETBIOS", PARM1="WTR05188",
  TYPES=ANY,
  APPS=ANY,
  ATYPES=ANY,
  SEVS=ANY,
  SENDERS=ANY
}
{ TAG=0x0001A500, HANDLER="ALERTLOG",
  TYPES=ANY,
  APPS=ANY,
  ATYPES=ANY,
  SEVS={ 0, 1, 2, 3, 4, 5 },
  SENDERS=ANY
}

[C:\]
```

### 4.3.1.3  SETALERT.CMD

The setalert command follows:

```
/**/
logfile="NETFINCL.LOG"
no_success=0
QName = RXQUEUE("create", "NFQUEUE")                    /* define q */
    if QName \= "NFQUEUE" then do
        call RXQUEUE "DELETE", "NFQUEUE"
        call RXQUEUE "DELETE", QName
        call RXQUEUE "create", "NFQUEUE"
        end
QName = RXqueue("SET", "NFQUEUE")                       /* make q active */

/* Get protocol and address for later use */
found_mngr=0
'CLS'
do while \found_mngr
    'CLS'
    say 'This program changes alertmanager settings in all'
    say 'machines in a selected group'
    say 'Please enter the systemname of the manager machine'
    say 'Attention this entry is casesensitive'
    say 'Press Ctrl+c then Enter for exit'
    parse pull mngr
    mngr='"''mngr''"'
    '@NFRSYSCL /GETSYS /SYS:'mngr,                      /* get mngr protocols */
    ' 2>nul ] rxqueue NFQUEUE'
    if queued()>0 then found_mngr=1
end
mngr_prot=""
mngr_addr=""
do while queued() > 0                                   /* put prot + addr in */
    parse pull t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 .        /* string variables */
    select
    when t1="{" then do
        mngr_prot=mngr_prot]]substr(t4,7,length(t4)-7)" "   /* protocol */
    end
    when pos("ADDR=",t1)<>0 then do
        mngr_addr=mngr_addr]]substr(t1,7,length(t1)-8)" "   /* address */
    end
    otherwise nop
    end
end
/* Get all defined Groups */
call getgroup
if result="" then exit
selgrp=result

init=1
call getsys selgrp init
init=0
/* check security and change if neseccary */
do while length(result)>0
    parse var result systag sysname sysproto sysaddr sysonline sysos
    QName = RXqueue("SET", "NFQUEUE")                       /* make q active */

    if sysonline="TRUE" then do                            /* is sys online */
        '@NFALRTCL /N:'sysproto'::'sysaddr' /LISTACT /ALL',  /* get actions */
        ' 2>nul ] rxqueue nfqueue'                          /* for system */

        do while queued()>0                                /* as long as actions */
```

```
                   parse pull . tag handler .                        /* check all actions */
                   if handler='HANDLER="ALERTPOPUP",' &,
                      wordpos(sysproto,mngr_prot)>0 then do          /* if found ...*/
                      '@NFALRTCL /N:'sysproto'::'sysaddr,            /* delete POPUP */
                      ' /DELACT',
                      ' /TAG:"'substr(TAG,7,8)'"',
                      ' 1>nul 2>nul'
                      if rc=0 then                                   /* log record */
                         wrc=lineout(logfile,date() time()" POPUP ACTION ",
                            "deleted on SYSTEM "sysname sysproto sysaddr)
                      else do
                         wrc=lineout(logfile,date() time()" Delete POPUP action",
                            " on SYSTEM "sysname sysproto sysaddr,
                            "not successfull rc="rc)
                         no_success=1
                         end
                      end
             end                                                     /* check if manager */
             if wordpos(sysproto,mngr_prot)>0 then do                /* supports proto */
                addr_index=wordpos(sysproto,mngr_prot)
                '@NFALRTCL /N:'sysproto'::'sysaddr,                  /* add ALERTFORWARD */
                ' /ADDACT:"ALERTFORWARD"',
                ' /PARM0:'word(mngr_prot,addr_index),
                ' /PARM1:'word(mngr_addr,addr_index),
                ' 1>nul 2>nul'
                if rc=0 then                                         /* log record */
                   wrc=lineout(logfile,date() time()" ALERTFORWARD action ",
                      "added on SYSTEM "sysname sysproto sysaddr)
                   else do
                      wrc=lineout(logfile,date() time()" Add POPUP ALERTFORWARD action",
                         " on SYSTEM "sysname sysproto sysaddr,
                         "not successfull rc="rc)
                      no_success=1
                      end
                end
             else
                wrc=lineout(logfile,date() time()" Mngr does not support any ",
                   " protocol the client "sysname,
                   " is able to communicate with")

             end
          else do
             wrc=lineout(logfile,date() time()' System:'sysname sysproto sysaddr,
                ' Addr:'sysaddr' is offline')
             no_success=1
             end
          call getsys selgrp init
       end
       wrc=lineout(logfile)                                          /* close log file */
       if no_success then
          say "Some Systems are offline. See log file"
       exit
```

If the setting of the alert manager action was successful, you can see the results
in the Alert Actions GUI.

## 4.4 The Critical File Monitor

The Critical File Monitor generates an alert every time a monitored file is changed. You have to define which files are to be monitored. To do this use the NFCRTFCL command. This command has two options related to the file type. One is for system files and the other is for standard files. Use the /SETSYS and /DELSYS parameters for system files such as config.sys. Note that in OS/2, for example, the WIN.INI is not a system file. See 2.1.6.1, "Critical File Monitor Command Line Interface Example" on page 36 for a list of system files. System files are entered without the drive and path name. All other file names need to be entered with their complete drive and path information. The Critical File Monitor works on an internal clock, so you may not see an immediate alert after a file is changed.

## 4.4.1 SETCRTF

After defining a queue and some other variables we ask the user what file name to monitor and store its name in a stem variable called file_info.1. Remember that for non-system files you have to enter its full path name. At the prompt the users are asked to add a \ at the end of the path but we check for that anyway and add one if needed. We also check to see if the severity parameter is in the correct range from 0 to 7. Then we ask for the group to work with and call GETGROUP to get all systems in that group.

The only things left are to check to see if this is a system file and to send the command NFCRTFCL to activate these critical files. The check to see if this is a system file or not is done by checking for a path entry in file_info.file_no.2. If there is no entry, then it's a system file. Writing the log record completes the task for this item. The program ends when all systems within the selected group are updated.

### 4.4.1.1 The Critical File Monitor Program

After starting the program with the setcrtf command the user has to enter the name of the critical file without the path.

```
This program adds filenames to the Critical File Monitor
in all machines in a selected group
Enter filename (without path). Leave blank for no more files
win.ini
```

Then the user has to enter the path for that file. If it is a system file such as config.sys, don't enter the path.

```
This program adds filenames to the Critical File Monitor
in all machines in a selected group
Enter filename (without path). Leave blank for no more files
win.ini
Enter path (i.e. c:\dir1\). Leave blank if system file
c:\os2\mdos\winos2\
```

The last question about the file is to find out its alert severity. Enter a value from 0 to 7 where 0 is the highest severity.

```
This program adds filenames to the Critical File Monitor
in all machines in a selected group
Enter filename (without path). Leave blank for no more files
win.ini
Enter path (i.e. c:\dir1\). Leave blank if system file
c:\os2\mdos\winos2\
Enter severity 0 - 7. 0 is highest
2
```

After all of those steps are completed, the program starts over again and the
user can enter another file name for the Critical File Monitor. If nothing is
entered, the group selection is displayed. Select the group by its number.

```
This program adds filenames to the Critical File Monitor
in all machines in a selected group
Enter filename (without path). Leave blank for no more files

Select one of the following groups
1 -> All
2 -> cm4279r
3 -> testgrp

1 - 3 -> 3

[D:\ITSO]
```

If you get any error messages, check the log file.

```
16 Oct 1996 18:01:17 Critical File  win.ini  added on SYSTEM "tme05188" NETBIOS WTR05188
16 Oct 1996 18:01:17 System:"tmecli1" TCPIP 8235tr04.itso.ral.ibm.com  Addr:8235tr04.itso.ral.ibm.com is offl
ine
16 Oct 1996 18:01:17 Critical File  win.ini  added on SYSTEM "tmecli1" TCPIP TMECLI1.itso.ral.ibm.com
16 Oct 1996 18:01:34 Add Critical File  win.ini  on SYSTEM "tmecli1" NETBIOS WTR05095 not successfull rc=5
16 Oct 1996 18:01:35 Critical File  win.ini  added on SYSTEM "tme05188" TCPIP NTSERVA.itso.ral.ibm.com
```

*Figure 134. Setcrtf Log File*


## 4.4.1.2 Critical File Monitor GUI
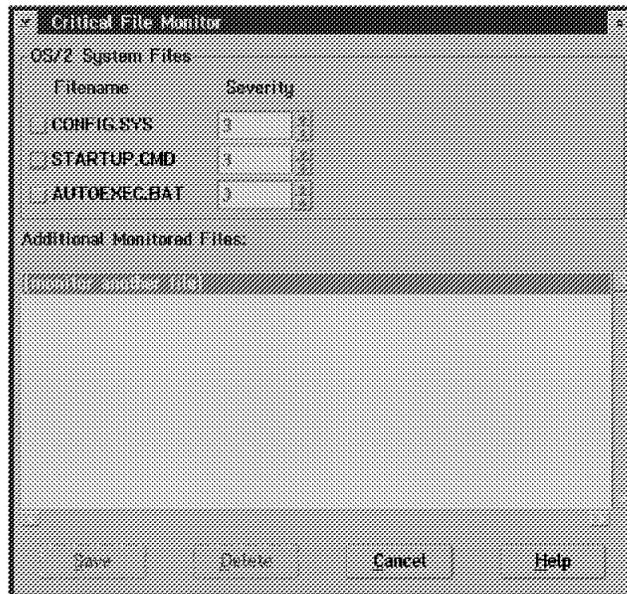
*Figure 135. Critical File Monitor*

---
> **Note**
>
> The GUI on the target (managed machine) will not show any files being monitored since the information is stored at the manger.  The manager can see the files in the GUI, as in the example below.
---
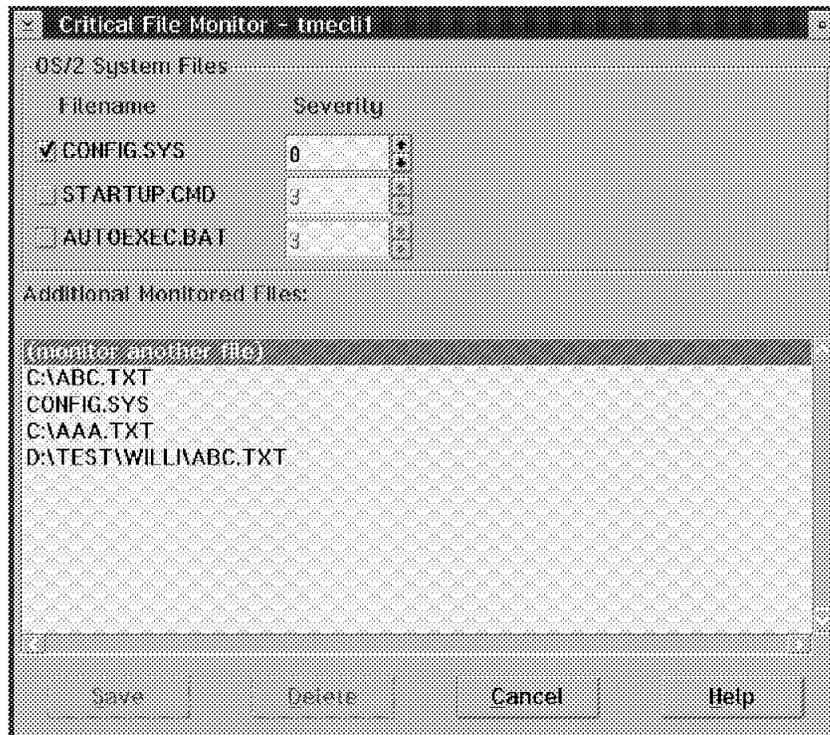


*Figure 136. Critical File Monitor Using the Remote Systems Manager*

### 4.4.1.3  NFCRTFCL /LIST Command Output

The Critical File Monitor line mode command output follows:

```
[C:\]nfcrtfcl /n:netbios::WTR05095 /lis
NetFinity Critical File Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ FILE="C:CONFIG.SYS", SEV=0, TYPE=SYSTEM_FILE }
{ FILE="C:\ABC.TXT", SEV=0, TYPE=STANDARD_FILE}
{ FILE="C:\AAA.TXT", SEV=0, TYPE=STANDARD_FILE}
{ FILE="D:TEST\WILLI\ABC.TXT", SEV=7, TYPE=STANDARD_FILE}

[C:\]
```

### 4.4.1.4  SETCRTF.CMD

Our sample program for the Critical File Monitor follows:

```
/**/
logfile="NETFINCL.LOG"
no_success=0
QName = RXQUEUE("create", "NFQUEUE")                        /* define q */
   if QName \= "NFQUEUE" then do
       call RXQUEUE "DELETE", "NFQUEUE"
       call RXQUEUE "DELETE", QName
       call RXQUEUE "create", "NFQUEUE"
       end
QName = RXqueue("SET", "NFQUEUE")                           /* make q active */

'CLS'
file_no=0
do forever                                                 /* get filenames */
   file_no=file_no+1
   'CLS'
   say 'This program adds filenames to the Critical File Monitor'
   say 'in all machines in a selected group'
   say 'Enter filename (without path). Leave blank for no more files'
   parse pull file_info.file_no.1
   if length(file_info.file_no.1)=0 then leave
   say 'Enter path (i.e. c:\dir1\). Leave blank if system file'
   parse pull file_info.file_no.2
   if length(file_info.file_no.2)>0 &,
      substr(file_info.file_no.2,length(file_info.file_no.2),1)<>"\" then
      file_info.file_no.2=file_info.file_no.2]]"\"
   do while pos(file_info.file_no.3,"01234567")=0
      say 'Enter severity 0 - 7. 0 is highest'
      parse pull file_info.file_no.3
   end
end
file_no=file_no-1
/* Get all defined Groups */
call getgroup
if result="" then exit
selgrp=result

init=1
call getsys selgrp init
init=0
do while length(result)>0
   parse var result systag sysname sysproto sysaddr sysonline sysos

   if sysonline="TRUE" then                                /* is sys online */
      do i=1 to file_no
         if length(file_info.i.2)=0 then                   /* system file */
            '@NFCRTFCL /N:'sysproto'::'sysaddr,
            ' /SETSYS:'file_info.i.1,
            ' /SEV:'file_info.i.3,
            ' 2>nul'
         else                                              /* standard file */
            '@NFCRTFCL /N:'sysproto'::'sysaddr,
            ' /SETSTD:'file_info.i.2]]file_info.i.1,
            ' /SEV:'file_info.i.3,
            ' 2>nul'


            if rc=0 then                                   /* log record */
               wrc=lineout(logfile,date() time()" Critical File ",
```

```
                             file_info.i.1,
                             " added on SYSTEM "sysname sysproto sysaddr)
                       else do
                       wrc=lineout(logfile,date() time()" Add Critical File ",
                             file_info.i.1,
                             " on SYSTEM "sysname sysproto sysaddr,
                             "not successfull rc="rc)
                       no_success=1
                       end
             end
         else do
            wrc=lineout(logfile,date() time()' System:'sysname sysproto sysaddr,
                  ' Addr:'sysaddr' is offline')
            no_success=1
            end

      call getsys selgrp init
   end
   wrc=lineout(logfile)                                    /* close logfi le */
   if no_success then
      say "Error occured or some Systems may offline. See log file"
   exit
```

---

## 4.5  The Process Manager

With the Process Manager you can observe processes in the clients and you can generate alerts if these processes stop.  You also can generate an alert when a process starts or even when the process runs.  Additionally, there is a parameter for the time the process has to run before the alert is generated.  For example, if you want to get an alert when the user stops his or her host emulation, you add a process monitor alert for that event with /ONSTOP=ENABLED.  Use /ONSTART=ENABLED to observe the start of a specific program or use /ONNORUN=5 when you want an alert generated when a program is not running for 5 minutes.

## 4.5.1  SETPROC

Our program asks the user for a process name and if the user wants to add or delete this process.  If the user answers A to add the program, we also ask for a severity for the alert.  It is not necessary to enter the path of the program.  Also you can enter /ALL if you want to delete all monitored processes on all machines in that group before adding new processes.  When you enter /ALL you will not be asked to add or delete the severity.  After getting the group and systems the program deletes or adds the desired processes for the selected system.  As in all other program samples we write a log record to document what happened.

### 4.5.1.1  The Process Manager Program

Start the program with its name.  As with the Critical File Monitor program you first have to enter the names and severities of the processes.  Also you can enter /all to delete all defined Process Manager alerts in the group.  The following sample screens show the necessary inputs and the responses.

```
This program adds or deletes processes to the
Process Manager Monitor in all machines in a selected group
Enter the program name (with .exe or .com).
Leave blank for no more processes
Enter /ALL if you want to delete all process monitoring
/all
```

As you can see after entering /all, the program loops back and asks you for a process name. It is not necessary to enter the full path name of the process. Then enter a D for delete or an A to add. The difference between this delete and the /all delete is that this delete deletes only the Process Manager entry with this specific name. All other Process Manager alert settings are still in effect. Another necessary entry is the severity which must be between 0 and 7.

```
This program adds or deletes processes to the
Process Manager Monitor in all machines in a selected group
Enter the program name (with .exe or .com).
Leave blank for no more processes
Enter /ALL if you want to delete all process monitoring
mahjongg.exe
Enter D for delete or A for add
a
Enter severity 0 - 7. 0 is highest
2
```

All entries are done for that specific process. The program loops and asks you again for a process name. Leave it blank if you are done. Select the group by entering its number.

```
This program adds or deletes processes to the
Process Manager Monitor in all machines in a selected group
Enter the program name (with .exe or .com).
Leave blank for no more processes
Enter /ALL if you want to delete all process monitoring

Select one of the following groups
1 -> All
2 -> cm4279r
3 -> testgrp

1 - 3 -> 3

Error occurred or some Systems may offline. See log file
[D:\ITSO]
```

You may want to look at the log file. Remember that the name of the log file is defined at the beginning of the program.

```
16 Oct 1996 18:09:27 Process  /all deleted from monitor  on SYSTEM "tme05188" NETBIOS WTR05188
16 Oct 1996 18:09:28 Process  mahjongg.exe added to monitor  on SYSTEM "tme05188" NETBIOS WTR05188
16 Oct 1996 18:09:28 System:"tmecli1" TCPIP 8235tr04.itso.ral.ibm.com  Addr:8235tr04.itso.ral.ibm.com is offl
ine
16 Oct 1996 18:09:29 Process  /all deleted from monitor  on SYSTEM "tmecli1" TCPIP TMECLI1.itso.ral.ibm.com
16 Oct 1996 18:09:45 Adding process  mahjongg.exe to monitor  on SYSTEM "tmecli1" TCPIP TMECLI1.itso.ral.ibm.com not succ
essfull rc=4
16 Oct 1996 18:09:46 Process  /all deleted from monitor  on SYSTEM "tmecli1" NETBIOS WTR05095
16 Oct 1996 18:09:47 Process  mahjongg.exe added to monitor  on SYSTEM "tmecli1" NETBIOS WTR05095
16 Oct 1996 18:09:47 Process  /all deleted from monitor  on SYSTEM "tme05188" TCPIP NTSERVA.itso.ral.ibm.com
16 Oct 1996 18:09:48 Process  mahjongg.exe added to monitor  on SYSTEM "tme05188" TCPIP NTSERVA.itso.ral.ibm.com
```

*Figure  137.  Setproc Log File*

### 4.5.1.2 Process Manager GUI

Start the remote Process Manager by double-clicking on the **Remote System Manager**.
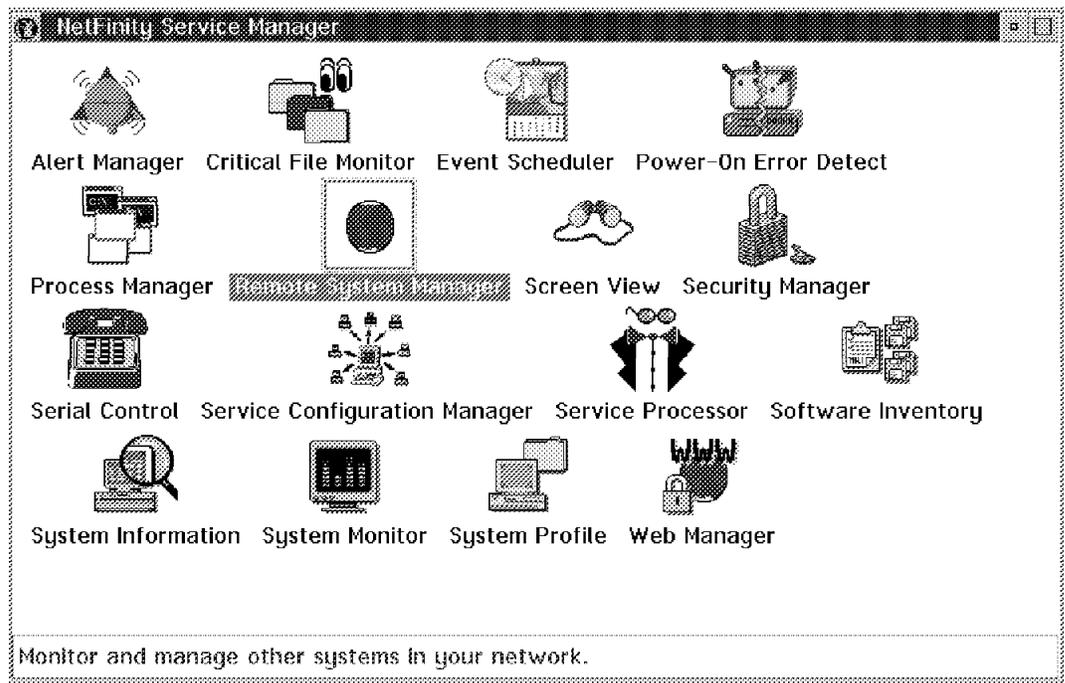


*Figure 138. NetFinity Service Manager*

Select the group with your target system by double-clicking on the appropriate group icon.
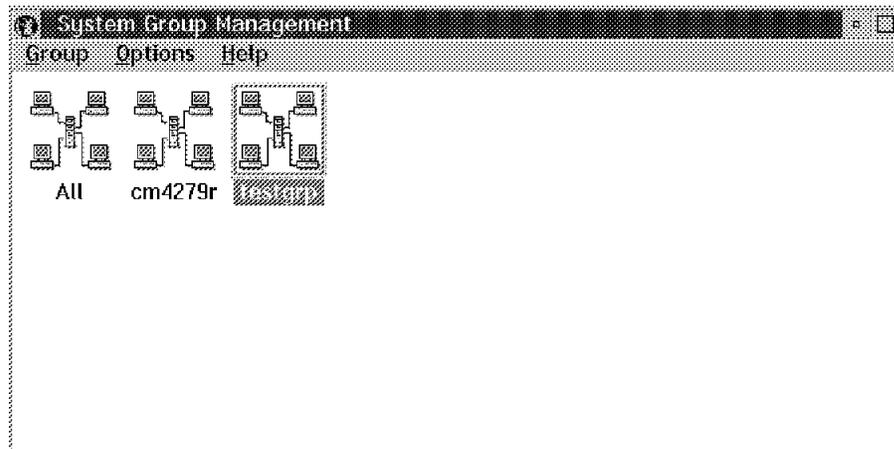


*Figure 139. System Group Management*

Select your target system by double-clicking on the system icon.
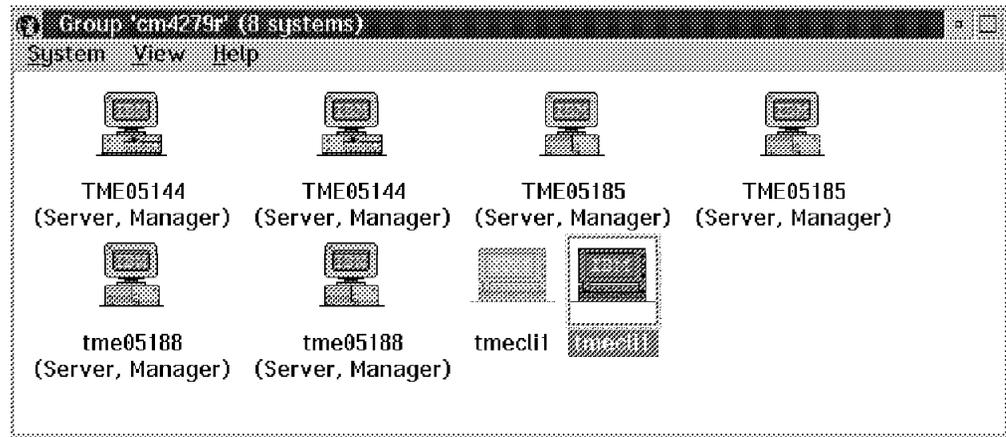
*Figure 140. Group 4279*

The NetFinity Service Manager window appears. If you look at the title bar you see that this is the Service Manager of your target machine. In this case it is for tmecli1.
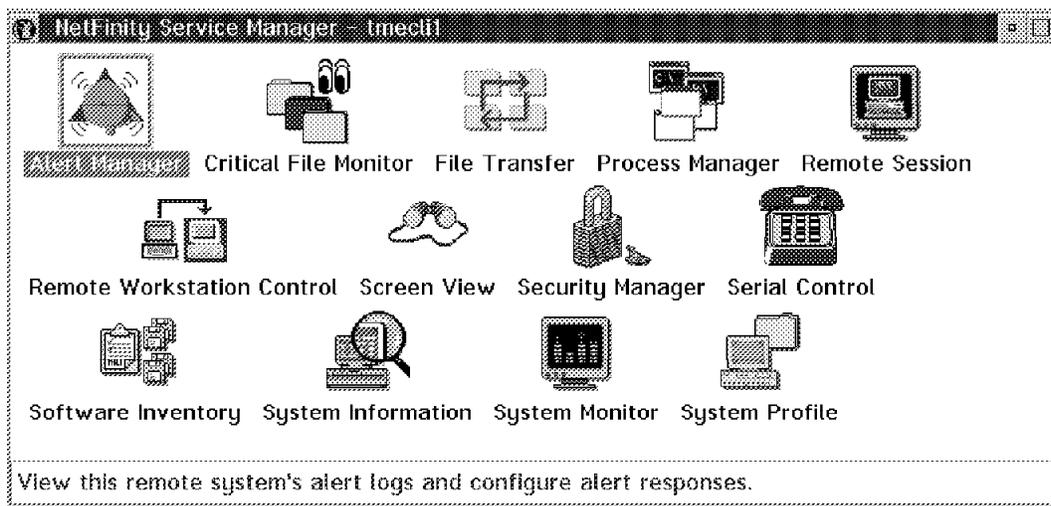


*Figure 141. NetFinity Service Manager - tmecli1*

Double-click on the **Process Manager** icon.
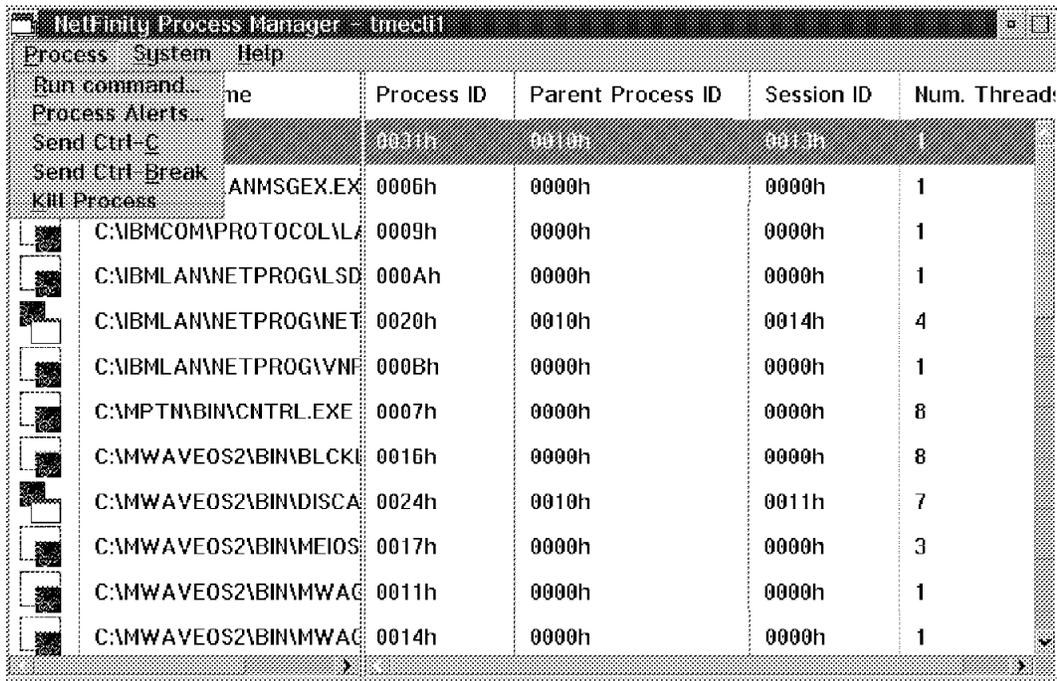
*Figure 142. NetFinity Process Manager*

Click on **Process Alerts** to get to the next window.
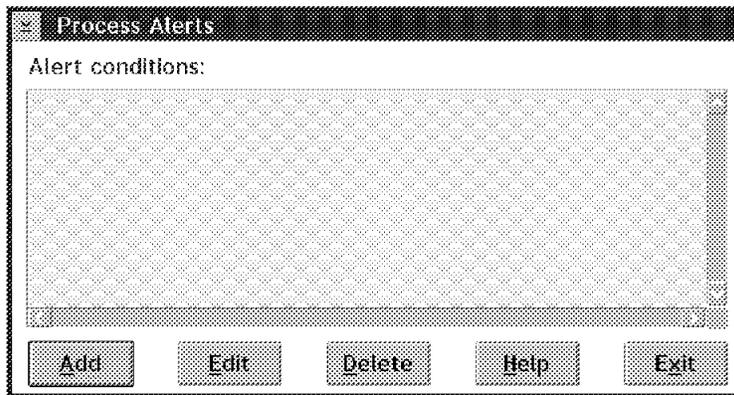


*Figure 143. Process Alerts*

Here you can define the process you want to monitor.
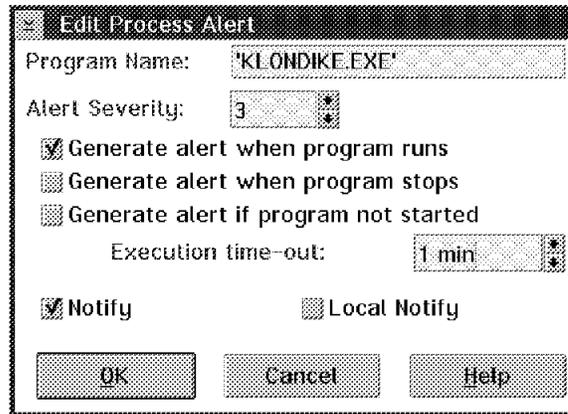
*Figure 144. Edit Process Alert*

### 4.5.1.3 Process Manager Command Line Output

The command line interface will permit us to accomplish with one command what just took several screens to navigate through:

```
[C:\]nfproccl /n:netbios::WTR05095 /listmon /all
NetFinity Process Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ TAG=0x47092E4D, PROCNAME="KLONDIKE.EXE", SEV=3,
  ONSTART=ENABLED, ONSTOP=DISABLED, ONNORUN=DISABLED
  NOTIFY={ "NETBIOS::WTR05188" }
}
{ TAG=0x482938F3, PROCNAME="PMSPOOL.EXE", SEV=0,
  ONSTART=DISABLED, ONSTOP=ENABLED, ONNORUN=DISABLED
  NOTIFY={ "NETBIOS::WTR05188" }
}

[C:\]
```

### 4.5.1.4 SETPROC.CMD

```
/**/
logfile="NETFINCL.LOG"
no_success=0
QName = RXQUEUE("create", "NFQUEUE")                        /* define q */
   if QName \= "NFQUEUE" then do
       call RXQUEUE "DELETE", "NFQUEUE"
       call RXQUEUE "DELETE", QName
       call RXQUEUE "create", "NFQUEUE"
       end
QName = RXqueue("SET", "NFQUEUE")                           /* make q active */
'CLS'
proc_no=0
do forever                                                 /* get filenames */
   proc_no=proc_no+1
   'CLS'
   say 'This program adds or deletes processes to the'
   say 'Process Manager Monitor in all machines in a selected group'
   say 'Enter the program name (with .exe or .com).'
   say 'Leave blank for no more processes'
   say 'Enter /ALL if you want to delete all process monitoring'
   parse pull proc_info.proc_no.1
   if length(proc_info.proc_no.1)=0 then leave
   do while pos(proc_info.proc_no.2,"AD")=0 &,
           translate(proc_info.proc_no.1)<>'/ALL'
      say 'Enter D for delete or A for add'
      pull proc_info.proc_no.2
   end
   do while pos(proc_info.proc_no.3,"01234567")=0 &,
           proc_info.proc_no.2='A' &,
           translate(proc_info.proc_no.1)<>'/ALL'
      say 'Enter severity 0 - 7. 0 is highest'
      parse pull proc_info.proc_no.3
   end
   if translate(proc_info.proc_no.1)='/ALL' then proc_info.proc_no.2="D"
end
if length(proc_info.1.1)=0 then exit
proc_no=proc_no-1
/* Get all defined Groups */
call getgroup
if result="" then exit
selgrp=result
init=1
call getsys selgrp init
init=0
trace ?r
do while length(result)>0
   parse var result systag sysname sysproto sysaddr sysonline sysos
   if sysonline="TRUE" then                                /* is sys online */
      do i=1 to proc_no
         if proc_info.i.2="A" then do                      /* add process */
            '@NFPROCCL /N&clolon.'sysproto'::'sysaddr,
            ' /ADDMON&clolon.'proc_info.i.1,
            ' /SEV&clolon.'proc_info.i.3,
            ' /ADDNOTIFY&clolon.HERE',
            ' 2>nul'
            if rc=0 then                                   /* log record */
               wrc=lineout(logfile,date() time()' Process ',
                  proc_info.i.1' added to monitor',
                  ' on SYSTEM 'sysname sysproto sysaddr)
```

```
                          else do
                          wrc=lineout(logfile,date() time()' Adding process ',
                              proc_info.i.1' to monitor',
                              ' on SYSTEM 'sysname sysproto sysaddr,
                              'not successfull rc='rc)
                          no_success=1
                          end
                  end
                  else do                                      /* delete process */
                      if translate(proc_info.i.1)='/ALL' then
                        del_str=proc_info.i.1
                      else
                        del_str='/PROCNAME&clolon.'proc_info.i.1
                  '@NFPROCCL /N&clolon.'sysproto'::'sysaddr,
                  ' /DELMON 'del_str,
                  ' 2>nul'
                      if rc=0 then                             /* log record */
                        wrc=lineout(logfile,date() time()' Process ',
                            proc_info.i.1' deleted from monitor',
                            ' on SYSTEM 'sysname sysproto sysaddr)
                        else do
                        wrc=lineout(logfile,date() time()' Deleting process ',
                            proc_info.i.1' from monitor ',
                            ' on SYSTEM 'sysname sysproto sysaddr,
                            'not successfull rc='rc)
                        no_success=1
                        end
                  end
              end
          else do
              wrc=lineout(logfile,date() time()' System&clolon.'sysname sysproto sysaddr,
                  ' Addr&clolon.'sysaddr' is offline')
              no_success=1
              end
          call getsys selgrp init
      end
      wrc=lineout(logfile)                                      /* close log file */
      if no_success then
          say "Error occurred or some Systems may offline. See log file"
      exit
```

## 4.6  The System Monitors

The system monitors provide status information for various system resources.
For example, CPU utilization, disk usage and process counts can be monitored.
Each of these monitors has a threshold to send an alert if the value for that
threshold is exceeded.  In the following sample program we show how to set
thresholds for a complete group.

### 4.6.1  SETSMON

The program starts with some definitions that we will need later on in the
application.  For example, we define the name of the log file and a variable that
holds the error status.  The contents of this variable will be changed when one
system is not reachable (system is offline) or when the return code of some
command line functions is not 0.  The following variables are pointers used with
stem variables.  Stem variables are similar to arrays.  These variables are the

index pointers to specific fields in the stem. The stem sys is built from data captured from the NFRSYSCL /GETSYS command and the NFSMONCL /GETMON command. The screen shows the output of the NFRSYSCL /GETSYS command is shown below:

```
{ SYSTAG=0x0003B6DA, SYSNAME="tme05188", PROTO=TCPIP,
  ADDR="NTSERVA.itso.ral.ibm.com",
  ONLINE=TRUE, ONLN=DISABLED, OFFLN=DISABLED, PING=30,
  GRPLIST={ 0x00020022, 0x001504F9 },
  ERRORCOND={  },
  OS=OS2, OS_VER=3.00, MANAGER, SERVER, MAC=08005ACF1BA3
}
```

The sys stem looks like the diagram below.

```
systag|sysname|sysproto|sysaddr|sysonline|sysos|monitor fields
                _____/                \
              /                                           \
            /                                              \
          |                                                 |
          |mon_tag|mon_name|mon_sample|mon_units|mon_record|
          |mon_tag|mon_name|mon_sample|mon_units|mon_record|
          |mon_tag|mon_name|mon_sample|mon_units|mon_record|
```

*Figure 145. Monitor Fields*

A description of the fields follows:

**Systag**    Refers to the SYSTAG in /GETSYS output (only the last 8 hex digits of the ID)

**Sysname**   The name of the system

**Sysproto**  The communication protocol (for example, TCP/IP or NetBIOS)

**Sysaddr**   The ADDR field (for example, sysmgmt.ibm.com)

**Sysonline** Contains TRUE or FALSE depending on the state when queried with /GETSYS

**Sysos**     Describes the operating system

This is all of the data from the NFRSYSCL /GETSYS command. Until now this stem was two-dimensional. By adding the monitor fields it becomes three-dimensional as you can see in the stem structure. We address these fields in the program with, for example, sys.sys_no.mon_name.mon_no where sys is the name of the stem, sys_no points to the system line, mon_name is the monitor field and mon_no points to the specific monitor line.

The output of the NFSMONCL /GETMON command.

```
{ MONITOR_ID=0xAB348194, NAME="CPU Utilization",
  SAMPLE=5000, UNITS_LBL="Percent", RECORDING=DISABLED,
  MINVAL=0.000000, MAXVAL=100.000000, VALUE=1.662093
}
```

The monitor fields in sys stem with reference to the output of the NFSMONCL /GETMON command follow:

**mon_tag**        The MONITOR_ID.  Only the last 8 hex digits of the ID.

**mon_name**       The NAME field.

**mon_sample**     The SAMPLE time in milliseconds.

**mon_units**      UNITS_LBL contains monitor units.

**mon_record**     Either ENABLED or DISABLED.  This controls if the sampled data
                   is to be stored.

Continuing on with the program we define two queues:  one to get the system
data in and the other to get the monitor data.  Then we selected the group by
calling the external procedure getgroup.  Within the following loop we first get
the system data and then all available monitors for this system.  We add the
monitor tag, name and the units to the system stem and to a string variable
called all_mon.  This is only done if the monitor name does not exist in the
all_mon variable so at least we have a string containing all monitor tags, names
and units available in all machines in the group.  The reason for that is simply
that not all machines have all monitors.  For example, a PC350 has no battery
remaining monitor.  This string variable is parsed in the next program block into
three stems called mtab_tbl, mon_tbl and units_tbl.  The mon_tbl table is
displayed in the next block and the user has four choices.  If the user presses
Enter, the second and maybe third or fourth page of monitors is displayed.
Entering an X ends the program.  Entering a number of a monitor starts the next
block.  If the user enters a 0 ALL, monitors on all machines in the are deleted.

In the next block the user is asked to enter a name for the monitor threshold and
the values for that threshold.  The user is also asked if all old monitor thresholds
are to be deleted.  If the answer is Y, the thresholds for that specific monitor are
deleted.

The last question wants the user to confirm what he or she has entered.  If the
answer to that is Y, the NetFinity 5.0 commands are executed and a log record is
written.  If this is done, the program returns to display the monitor table and the
user can add another monitor threshold or end execution by entering X.

### 4.6.1.1  System Monitor GUI
As with all NetFinity 5.0 services you can set up monitors for a single machine or
a group of machines with the GUI.

Double-click on **Remote System Manager**.
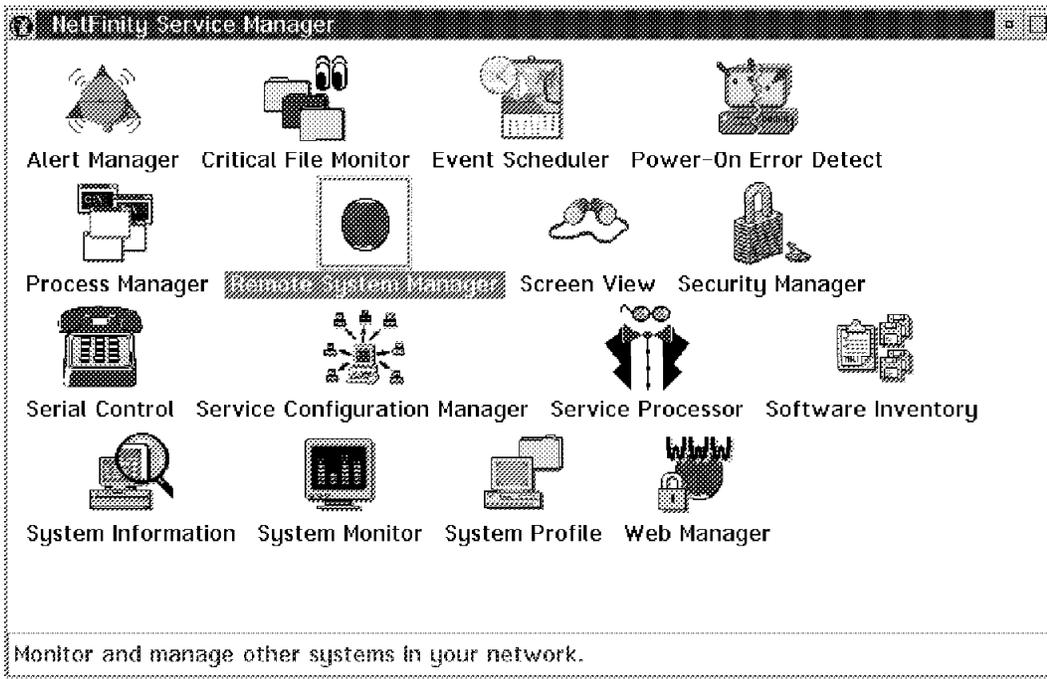
Figure 146. Remote System Manager

Double-click on the group that contains the target system.
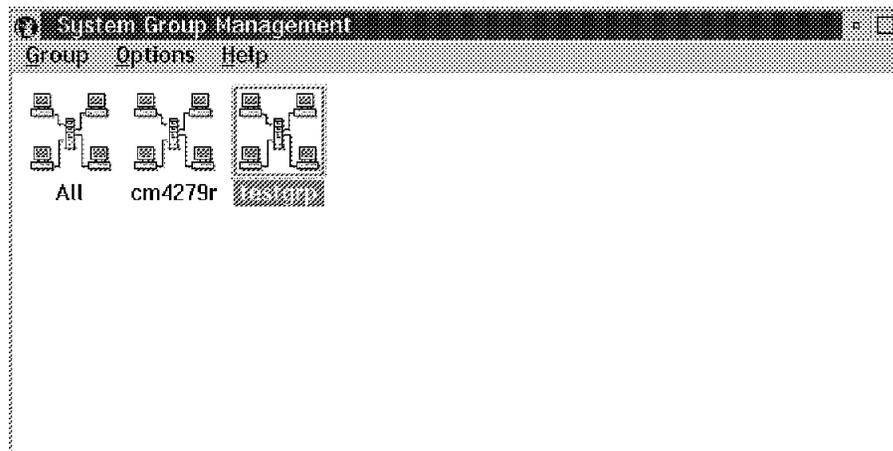


Figure 147. System Group Management

Select the system where you want to define the threshold.
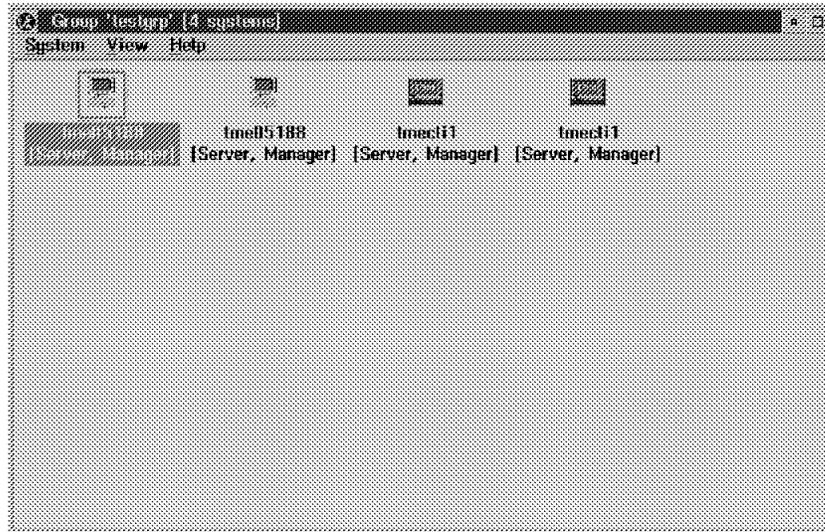
*Figure 148. Group - testgrp*

You now see the NetFinity Service Manager window of the remote system. Select **System Monitor** by double-clicking on its icon.
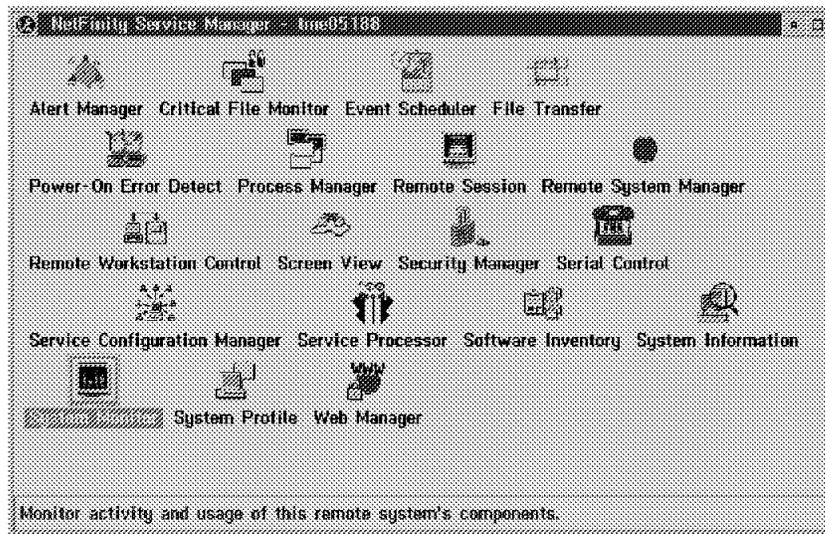


*Figure 149. NetFinity Service Manager*

Double-clicking on the **System Monitor** icon starts the GUI for it. Initially, only the CPU monitor will be displayed. You can add any of the other monitors that are available for your hardware.
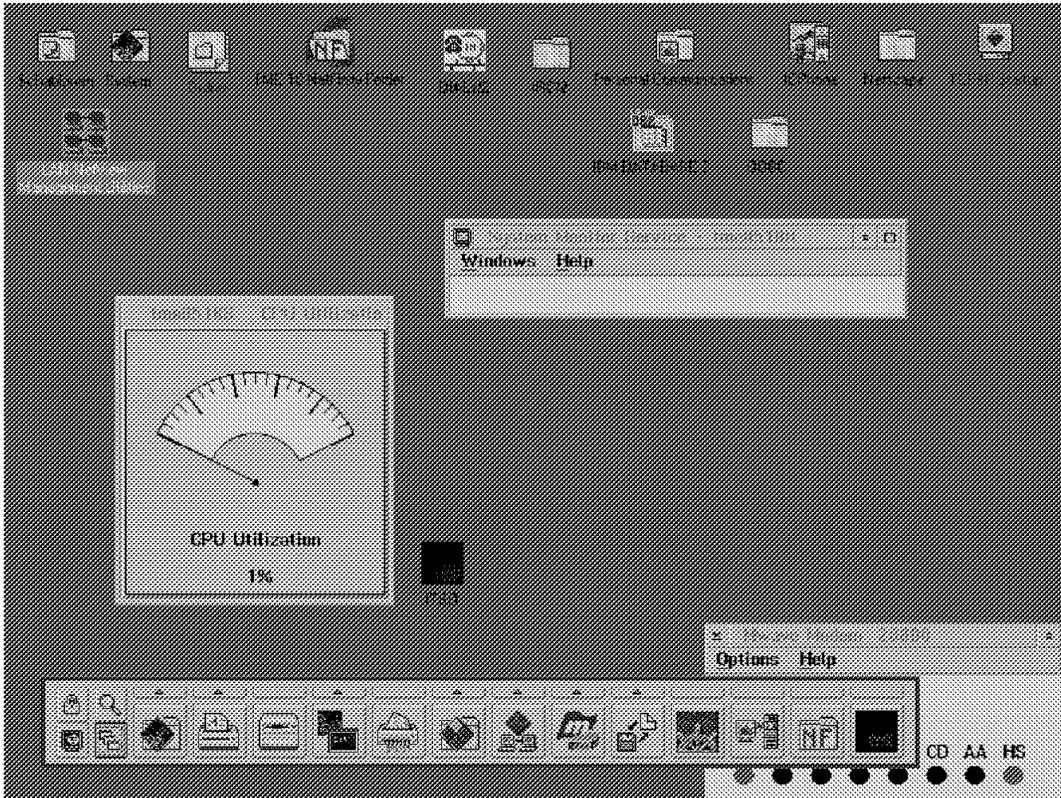
*Figure 150. System Monitor Service*

Select the desired monitor by clicking on **Windows** in the System Monitor Service window. Select **Show Monitors** from the pull-down menu. The Select Visible Monitors window will appear.

---
**Note**

If the System Monitor Service window is invisible, click with the right mouse button on one of the visible monitors and select **Main Menu** from the pull-down.

---

Select the monitor you need from the Select Visible Monitors window to set a threshold and click on **Accept**.
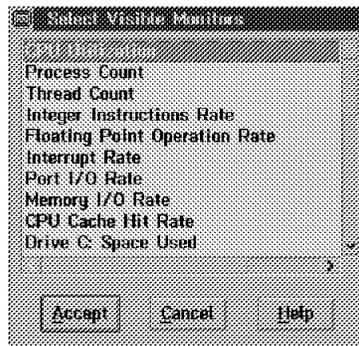


*Figure 151. Select Visible Monitors*

Click with the right mouse button on a monitor and then select **Open** and **Threshold**. The next window in which you can define your threshold values is displayed.
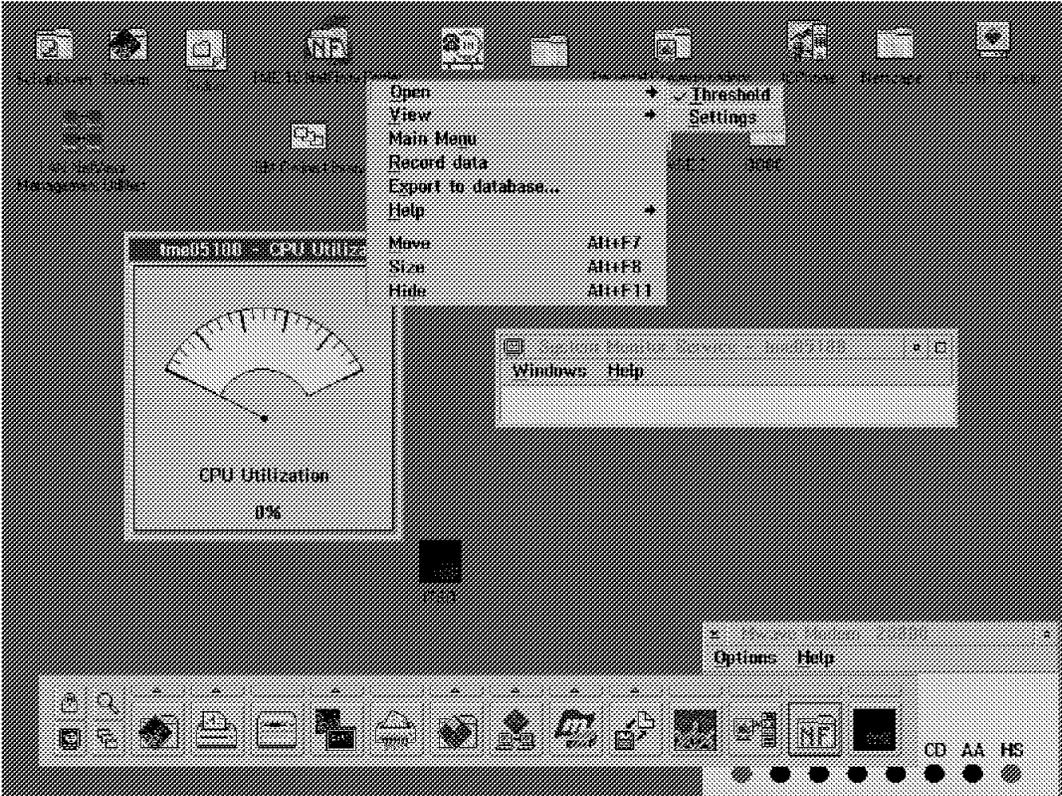


*Figure 152. System Monitors*
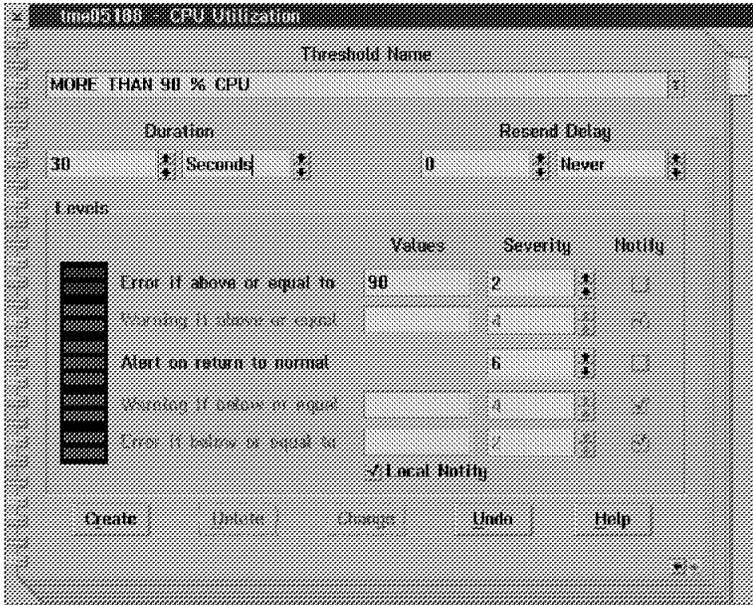


*Figure 153. CPU Utilization*

If you have defined the threshold as in the sample above, you see the following output when you start a NFSMONCL /GETMON /ALL on the target system.

```
[C:\lmu2]nfsmoncl /getmonthr /all
NetFinity System Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ MONITOR_ID=0xAB348194, THRESHNAME="CPU more than 90 %",
  DUR=30, RPT=0, HIERRVAL=90.000000, HIERRSEV=2, HIERRNTFY=ENABLED,
  HIWRNVAL=NONE,
  LOWRNVAL=NONE,
  LOERRVAL=NONE,
  RTNSEV=0, RTNNTFY=DISABLED, LOCALNTFY=DISABLED
}

[C:\lmu2]
```

All the setup work was done using the GUI. It can all be done with one
command. In addition, the GUI lets you do only one system at a time. Using the
command line interface, you can set up an entire group.

### 4.6.1.2  SETSMON Program
When you start the SETSMON.CMD you are prompted to enter the group.

```
[D:\itso]setsmon
Select one of the following groups
1 -> All
2 -> cm4279r
3 -> testgrp

1 - 3 ->
```

Next you can see if the program gets data from the machines within that group.
If it says No data returned, the machine is probably offline.

```
[D:\itso]setsmon
Select one of the following groups
1 -> All
2 -> cm4279r
3 -> testgrp

1 - 3 -> 2
Getting monitors from system "tme05188" via NETBIOS WTR05188
OK
Getting monitors from system "TME05185" via NETBIOS WTR05185
OK
Getting monitors from system "tmecli1" via NETBIOS WTR05095
OK
Getting monitors from system "TME05144" via NETBIOS WTR05144
No data returned
```

All available monitors are displayed next and you have to select one. As we
mentioned before you can also enter X to end the program or press Enter to
page forward. Another choice is to enter 0 which deletes ALL thresholds on al
machines in the selected group. If you enter 0, you will not be asked any further
questions but the program returns to the monitor list after deleting the
thresholds.

```
     1   Battery Remaining
     2   CPU Utilization
     3   Process Count
     4   Thread Count
     5   Integer Instructions Rate
     6   Floating Point Operation Rate
     7   Interrupt Rate
     8   Port I/O Rate
     9   Memory I/O Rate
    10   CPU Cache Hit Rate
    11   Drive C: Space Used
    12   Drive D: Space Used
    13   Drive C: Space Remaining
    14   Drive D: Space Remaining
    15   Drive C: % Space Used
    16   Drive D: % Space Used
    17   Print Jobs Queued
    18   Locked Memory
    19   Memory Usage
    20   Disk 1: Workload

  Select a monitor by its number.  Press Enter to display next page or X
  for program end.
  Enter 0 if you just want to delete all thresholds
```

If you select a monitor by its number, you are asked again whether you want to delete the old thresholds or not. If you enter Y, all thresholds for that specific monitor are deleted. All other thresholds are not touched. You then have to enter a threshold name. If you use a name that matches an existing threshold, it will not be added. Instead it will get edited. Next you have to enter the values for the attributes of this threshold. You do not have to enter all values, just press Enter if you don't want this value to be set. The default will then be taken. The duration attribute is set in seconds. If you wish for the duration to be one day, you would set it to 86400 seconds. When you have finished with the data entry you will be prompted to verify that all entries are OK. If you answer N (for no) you step through the questions again.

```
  Threshold settings for Battery Remaining in Percent

  Enter Y or N. Or enter R to return to the monitor list.
  Delete all other threshold settings ? Y/N/R--> n

  Enter threshold name -------> CPU more than 90 percent
  High Error value -----------> 90
  High Error severity 0-7 ----> 2
  High Warning value --------->
  High Warning severity 0-7 -->
  Low Error value ------------>
  Low Error severity 0-7 ----->
  Low Warning value ---------->
  Low Warning severity 0-7 --->
  Duration value (sec) -------> 30
  Return to normal severity -->

  Everything OK ? Y/N ->
```

There is a log file written for this REXX exec. An example of setsmon.log follows:

```
17 Oct 1996 11:34:58 System: "tmecli1" via  TCPIP Addr: 8235tr04.itso.ral.ibm.com  is offline
17 Oct 1996 11:36:20 Monitor threshold "CPU > 90 FOR 30 SEC"  added on SYSTEM  "tme05188" NETBIOS WTR05188
17 Oct 1996 11:36:22 Monitor threshold "CPU > 90 FOR 30 SEC"  added on SYSTEM  "tmecli1" TCPIP TMECLI1.itso.ral.ibm.com
17 Oct 1996 11:36:24 Monitor threshold "CPU > 90 FOR 30 SEC"  added on SYSTEM  "tmecli1" NETBIOS WTR05095
17 Oct 1996 11:36:26 Monitor threshold "CPU > 90 FOR 30 SEC"  added on SYSTEM  "tme05188" TCPIP NTSERVA.itso.ral.ibm.com
17 Oct 1996 15:45:43  Monitor thresholds  deleted on SYSTEM  "tme05188" NETBIOS WTR05188
17 Oct 1996 15:46:20 System: "tmecli1" via  TCPIP Addr: 8235tr04.itso.ral.ibm.com  is offline
17 Oct 1996 15:46:27  Monitor thresholds  deleted on SYSTEM  "tme05188" NETBIOS WTR05188
17 Oct 1996 15:46:28  Monitor thresholds  deleted on SYSTEM  "tmecli1" TCPIP TMECLI1.itso.ral.ibm.com
17 Oct 1996 15:46:29  Monitor thresholds  deleted on SYSTEM  "tmecli1" NETBIOS WTR05095
17 Oct 1996 15:46:30  Monitor thresholds  deleted on SYSTEM  "tme05188" TCPIP NTSERVA.itso.ral.ibm.com
```

*Figure  154.  Set System Monitor Log File*

## 4.6.1.3  SETSMON.CMD

```
/**/
no_success=0                                                    /* successfull oper. */
logfile="SETSMON.LOG"                                           /* def log file name */
systag=1                                                        /*  stem index    */
sysname=2                                                       /*      ""        */
sysproto=3                                                      /*      ""        */
sysaddr=4                                                       /*      ""        */
sysonline=5                                                     /*      ""        */
sysos=6                                                         /*      ""        */
mon_tag=10                                                      /*      ""        */
mon_name=11                                                     /*      ""        */
mon_sample=12                                                   /*      ""        */
mon_units=13                                                    /*      ""        */
mon_record=14                                                   /*      ""        */
no_success=0                                                    /*      ""        */
hierrval=10                                                     /*      ""        */
hierrsev=11                                                     /*      ""        */
hiwrnval=12                                                     /*      ""        */
hiwrnsev=13                                                     /*      ""        */
loerrval=14                                                     /*      ""        */
loerrsev=15                                                     /*      ""        */
lowrnval=16                                                     /*      ""        */
lowrnsev=17                                                     /*      ""        */
dur=18                                                          /*      ""        */
rtnsev=19                                                       /*      ""        */

QName = RXQUEUE("create", "NFQUEUE")                            /* define nf q */
if QName \= "NFQUEUE" then do
   call RXQUEUE "DELETE", "NFQUEUE"
   call RXQUEUE "DELETE", QName
   call RXQUEUE "create", "NFQUEUE"
   end
QName = RXQUEUE("create", "MONQUEUE")                           /* define mon q */
if QName \= "MONQUEUE" then do
   call RXQUEUE "DELETE", "MONQUEUE"
   call RXQUEUE "DELETE", QName
   call RXQUEUE "create", "MONQUEUE"
   end
QName = RXqueue("SET", "NFQUEUE")                               /* make nf q active */

call getgroup                                                  /* Get defined Groups */
if result="" then exit
selgrp=result
sys_no=0
init=1
call getsys selgrp init                                        /* get system data */
init=0
all_mon=''

do while length(result)>0
   sys_no=sys_no+1
   parse var result sys.sys_no.systag sys.sys_no.sysname sys.sys_no.sysproto,
        sys.sys_no.sysaddr sys.sys_no.sysonline sys.sys_no.sysos
   if translate(sys.sys_no.sysonline)='TRUE' then do
      say 'Getting monitors from system 'sys.sys_no.sysname' via ',
         sys.sys_no.sysproto sys.sys_no.sysaddr
      '@NFSMONCL /N:'sys.sys_no.sysproto'::'sys.sys_no.sysaddr,    /* get monitors */
            '/GETMON /ALL 2>nul ] rxqueue MONQUEUE'
      QName = RXqueue("SET", "MONQUEUE")                         /* actiate mon-q */
```

```
        mon_no=0
        if queued()>0 then say 'OK'
        else say 'No data returned '
        do while queued()>0
           parse pull mon_line
           if word(mon_line,1)='{' & pos('ATTRIB',mon_line)=0 then do   /* line 1 */
              mon_no=mon_no+1
              parse var mon_line ' MONITOR_ID=0x' t2 ', NAME=' t3 ',' .
              sys.sys_no.mon_tag.mon_no=t2                              /* mon tag */
              sys.sys_no.mon_name.mon_no=t3                             /* mon name */
              parse pull mon_line
              parse var mon_line t4 ',' t5 ',' t6 ',' .
              sys.sys_no.mon_sample.mon_no=substr(t4,10,length(t4)-9) /* sample rate */
              sys.sys_no.mon_units.mon_no=substr(t5,12,length(t5)-11) /* units_label */
              sys.sys_no.mon_record.mon_no=substr(t6,12,length(t6)-11) /* recording */
              if pos(t3,all_mon)=0 then do
                 all_mon=all_mon]]t2]]d2c(30)]]t3]]d2c(30)]],
                         sys.sys_no.mon_units.mon_no]]d2c(30)
              end
           end
        end
        QName = RXqueue("SET", "NFQUEUE")                       /* make nf-q active */
     end
     else do                                                   /* write log file */
        wrc=lineout(logfile,date() time()' System:',
            sys.sys_no.sysname' via ',
            sys.sys_no.sysproto' Addr:',
            sys.sys_no.sysaddr,
            ' is offline')
        no_success=1
        end
     call getsys selgrp init
  end

  mon_no=0                                                      /* create mon table */
  delim=d2c(30)
  do while length(all_mon)>0                                    /* stem mon_tbl holds */
     mon_no=mon_no+1                                            /* the monitors found */
     parse var all_mon mtag_tbl.mon_no (delim) mon_tbl.mon_no (delim),
           units_tbl.mon_no (delim) all_mon
  end



  mon_tbl_ix=0
  del_old_thr=''
  do forever                                                   /* display table to */
     'CLS'
     do disp_line = 1 to 20                                    /* select the monitor */
        mon_tbl_ix=mon_tbl_ix+1
        say mon_tbl_ix mon_tbl.mon_tbl_ix
        if disp_line=20 ] mon_tbl_ix=mon_no then do
           if mon_tbl_ix=mon_no then do
              mon_tbl_ix=0
              disp_line=20
              end
           say''
           say 'Select a monitor by its number.',
            ' Press Enter to display next page or X for end'
```

```
                say 'Enter 0 if you just want to delete all thresholds'
            pull mon_select                                    /* select from list */
            select
            when datatype(mon_select)='NUM' then
                if mon_select >= 0 & mon_select <= mon_no then do
                    call set_mon                               /* change thresholds */
                end
            when mon_select='X' then leave
            otherwise nop
            end
            'cls'
        end
    end
    if mon_select='X' then leave
end
wrc=lineout(logfile)                                           /* close log file */
if no_success then
    say "Some Systems are offline or operation failed. See log file"
exit


set_mon:                                                       /* changing thresholds */
target_mon_id=mtag_tbl.mon_select
entry_ok='N'
do while entry_ok<>'Y'
    'cls'                                                      /* building command */
    if mon_select<>0 then do
        say 'Threshold settings for 'mon_tbl.mon_select' in 'units_tbl.mon_select
        say ''
        say 'Enter Y or N. Or enter R to return to the monitor list.'
        do while pos(del_old_thr,'YNR')=0
            call charout ,'Delete all other threshold settings ? Y/N/R--> '
            pull del_old_thr
        end
        if del_old_thr='R' then do
            del_old_thr=''
            return
            end
        say''

        call charout ,'Enter threshold name -------> '
        pull thr_name
        do while length(thr_name)=0 & del_old_thr<>'Y' then do
            call charout ,'Enter threshold name -------> '
            pull thr_name
        end

        thr_name='"']]thr_name]]'"'
        prep_string=''

        call charout ,'High Error value -----------> '
        pull hierrval
        if length(hierrval)>0 then prep_string=prep_string]]'/HIERRVAL:']]hierrval]]' '

        call charout ,'High Error severity 0-7 ----> '
        hierrsev=get_severity()
        if length(hierrsev)>0 then prep_string=prep_string]]'/HIERRSEV:']]hierrsev]]' '

        call charout ,'High Warning value ---------> '
```

```
        pull hiwrnval
        if length(hiwrnval)>0 then prep_string=prep_string]]'/HIWRNVAL:']]hiwrnval]]' '

        call charout ,'High Warning severity 0-7 --> '
        hiwrnsev=get_severity()
        if length(hiwrnsev)>0 then prep_string=prep_string]]'/HIWRNSEV:']]hiwrnsev]]' '

        call charout ,'Low Error value ------------> '
        pull loerrval
        if length(loerrval)>0 then prep_string=prep_string]]'/LOERRVAL:']]loerrval]]' '

        call charout ,'Low Error severity 0-7 -----> '
        loerrsev=get_severity()
        if length(loerrsev)>0 then prep_string=prep_string]]'/LOERRSEV:']]loerrsev]]' '

        call charout ,'Low Warning value ----------> '
        pull lowrnval
        if length(lowrnval)>0 then prep_string=prep_string]]'/LOWRNVAL:']]lowrnval]]' '

        call charout ,'Low Warning severity 0-7 ---> '
        lowrnsev=get_severity()
        if length(lowrnsev)>0 then prep_string=prep_string]]'/LOWRNSEV:']]lowrnsev]]' '

        call charout ,'Duration value (sec) -------> '
        pull dur
        if length(dur)>0 then prep_string=prep_string]]'/DUR:']]dur]]' '

        call charout ,'Return to normal severity --> '
        pull rtnsev
        if length(rtnsev)>0 then prep_string=prep_string]]'/RTNSEV:']]rtnsev]]' '
        say ''
        say 'Everything OK ? Y/N -> '
        pull entry_ok
        end
    else do
        del_old_thr='Y'
        entry_ok='Y'
        thr_name=''
        end
end

do sys_sel=1 to sys_no

    if translate(sys.sys_sel.sysonline)='TRUE' then do
    if del_old_thr='Y' then do
        if mon_select=0 then
          '@NFSMONCL /N:']]sys.sys_sel.sysproto]]'::']],  /* del all thresholds */
                  sys.sys_sel.sysaddr]],
                  ' /DELMONTHR /ALL 1>nul 2>nul'
        else                                            /* del selected thresh. */
          '@NFSMONCL /N:']]sys.sys_sel.sysproto]]'::']],
                  sys.sys_sel.sysaddr]],
                  ' /DELMONTHR /MONID:'target_mon_id,
                  ' 1>nul 2>nul'
        say 'Deleting on system 'sys.sys_sel.sysname
        if rc=0 then                                    /* log record */
            wrc=lineout(logfile,date() time(),
              " Monitor thresholds",
              " deleted on SYSTEM ",
```

```
                          sys.sys_sel.sysname ,
                          sys.sys_sel.sysproto ,
                          sys.sys_sel.sysaddr)
                  else do
                  wrc=lineout(logfile,date() time(),
                      " Delete monitor thresholds ",
                      " on SYSTEM ",
                      sys.sys_sel.sysname ,
                      sys.sys_sel.sysproto ,
                      sys.sys_sel.sysaddr,
                      " not successfull rc="rc)
                  no_success=1
                  end
            if mon_select=0 then iterate                    /* no more work */
        end                                                 /* end deletion */


                                                            /* add threshold */
        cmd_string='@NFSMONCL /N:']],                       /* build command */
                  sys.sys_sel.sysproto]]'::']],
                  sys.sys_sel.sysaddr]]' /MONID:'target_mon_id,
                  ' /ADDMONTHR:']]thr_name]]' ']]prep_string]],
                  '/HIERRNTFY:ENABLE 1>nul 2>nul'
        cmd_string                                          /* exec command */
        say 'Adding threshold on system 'sys.sys_sel.sysname
        if rc=0 then                                        /* log record */
            wrc=lineout(logfile,date() time(),
                " Monitor threshold "thr_name,
                " added on SYSTEM ",
                sys.sys_sel.sysname ,
                sys.sys_sel.sysproto ,
                sys.sys_sel.sysaddr)
            else do
                wrc=lineout(logfile,date() time(),
                    " Add monitor threshold "thr_name,
                    " on SYSTEM ",
                    sys.sys_sel.sysname ,
                    sys.sys_sel.sysproto ,
                    sys.sys_sel.sysaddr,
                    " not successfull rc="rc)
                no_success=1
                end

    end                                                     /* end if online */
end
del_old_thr=''
return


get_severity:
sev=''
do while pos(sev,'0 1 2 3 4 5 6 7')=0
    pull sev
    if sev=''  then return sev
    if pos(sev,'0 1 2 3 4 5 6 7')=0 then say 'Severity must be 0 to 7. Please reenter'
end
return sev
```

# Chapter 5.  LMU Migration

This chapter shows a mapping between LMU for OS/2 functions and NetFinity.  It also gives examples of how to use the new NetFinity functions to replace some existing LMU functions.

When looking at the table of contents in *LAN NetView Management Utilities for OS/2*, SC30-3555-06, we see that each of these 12 chapters can be addressed by at least one NetFinity 5.0 component, as we have outlined in the NetFinity 5.0 redbooks.

1.  LMU Overview - see Chapter 1, "Project Overview" on page 1.

2.  Starting the Components - see Chapter 1, "Project Overview" on page 1.

3.  Using Fault Management - see 2.1.4, "Alert Manager - NFALRTCL.EXE" on page 23.

4.  Customizing the Graphical User Interface - see Chapter 3, "Webability" on page 43.

5.  Controlling LMU Components and Systems - see Chapter 2, "Managing Clients from the Command Line" on page 9.

6.  Collecting Vital Product Data - see *NetFinity V5.0 Database Support*, SG24-4808.

7.  Monitoring System Performance - see 2.1.5, "System Monitor - NFSMONCL.EXE" on page 31 and 5.4, "Monitoring Processes and System Performance" on page 196.

8.  Creating Generic Alerts from User Programs - see 5.5.1, "Generate Alerts" on page 220.

9.  Scheduling Events - see 3.2.4, "Event Scheduler" on page 61.

10. Managing LMU from Other Platforms - see 5.1, "SNMP Configuration."

11. Using the LMU Database - see 5.1, "SNMP Configuration" and 5.4.3, "SPM/2 and Database Queries" on page 203.

12. Managing User Data - see *TME 10 NetFinity V5.0 Database Support*, SG24-4808.

## 5.1  SNMP Configuration

The configuration of SNMP is different for the operating systems we used.  It causes many problems, which here we describe how to avoid.

### 5.1.1  SNMP Configuration with the SystemView Agent for OS/2

If you use the SystemView Agent for OS/2 that comes with OS/2 Warp V4, you will use the configuration program that is located in \svca\bin\agent\hrmcnfig.exe to set up your SNMP configuration.

In previous releases of the SystemView Agent for OS/2, the configuration folder appeared during the installation process.  If you chose to install the SystemView Agent for OS/2 as part of your base OS/2 V4 installation, then it only installs the product without giving you any option to tailor it.
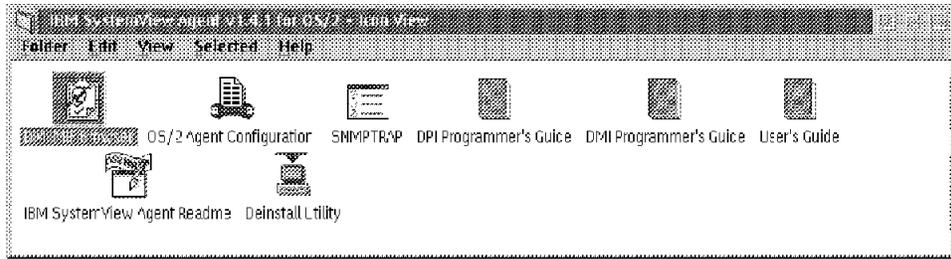
*Figure 155. IBM SystemView Agent for OS/2 Folder*

The first thing you need to set up is a trap destination. That is where all SNMP traps will get sent. You can have more than one trap destination.
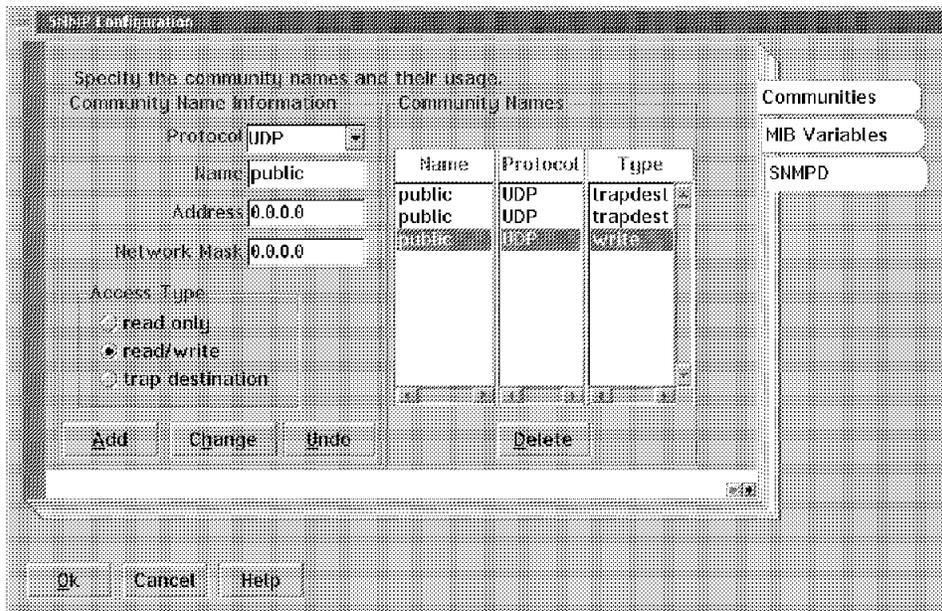


*Figure 156. SNMP Configuration - Communities, Trap Destination*

The next thing we need is the access definition. You have to grant read-only or read/write access to either a specific address or to a subnet. In our example we grant read/write access to everyone.
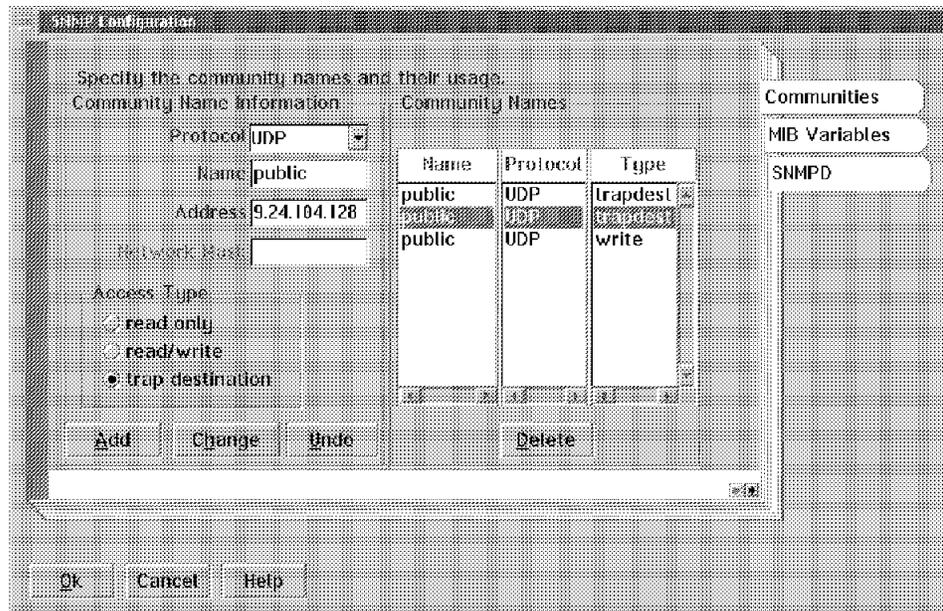
Figure 157. SNMP Configuration - Communities, Access Definition

There are also some entries for the MIB values we defined. You should update the fields for Description, Contact, Name and Location. This information will be used during normal SNMP management.
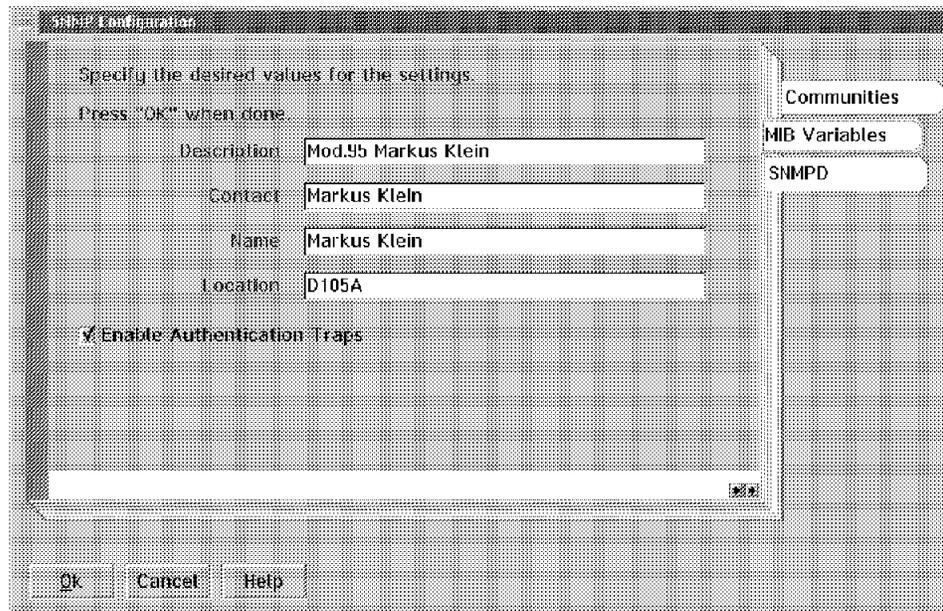


Figure 158. SNMP Configuration - MIB Variables

Finally we need to specify the parameters for the start of the SNMP daemon SNMPD.EXE. By default all transports and all DPIs are specified, but we only needed some of them. We used UDP as our transport mechanism and TCP for our DPI specification. If you wish to use some other functions of TCP/IP such as the subagent MIB_2.EXE, you will also need to specify shm on the UDP parameter. In addition, we added a debug parameter, which shows us all incoming SNMP requests and all outgoing SNMP responses and traps. Start snmpd ? to see all debugging (and other) options.

The customization will create a file SVASTART.CMD which will look like the following:

```
start snmpd.exe -transport udp -dpi tcp shm
start dmisa.exe
```

We added the parameter -d 7 for debugging options.
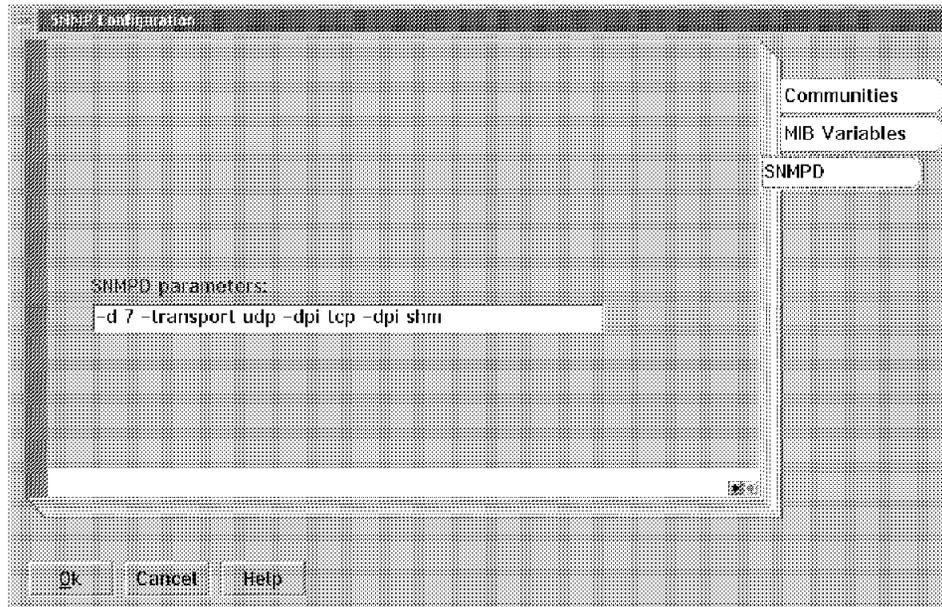


*Figure 159. SNMP Configuration - SNMPD Parameters*

After you have entered all your information click on **OK**. Restart your computer and SNMP should be ready to receive requests.

---
**Remark**

If you want to start MIB_2.EXE with the startup of TCP/IP, create a file x:\tcpip\etc\tcpexit.cmd that looks like:

```
start /min mib_2.exe
```
---

## 5.1.2 SNMP Configuration in TCP/IP for OS/2 Version 3.x

If you are using OS/2 Warp Version 3, SNMP will get installed as part of the TCP/IP installation process. Therefore, the configuration takes place in the TCP/IP Configuration folder. You will still have to provide the same type of information as we just did in the SystemView Agent for OS/2.

First you specify the Contact Name and the System Location which will get placed in the MIB.
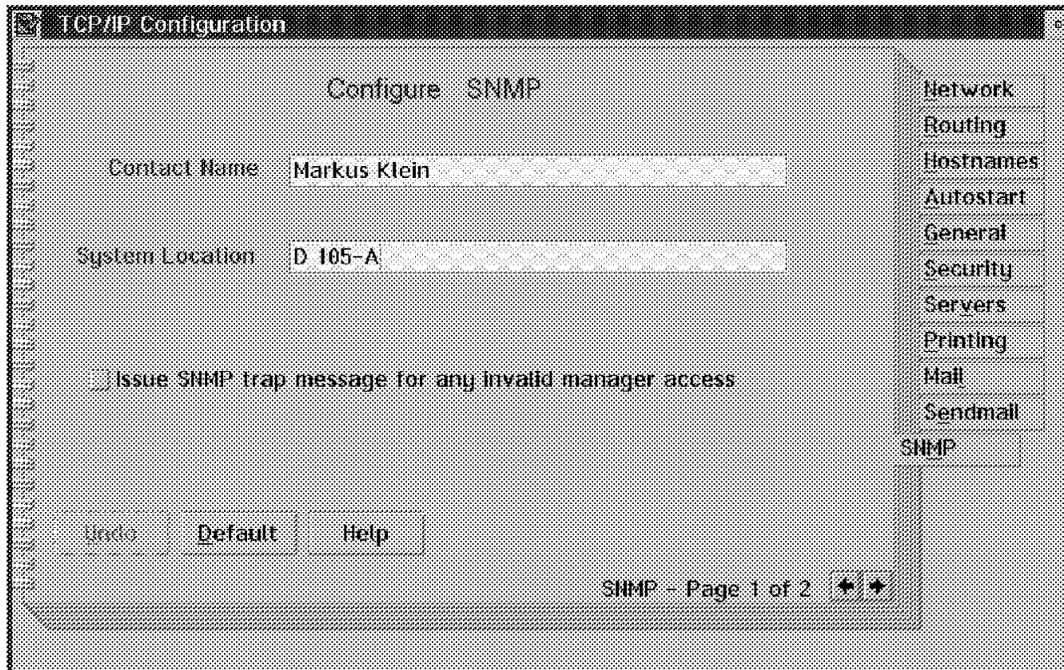
*Figure 160. TCP/IP Configuration - SNMP Page 1*

Page two shows the definitions for SNMP Trap Destination and SNMP Manager Access Authorization. By default no entries are made here.
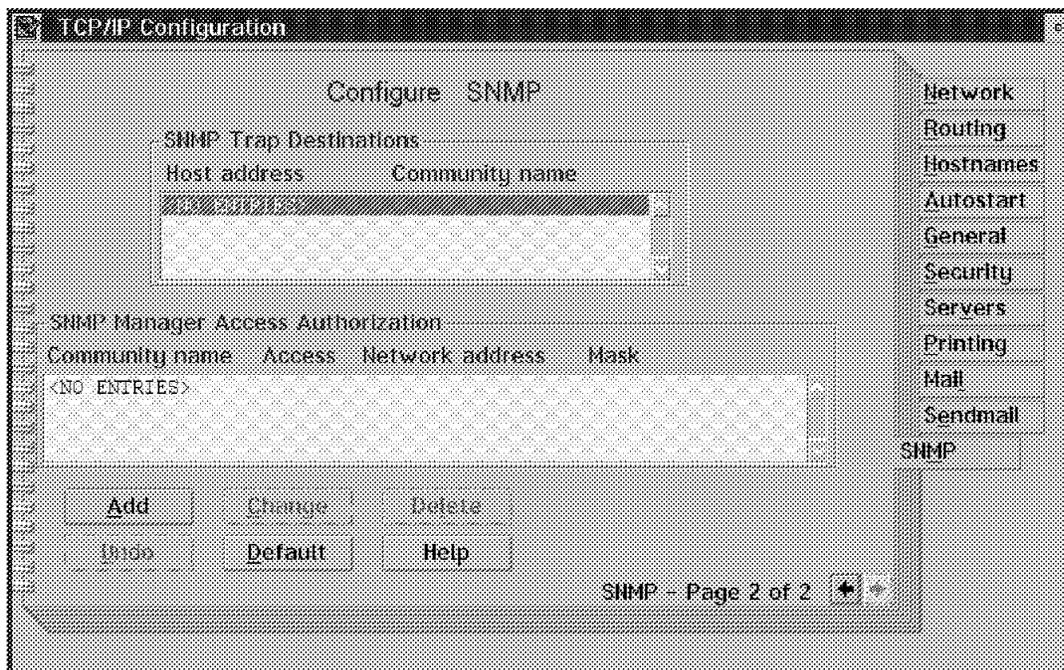


*Figure 161. TCP/IP Configuration - SNMP Page 2*

First you need to add a trap destination. Specify the IP Address or hostname of the machine that is going to receive the traps and the community name.
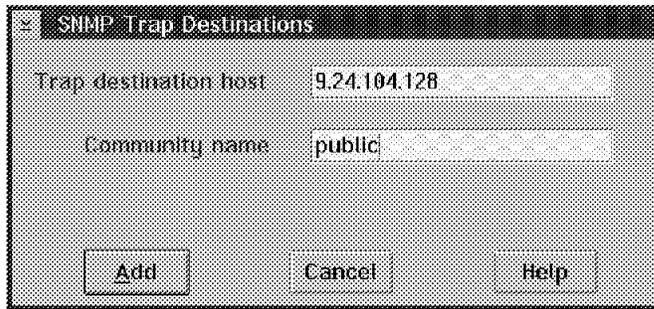
*Figure 162. TCP/IP Configuration - Add Trap Destination*

For every specified community name, there must be an SNMP Manager Access Authorization defined. We didn't restrict the read/write access in our examples.
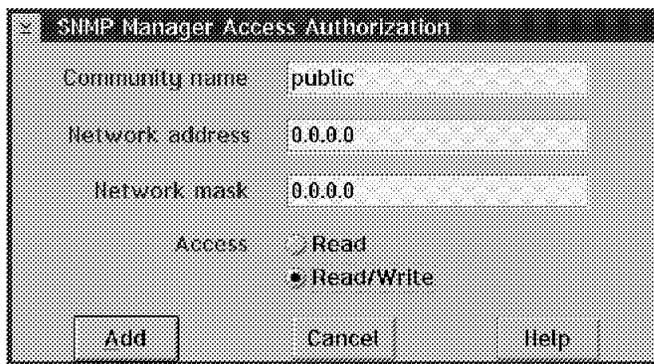


*Figure 163. TCP/IP Configuration - Add SNMP Manager Access Authorization*

After this, the second page of the SNMP customization should look like the following:
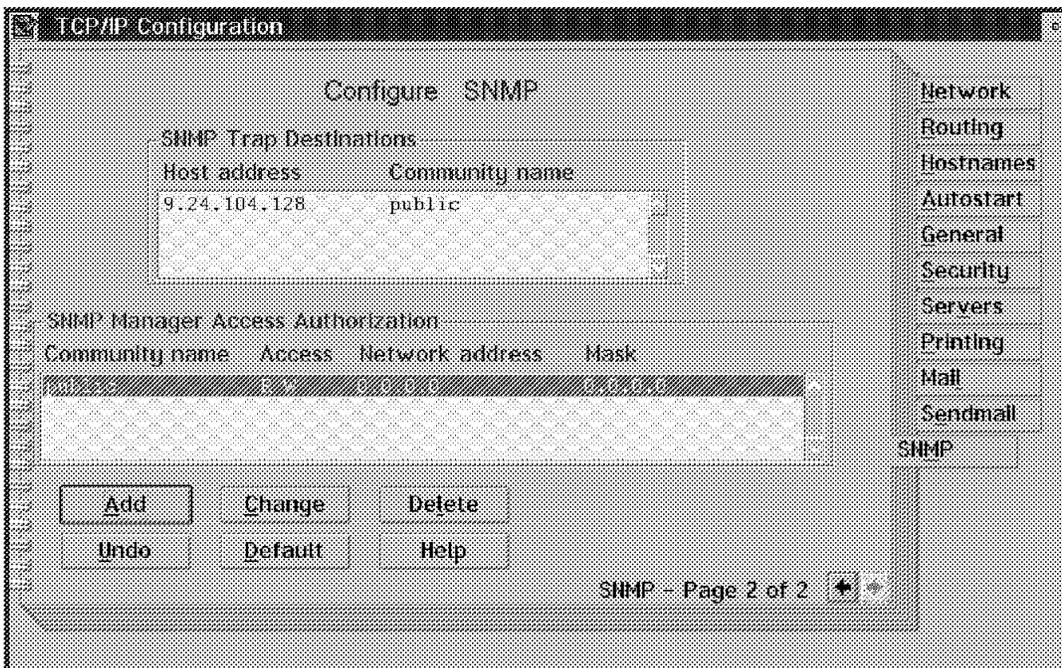


*Figure 164. TCP/IP Configuration - SNMP Page 2 with Definitions*

As indicated earlier, as of TCP/IP Version 3.0 the SNMP daemon SNMPD.EXE no longer includes the complete MIB. You have to start the MIB_2.EXE subagent. There is also a bug in the configuration folder for TCP/IP Version 3.0, which makes it impossible to configure the parameters for SNMPD.EXE correctly. To solve this, add a file x:\tcpip\etc\tcpexit.cmd and start both SNMPD and MIB_2 from there.



```
E.EXE   \tcpip\etc\tcpexit.cmd
File  Edit  Options  Help
start /min snmpd.exe -transport udp -dpi tcp -dpi shm
start /min mib_2.exe
```
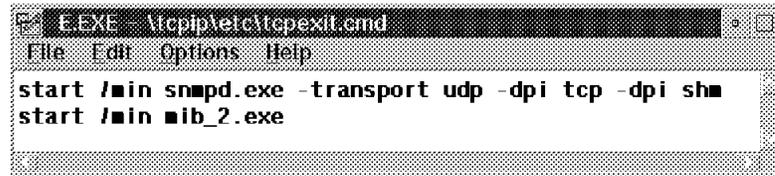
*Figure 165. Start of SNMP and MIB Daemons in x:\TCPIP\ETC\TCPEXIT.CMD*

Restart the computer and SNMP is ready to be used.

## 5.1.3 SNMP Configuration in Windows NT 3.51

SNMP in Windows NT 3.51 is part of the TCP/IP installation. During setup it is one of the selectable options. If you want to customize it later, you can find it in the control panel's network section.
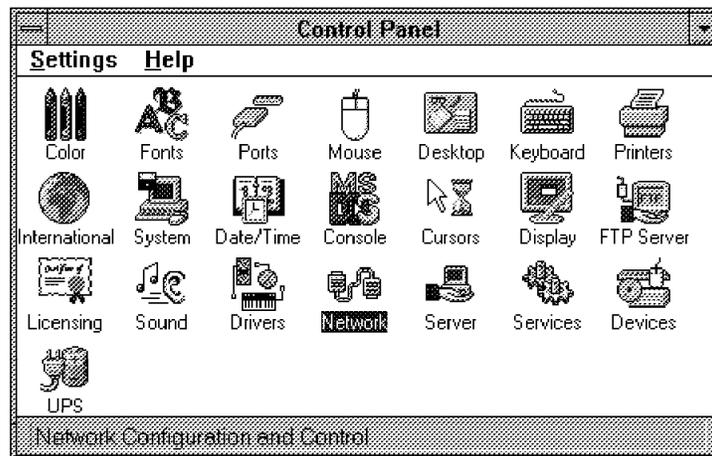


*Figure 166. Control Panel*

To configure SNMP double-click on **SNMP Service** in the Network Services and adapter cards section.
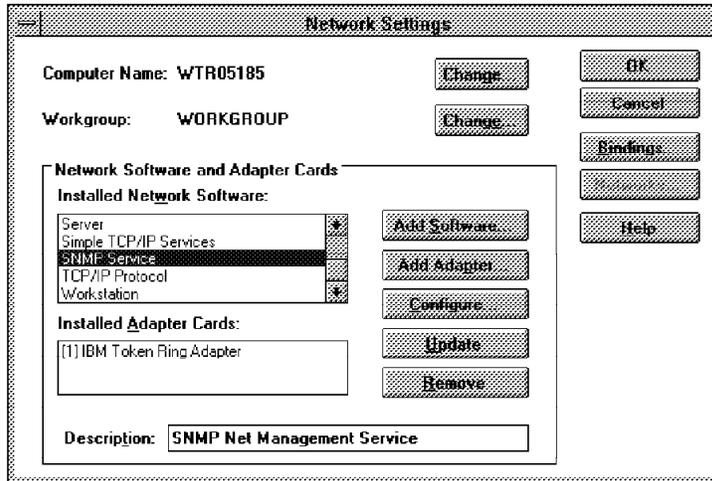
*Figure 167. Network Settings - SNMP*

The first thing to do is to define a community name. Enter it and select **Add**. For each of the community names you defined, you can now specify a list of trap destination hosts. The trap address field will not be usable until you specify a community name.
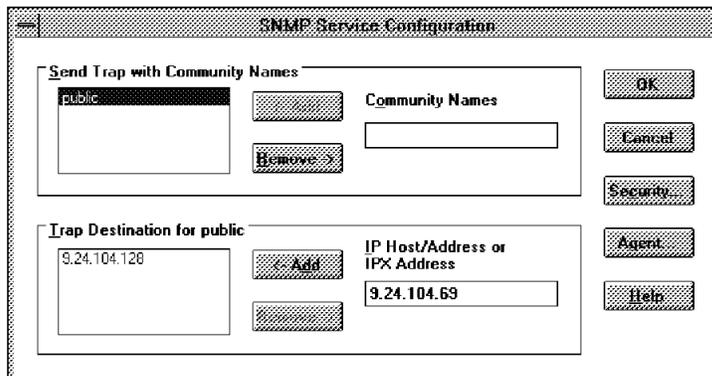


*Figure 168. SNMP Service Configuration*

The community name is a standard part of SNMP V1 security. You can specify the community name for individual hosts. You can also specify that an authentication trap gets sent when there is an unauthorized access attempt.
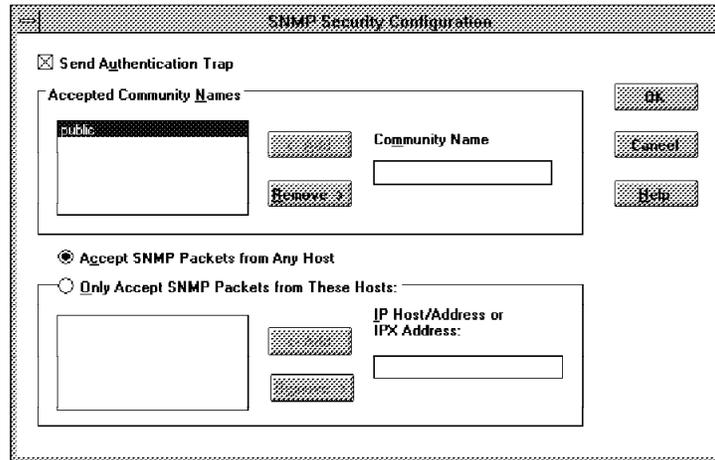
*Figure 169. SNMP Security Configuration*

The SNMP agent information relates to the normal MIB variables for syscontact, syslocation and other system information.
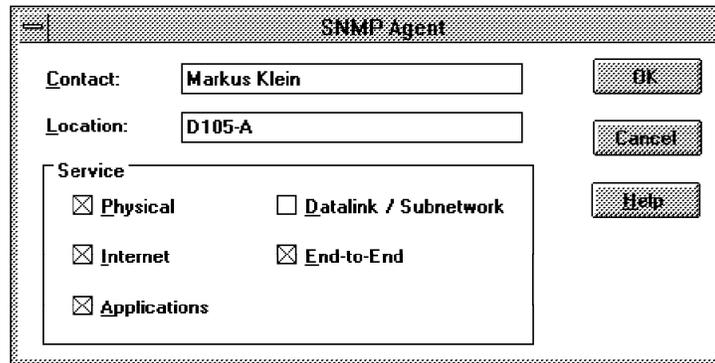


*Figure 170. SNMP Agent*

## 5.2 LMU QueryVPD vs. NetFinity 5.0 System Information

The LMU command QUERYVPD /R and the NetFinity 5.0 command NFSYSICL produce similar but not equal results. The QUERYVPD from LMU gives you only hardware relevant data while the NetFinity 5.0 command returns all system relevant data. The LMU command returns the software relevant data only if you specify the /R option which tells the command to write the output to the LMU manager's database. We found also that there is a major time difference in executing the NetFinity 5.0 commands. NetFinity 5.0 executes much faster than LMU and it can communicate with its client over various protocols while LMU always needs NetBIOS or SPX/IPX. From there you can extract the data either with a command or with a GUI tool such as Visualizer Flight. We show samples for both ways. Entering the command LMUCMD WTR05188 QUERYVPD resulted in the following output.

```
IBM LAN NetView Management Utilities
5622-153 (C) Copyright IBM Corp. 1991, 1996. All rights reserved.

LMU9010I Command sent to target WTR05188; waiting for output
LMU9024I Output:
IBM LAN NetView Management Utilities
5622-153 (C) Copyright IBM Corp. 1991, 1996. All rights reserved.

IBM LAN NetView Management Utilities Maintenance Level LM00240


******************************************************************************
*                            Report Generation                              *
******************************************************************************


     Date of report.................1996-10-23
     Time of report.................10.14.03


******************************************************************************
*                          Operating Environment                            *
******************************************************************************


     Operating System...............OS/2 3.00


******************************************************************************
*                          Hardware Configuration                           *
******************************************************************************


     Machine Type...................IBM PC 350
     Product Number.................6586-7XH
       System Serial Number        23ATAAG
     Processor......................Pentium
       Processor Speed             133 MHz
     CoProcessor....................Pentium
     Bus Type.......................PCI/ISA

     Total Memory...................81524 KB = 79.6 MB

     Equipment List.................1 Parallel Port(s)
                                    2 Serial Port(s)
                                    1 Diskette Drive(s)
                                    1 Fixed Disk(s)
                                    Math CoProcessor

     Serial Port 1..................COM1
       Baud rate                    1200 bps
       Data bits                    7
       Parity                       Even
       Stop bits                    1
     Serial Port 2..................COM2
       Baud rate                    1200 bps
       Data bits                    7
       Parity                       Even
       Stop bits                    1

     Diskette Drive 1...............3.50″  - 1474K - 80 Tracks - Type 4

     Fixed Disk 1...................1.1 GB = 1220.0 MB = 1249920 KB

     Keyboard Type..................101/102 Key Enhanced (ID AB41)
```

```
   Current Display.................PS/2 Color 8512/8513/8515

   Total Slots....................5

   PCI Devices Found..............3

   Bus 0 device 0.................INTEL : Host Bridge
   Bus 0 device 7.................INTEL : ISA Bridge
   Bus 0 device 8.................S3 : 964 Graphics Accelerator

********************************************************************************
*                          LAN Adapter Information                            *
********************************************************************************

   LAN Adapter 0

      Node Address.................400052005188   (02004A008A11)
      Universal Address............08005ACF1BA3   (10005AF3D8C5)

********************************************************************************
*                              Logical Drives                                 *
********************************************************************************

   HPFS    Drive C................ 295.7 of  500.0 MB free ==>  40% full
   LAN     Drive E................   0.4 of    2.1 GB free ==>  80% full

********************************************************************************
*                                User Data                                    *
********************************************************************************

   1)  Assigned_user            [Strauss, Thomas]
   2)  User Serial #            [123456]
   3)  User department #        [6565]
   4)  User Internal Phone #    [555-5555]
   5)  User External Phone #    [555-555-5555]
   6)  Building                 [1111]
   7)  Floor                    [001]
   8)  Location Office #         [XXXX]
   9)  Location Internal Phone # [555-5555]
   10) Location External Phone # [555-555-5555]
   11) Owning_Manager           [Last, First        ]
   12) Owning_Department        [XXX]
   13) T/R Port ID              [XXXXXXXXXX]
   14) Power Outlet             [XXXXXXX]
   15) Equipment     Format: {Machine_Type,Model,Serial_Number,Date_Installed}
   ----------------------------------------------------------------
   [Start of equipment
   {IBM-6586,7XH,23-1234567,01-01-1995}     System Unit
   {IBM-8514,001,00-1097175,01-01-1995}     Display
   {IBM-8513,001,23-CLV46,01-01-1995}       Display
   {IBM-1391401,,4655517,01-01-1995}        Keyboard
   {IBM-90X6778,,1306518,01-01-1995}        Mouse
   {IBM-4216,031,41-8886A,01-01-1995}       Printer
   {IBM-6180,,B1506,01-01-1995}             Plotter
   End of equipment]
```

```
*****************************************************************************
*                          Critical File Information                        *
*****************************************************************************


        C:\CONFIG.SYS
          Location                    C:\
          Date                        1996-10-21
          Time                        13.02.28
          Size                        5405 bytes

        C:\STARTUP.CMD
          Location                    C:\
          Date                        1996-10-15
          Time                        17.34.58
          Size                        68 bytes

        C:\IBMCOM\PROTOCOL.INI
          Location                    C:\IBMCOM\
          Date                        1996-10-15
          Time                        15.22.00
          Size                        1609 bytes

        C:\IBMLAN\IBMLAN.INI
          Location                    C:\IBMLAN\
          Date                        1996-09-20
          Time                        11.04.32
          Size                        13681 bytes



    Return code = 0
    LMU9021I End-of-data
```

As you can see there is no information for the installed software on this system.
The data regarding critical file and user vpd are extracted from the uservpd.dat
and the critfile.def files on the client. As we indicated before, you have to use
the /R or /D option to get that information. The command LMUCMD WTR05188
QUERYVPD /R writes the data to the LMU database. The machine name holding
the database is defined in the LMU.CTL file. Alternatively you can give the name
of the managing system that contains the databases with the /R parameter such
as /RWTR05095 where WTR05095 is the NetBIOS name. Following is the
command dialog.

```
[D:\ITSO]LMUCMD WTR05188 QUERYVPD /R
IBM LAN NetView Management Utilities
5622-153 (C) Copyright IBM Corp. 1991, 1996. All rights reserved.

LMU9010I Command sent to target WTR05188; waiting for output
LMU9024I Output:
IBM LAN NetView Management Utilities
5622-153 (C) Copyright IBM Corp. 1991, 1996. All rights reserved.

IBM LAN NetView Management Utilities Maintenance Level LM00240

LMU6011I Transport of data to managing system with database successful


Return code = 0
LMU9021I End-of-data

[D:\ITSO]
```

Now the data is in the database and if we want to see it, we have to extract it
with one of the following examples. First using the DB2/2 command line, you
have to connect to the database by issuing the connect command.

```
[D:\ITSO]db2 connect to lmu2

   Database Connection Information

 Database product      = DB2/2 2.1.1
 SQL authorization ID   = BLSTRAS
 Local database alias   = LMU2


[D:\ITSO]
```

Next you have to select the data you want to see. LMU catalogs four views in
the database: SOFTWARE, CONFIG, CONFIGURATION and HARDWARE. For our
example we needed the SOFTWARE view and the statement to get to it follows:

```
[D:\ITSO]db2 SELECT * FROM SOFTWARE WHERE
(SOFTWARE."COMPUTER" = 'WTR05188' > wtr05188.vpd

[D:\ITSO]
```

We redirected the output to a file named wtr05188.vpd.

| COMPUTER | FILE_EXTENSION | DRIVE | PATH | COMPONENT_ID | PROGRAM_NAME | VERSION | CURRENT_CSD | PREVIOUS_CSD |
|---|---|---|---|---|---|---|---|---|
| WTR05188 | UML | C | \TCPIP\UMAIL | 562260112 | IBM Ultimedia Mail Lite Client for OS/2 | 2.10 | UN00000 | UN00000 |
| WTR05188 | TCP | C | \TCPIP\BIN | 562281300 | IBM TCP/IP Version 3.0 for OS/2 | 3.00 | IC00000 | IC00000 |
| WTR05188 | PCS | C | \PCOMOS2 | 562276201 | IBM Personal Communications AS/400 and 3270 | 4.10 | WR08000 | WR00000 |
| WTR05188 | MUG | C | \MUGLIB | 562246104 | IBM OS/2 User Profile Management | 4.06 | WR08000 | WR00000 |
| WTR05188 | DHC | C | \MPTN | 562290900 | IBM OS/2 Dynamic Host Configuration Protocol Services | 3.10 | WR08200 | WR08000 |
| WTR05188 | GRE | C | \OS2\INSTALL | 562274700 | IBM OS/2 32-bit Graphics Engine | 3.00 | XR03003 | XR03003 |
| WTR05188 | OS2 | C | \OS2\INSTALL | 562274700 | IBM OS/2 Base Operating System | 3.00 | XR03003 | XR03003 |
| WTR05188 | SDS | C | \OS2\INSTALL | 96F8647DS | Distributed SOM Framework | 2.01.1 | SM20004 | SM20003 |
| WTR05188 | SEM | C | \OS2\INSTALL | 96F8647EM | SOM Event Management Framework | 2.01.1 | SM20004 | SM20003 |
| WTR05188 | SIR | C | \OS2\INSTALL | 96F86471R | SOMobjects Interface Repository Framework | 2.01.1 | SM20004 | SM20003 |
| WTR05188 | SRK | C | \OS2\INSTALL | 96F8647RK | SOM Run-time Kernel | 2.01.1 | SM20004 | SM20003 |
| WTR05188 | SUT | C | \OS2\INSTALL | 96F8647UT | SOMobjects Utility Classes | 2.01.1 | SM20004 | SM20003 |
| WTR05188 | VID | C | \OS2\INSTALL | 562107701 | S3 Trio64/64V+ | 2.80 | S328019 | S328019 |
| WTR05188 | MNF | C | \NETFIN | 564202000 | TME 10 NetFinity Manager for OS/2 | 4.00.1 | NM00401 | NM00401 |
| WTR05188 | SNF | C | \NETFIN | 564201000 | TME 10 NetFinity Services for OS/2 | 4.00.1 | NS00401 | NS00401 |
| WTR05188 | UPE | C | \MUGLIB | 562246105 | IBM OS/2 User Profile Management - Extended | 4.06 | IP08000 | IP08000 |
| WTR05188 | MPT | C | \MPTN | 562290900 | IBM OS/2 Socket/Multi-Protocol Transport Services | 5.00 | WR08200 | WR08000 |
| WTR05188 | MPM | C | \MMOS2\INSTALL | 562274700 | IBM Multimedia Presentation Manager/2 | 3.00 | XR03003 | XR03003 |
| WTR05188 | LMU | C | \LMU2 | 562215301 | IBM LAN NetView Management Utilities for OS/2 | 1.21 | LM00240 | LM00240 |
| WTR05188 | PER | C | \IBMLAN | 569653000 | IBM Peer for OS/2 | 1.00 | IP02000 | IP02000 |
| WTR05188 | REQ | C | \IBMLAN | 569653000 | IBM OS/2 LAN Requester | 4.06 | IP08000 | IP08000 |
| WTR05188 | I18 | C | \IBMI18N | 562290900 | IBM I18N Toolkit | 1.02 | WR08200 | WR08200 |
| WTR05188 | TRP | C | \IBMCOM | 562290900 | IBM OS/2 LAN Adapter and Protocol Support | 5.00 | WR08200 | WR08000 |
| WTR05188 | WCB | C | \GRPWARE | 562267200 | OS/2 WARP Connect with WIN-OS2 | 3.00 | IP08000 | IP08000 |

24 record(s) selected.

The following screens show how to get the data with Visualizer Flight.
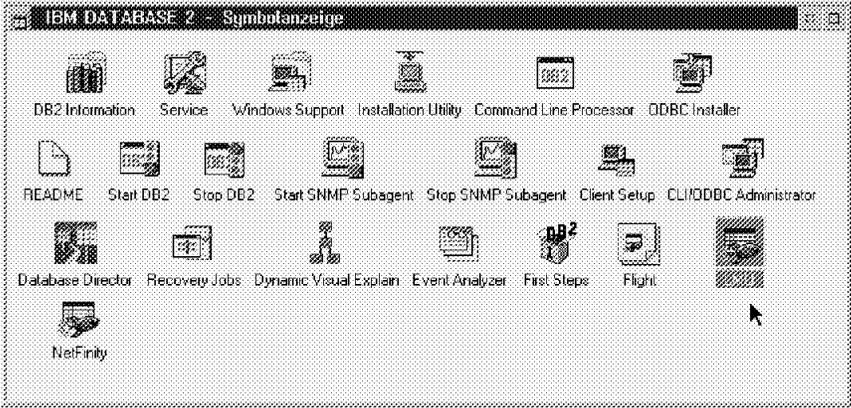


*Figure 171. IBM Database/2*

Drag a copy of the object from the Flight Template to another location. We called our copy of it LMU. Double-click on the new object to get to the Visualizer Flight Query window.



*Figure 172. Visualizer Flight Query Pilot - DB Selection*

Click on **LMU2** to select the LMU database. Then click on **Select Tables** in the tool bar at the bottom.
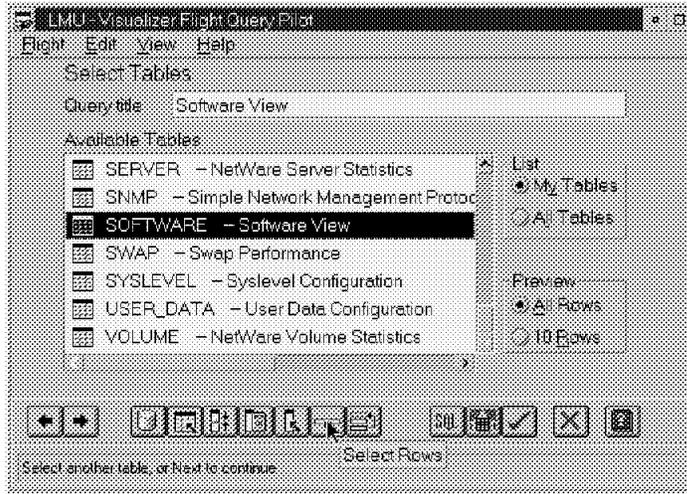
*Figure 173. Visualizer Flight Query Pilot - Table Selection*

Enter a name for your query. Then scroll the Available Tables list down and select **SOFTWARE - Software View**. Click on **Select Rows** in the tool bar at the bottom.
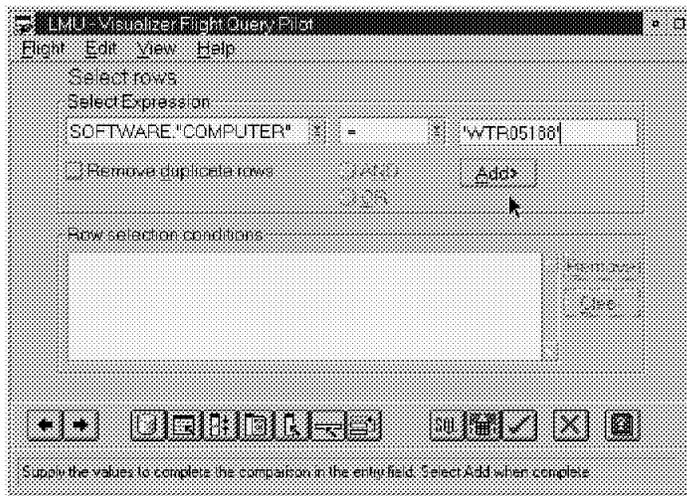


*Figure 174. Visualizer Flight Query Pilot - Select Rows*

Select **SOFTWARE.″COMPUTER″** in the left part of the expression. Select **=** as the compare operand and enter the name of your workstation in the right-most field. This field is case-sensitive and you have to put the name in quotes. Then click on **Add**.
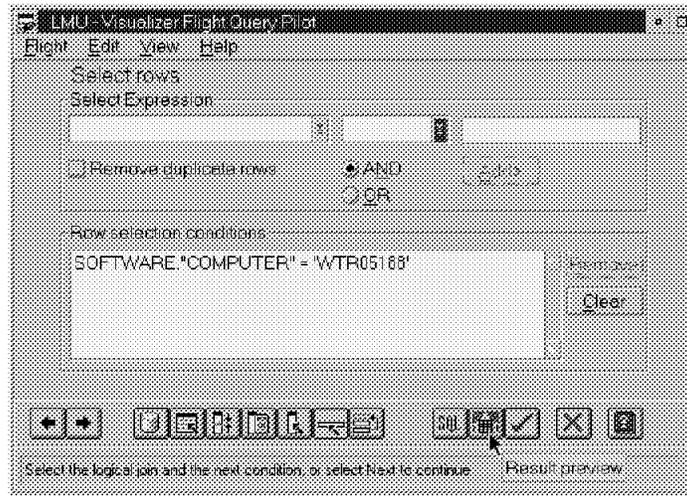
*Figure 175. Visualizer Flight Query Pilot - Run SQL*

Click on the **Result Preview** button in the tool bar. You also can click on **SQL** to
see the SQL statement you just created (see Figure 177 on page 188). If you do
this, you have to click on the **Play** button in the tool bar to run the SQL
statement. The result looks like the following:



*Figure 176. Visualizer Flight Query Pilot - Result Table*

*Figure 177. Visualizer Flight Query Pilot - SQL User*

The NetFinity 5.0 command NFSYSICL returns all the data with one command and displays the data on the screen. If you reroute the output, the data is written into a file or to the printer. Also you can write the data to a database or generate a history file which can be viewed with the SINFG30 command. If your history file is named wtr05188.hst, enter the command SINFG30 /F:wtr05188.hst.

In order for the nfsysicl command to work on your local ID, you must make sure that you have set up the security access rights in the incoming password service for it. Once we added System Information to the list of services that public can use, we entered the following command: nfsysicl /N:netbios::WTR05188 /rpt:wtr05188.vpd. The result of the command is written to a file named wtr05188.vpd on the manager's local disk. A subset of the file's output follows:

```
*********** System Information Tool -  ***********
*********** Adapter Information ***********
Expansion Slot Layout
                    Slot  5 : Undetectable
                    Slot  4 : Undetectable
                    Slot  3 : Undetectable
                    Slot  2 : Undetectable
                    Slot  1 : Undetectable
 (Back of computer)
System Board Devices
PCI Interface Level : 2.10
Number of last PCI bus in system : 0
PCI Hardware Characteristics
Configuration Space Access Mechanism 1 supported
                    Host/PCI bridge
                    PCI/ISA bridge
                    Mass storage controller
                    VGA-compatible controller
*********** Host/PCI bridge ***********
Manufacturer                  : INTEL CORPORATION
Bridge device                 : Host/PCI bridge
Class Code                    : 0006
PCI Bus Number 0              : Device Number 00
Vendor ID                     : 8086
Device ID                     : 122D
Revision ID                   : 0002
Cache Line Size               : 0000
Latency Timer                 : 0040
Min_Gnt                       : 0000
Max_Lat                       : 0000
Interrupt Line                : 0000
Interrupt Pin                 : 0000
Expansion ROM Base Address    : 0000
Header Type                   : 0000
BIST                          : 0000
Command Register
Command Register              : 0006
PCI I/O Access                : Disabled
PCI Memory Access             : Enabled
PCI Bus Master                : Enabled
Monitor PCI Special Cycles    : Disabled
Memory Write and Invalidate   : Disabled
Palette Snoop                 : Disabled
Parity Error Response         : Disabled
Wait Cycle                    : Disabled
System Error                  : Disabled
Fast Back-to-Back Transactions : Disabled
Status Register
Status Register               : 2200
Device Select Timing (DEVSEL#) : Medium
Received Master Abort
*********** System Information Tool -  ***********
```

*Figure 178. NFSYSICL Output*

The data that we showed is only a small subset of what you will see when you
issue the command. The actual output was over 800 lines long, so we only
included the first 50 lines.

If the output is sent to a database, the data is stored in different tables in the
NFDB database. You can use the Visualizer Flight, any SQL tool or the
command line interface to DB2/2. It is most difficult to use the command line
interface to DB2/2 since there are a lot of fields in different tables. It's easier to
use a GUI tool. There is also an online document for the database tables.

## 5.3  SNMP Traps and Topology

This chapter compares the trap handling of LMU's LMUSNMPD.EXE with the SNMP trap alert action of NetFinity 5.0.

### 5.3.1  MIB

The MIB files for NetFinity 5.0 were located on NetWare Diskette 1. We copied the following four MIB files:

- NETFIN.MIB
- NETFINMS.MIB order to
- NETFINSV.MIB
- SYSTEM.MIB

We placed them into TME 10 NetView V4.1.2's /usr/OV/snmp_mib directory and used the xnmloadmib command to load them all into TME 10 NetView.

In order to prepare NetFinity 5.0 for SNMP support we had to run an initialization routine that was documented in the README file. We first entered SINFB30 /DMIALL. Following that we made sure that SNMPD was executing and we started the NetFinity 5.0 SNMP routine NETFSA.EXE.

Figure 179 on page 191 shows the loading of the MIB in TME 10 NetView for AIX and how to display some of the actual values.

From the Load MIB From File window, you should load the MIBs:

- NETFIN.MIB
- NETFINMS.MIB
- NETFINSV.MIB
- SYSTEM.MIB

You will have to load them one at a time. You can also load them using a command shell instead of waiting for the GUI to finish.
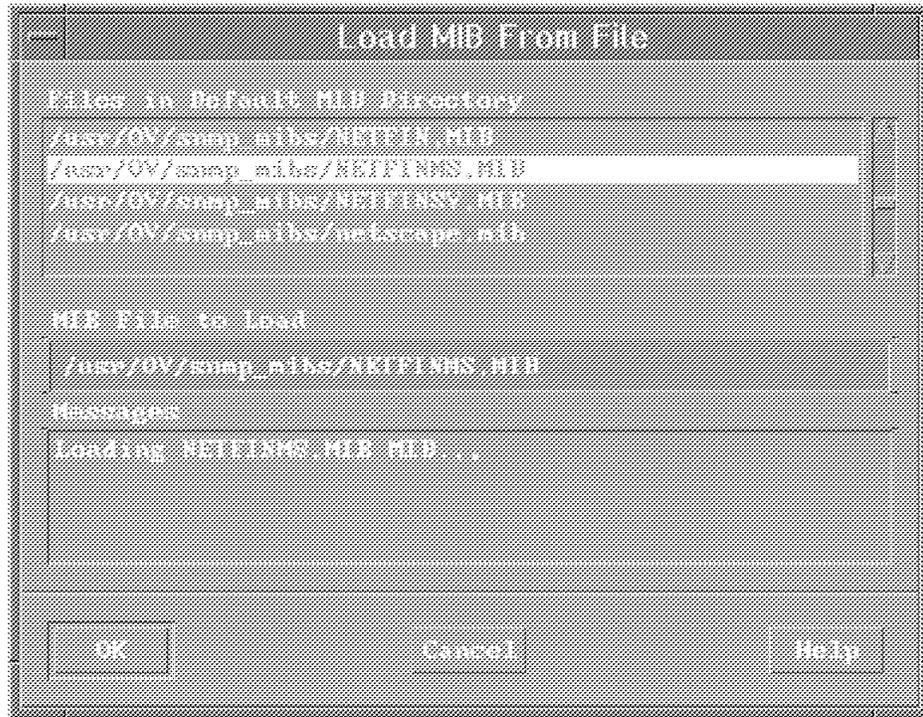
*Figure 179. TME 10 NetView AIX - MIB Browser*

## 5.3.2 Setting Up SNMP Trap Handling and Forwarding

The installation and customization of SNMP itself was shown in 5.1, "SNMP Configuration" on page 171. Let's now look at the settings that need to be done in LMU and NetFinity 5.0.

### 5.3.2.1 LMU

In LMU some initial customization needs to be done. The following excerpts from the LMU.CTL show the keys where machine names or addresses have to be inserted.

```
⋮
APP(LMU_UTILITY),
    KEY(SNMP_PROXY_AGENT),
    ASCIIZ(WTRO5144);
⋮
APP(LMU_UTILITY),
    KEY(PROXY_QUERY_MANAGING_SYSTEMS),
    ASCIIZ(WTRO5144);
⋮
```

The first variable indicates the machine where LMUSNMPD is running. The second defines which machines are to be queried at the startup of LMUSNMPD. After these definitions are done, run LMUCUST with either PROXY or PROXY_DB as one of the options. If everything is set up correctly, the LMU SNMP Proxy Agent will start after the LMUCLI. Traps flow from LMU Fault Manager to the LMU Proxy Agent to the TCP/IP Proxy Agent to the trap destination set in the TCP/IP configuration.

> **Important**
>
> If you are using TCP/IP for OS/2 Version 3.x or higher, don't forget to start the MIB_2 subagent.

To test the connection you can create an LMU Event and send it to the LMU SNMP Proxy Agent.

```
LMUEVENT /p9.24.104.100 WTR05144 SNMPTrap
```

The output on the SNMP Trap Monitor or in the SNMPTRAP.LOG shows some information about the error.

```
                Display of SNMP trap, from 9.24.104.100 port 161

community:      public ('7075626c6963' h)
enterprise oid: 1.3.6.1.4.1.2.6.14
agentAddress:   9.24.104.100 ('09186864' h)
generic-trap:   enterpriseSpecific ('00000006' h)
specific-trap:  3 ('00000003' h)
time-stamp:     244500 - 40 minutes, 45.0 seconds
ibm.6.14.4.2.1.1.2 : 2
ibm.6.14.4.2.1.2.2 : '57545233035313434' h          <- WTR05144
ibm.6.14.4.2.1.3.2 : 2
ibm.6.14.4.2.1.4.2 : 2
ibm.6.14.4.2.1.5.2 : '34303030355323003035313434' h  <- 400052005144
ibm.6.14.4.2.1.6.2 : '30313035' h                    <- 0105
ibm.6.14.4.2.1.7.2 : '575452444d' h                  <- WTRDM
ibm.6.14.4.2.1.8.2 : '4d414b4c45494e' h              <- MAKLEIN
ibm.6.14.4.2.1.9.2 : '57545233035313434' h           <- WTR05144
ibm.6.14.4.2.1.10.2 : 1
ibm.6.14.4.2.1.11.2 : '' h
```

The following screen shows the corresponding part of the LMU MIB.

```
lmuTopologyIndex            1.3.6.1.4.1.2.6.14.4.2.1.1.     number
lmuTopologyName             1.3.6.1.4.1.2.6.14.4.2.1.2.     display
lmuTopologyType             1.3.6.1.4.1.2.6.14.4.2.1.3.     number
lmuTopologyStatus           1.3.6.1.4.1.2.6.14.4.2.1.4.     number
lmuTopologyNodeAddress      1.3.6.1.4.1.2.6.14.4.2.1.5.     display
lmuTopologyNodeType         1.3.6.1.4.1.2.6.14.4.2.1.6.     display
lmuTopologyDomain           1.3.6.1.4.1.2.6.14.4.2.1.7.     display
lmuTopologyUserid           1.3.6.1.4.1.2.6.14.4.2.1.8.     display
lmuTopologyManagingSystem   1.3.6.1.4.1.2.6.14.4.2.1.9.     display
lmuTopologyManage           1.3.6.1.4.1.2.6.14.4.2.1.10.    number
lmuTopologySymbolicName     1.3.6.1.4.1.2.6.14.4.2.1.11.    display
```

### 5.3.2.2  NetFinity 5.0

In NetFinity 5.0, SNMP support is automatically set up if SNMP is installed.

> **Important**
>
> If you use the IBM SystemView Agent for OS/2 to get SNMP, you have to copy DPI32DLL.DLL from the x:\SVA\DPI\DPI_1.1 directory to x:\TCPIP\DLL. This is the DLL NetFinity 5.0 is looking for to determine whether SNMP is installed. You do not have to restart OS/2. You will only have to restart NetFinity 5.0.

There are two different alert actions in the NetFinity 5.0 Alert Manager that use SNMP:

1. Send SNMP Alert through IBM TCP/IP

2. Map Alert to SNMP trap

The following screens and SNMP traps are examples of those two alert actions.
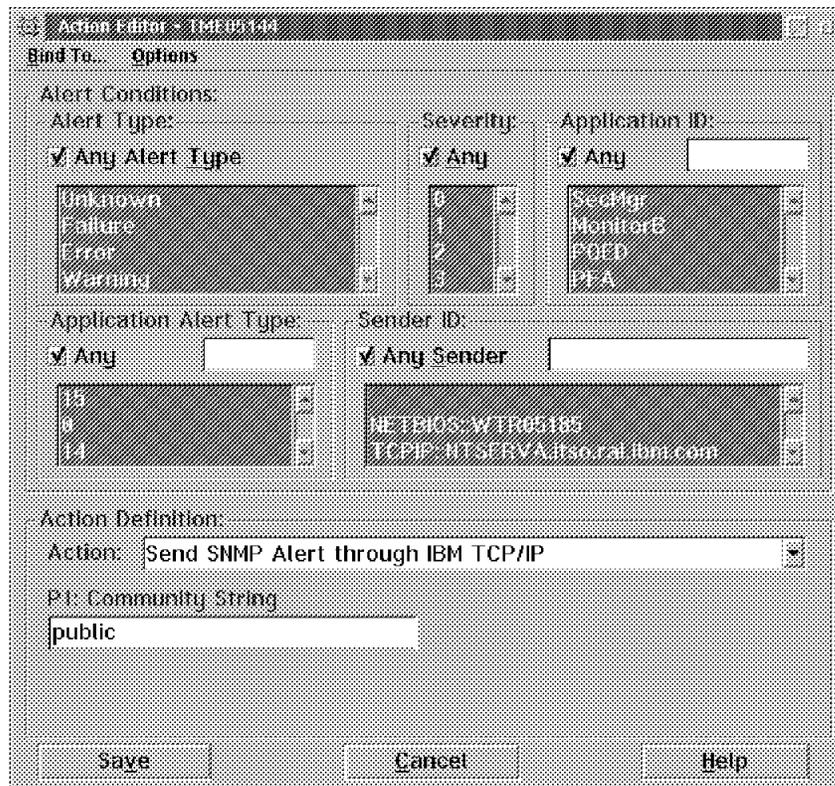


*Figure 180. NetFinity 5.0 Alert Action*

If an error occurs (access granted warning in this case), the SNMP Trap Monitor shows the following output.

```
               Display of SNMP trap, from 9.24.104.100 port 161

community:       public ('7075626c6963' h)
enterprise oid: 1.3.6.1.4.1.2.6.71
agentAddress:    9.24.104.100 ('09186864'h)
generic-trap:    enterpriseSpecific ('00000006'h)
specific-trap:   1 ('00000001'h)
time-stamp:      1456200 - 4 hours, 2 minutes, 42.0 seconds
ibm.6.71.1.0 : '5573657220494420274d414b4c45494e272066726f6d2041...'
ibm.6.71.2.0 : '544d453035313434' h
ibm.6.71.3.0 : '30353a32303a333670' h
ibm.6.71.4.0 : '31302d32322d31393936' h
ibm.6.71.5.0 : 7
ibm.6.71.6.0 : '5365637572697479920496e666f726d6174696f6e' h
ibm.6.71.7.0 : '5365634d67722020' h
ibm.6.71.8.0 : 20
ibm.6.71.9.0 : '' h
```

To interpret the entries, see the NetFinity 5.0 MIB.

Here is the example for the mapping alert action.
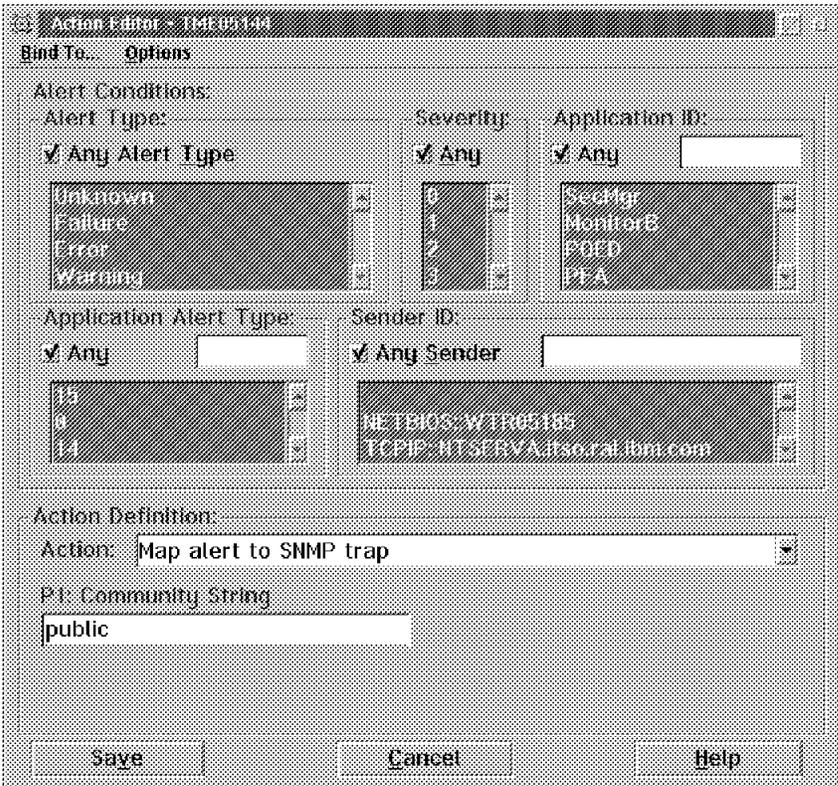


*Figure 181. NetFinity 5.0 Alert Action*

The SNMP trap in the trap monitor looks like this.

```
                    Display of SNMP trap, from 9.24.104.100 port 161

community:      public ('7075626c6963' h)
enterprise oid: 1.3.6.1.4.1.2.6.71.100.12
agentAddress:   9.24.104.100 ('09186864'h)
generic-trap:   enterpriseSpecific ('00000006'h)
specific-trap:  1 ('00000001'h)
time-stamp:     1473100 - 4 hours, 5 minutes, 31.0 seconds
ibm.6.71.1.0 : '55736572204944420274d414b4c45494e272066726f6d2041...'
ibm.6.71.2.0 : '544d453035313434' h
ibm.6.71.3.0 : '30353a32333a323670' h
ibm.6.71.4.0 : '31302d32322d31393936' h
ibm.6.71.5.0 : 7
ibm.6.71.6.0 : '536563757269747920496e666f726d6174696f6e' h
ibm.6.71.7.0 : '5365634d67722020' h
ibm.6.71.8.0 : 20
ibm.6.71.9.0 : '' h
```

## 5.3.3  Topology Information

The current version of LMU can communicate with LMU for AIX.  One of the
functions that this enables is for topology information to be seen at TME 10
NetView for AIX.  There are two possible substitutions for this function.  From
any Web browser, you can use the Webability function to view the Remote
System Manager and see anything that the NetFinity 5.0 end user can see.  In
addition, from any MIB browser you can perform snmpgets and view the
topology information.  If you use one of TME 10 NetView for AIX's APIs, you can
then import that information into its database.  An extract from the MIB follows:

```
SRemoteSystems ::= SEQUENCE {
    a11SystemTag                        DmiInteger,
    a11SystemName                       DmiDisplaystring,
    a11ProtocolName                     DmiDisplaystring,
    a11NetworkAddress                   DmiDisplaystring,
    a11SystemState                      DmiInteger,
    a11Server                           DmiInteger,
    a11Manager                          DmiInteger,
    a11OperatingSystemType              DmiInteger,
    a11OsMajorVersion                   DmiInteger,
    a11OsMinorVersion                   DmiInteger,
    a11SystemModelId                    DmiOctetstring,
    a11SystemModelName                  DmiDisplaystring,
    a11SystemOn-lineNotify              DmiInteger,
    a11SystemOff-lineNotify             DmiInteger,
    a11PresenceCheckInterval            DmiInteger,
    a11MacAddress                       DmiOctetstring
}
```

Using the following command you can get a list of all of the systems:

snmpwalk -c public barryps2 .1.3.6.1.4.1.2.6.71.200.3.1.11.1.2

You can then write a small shell script to extract the system that you are
working with and issue rexec commands against those systems, or use a Web
browser to issue commands.  Assuming you know the rexec user ID and
password, an example of a command issued from TME 10 NetView for AIX might
be:

rexec -n barryps2 nfproccl /getproc

All of the output flows back to your aixterm window. With this capability, you don't need to worry about having an OS/2 system in your network as you did with LMU. This makes the management more platform-independent.

## 5.4 Monitoring Processes and System Performance

LMU provides several tools to monitor the system performance or to watch applications from the command line. The new command line interface of NetFinity 5.0 helps you to provide system performance monitoring as well. In addition, there is a new plug-in module for SPM/2 that works with NetFinity 5.0.

Contained in this package are two main components: the SPM/2 data collector/monitor (NfSpmMon) and the centralized database manager (NfSpmBas). NfSpmMon is a port of LMU's PERFCAP2 utility into the NetFinity environment. This allows customers to continue using their existing PERFCAP2 profiles, while making use of NetFinity alerts, alert actions, and transport mechanisms. Like PERFCAP2, NfSpmMon starts the SPM/2 collection facility on the local system to collect OS/2 system performance data from either local or remote SPM/2 clients. As data is collected and analyzed, alerts are generated and/or database records are created and transferred to a centralized DB/2 database. NfSpmBas creates and manages a central database for all NfSpmMon clients (monitors). Because the NfSpmBas database is modeled after the PERFCAP2 portion of the LMU database, preexisting database queries can be easily ported to this new environment.

The prerequisites for this package are NetFinity 5.0 and SPM/2 V2.01. DB/2 2 is optionally required for customers wishing to centralize OS/2 system performance data in a database.

SPM/2 information can now be integrated into NetFinity 5.0. To take advantage of this you will need to have SPM/2 already installed. In the following example, we show how to capture some information from SPM/2 and also how to create some alerts based upon conditions that SPM/2 is monitoring. Some of the resources that SPM/2 can monitor are:

- CPU utilization
- Disk drives
- IBM LAN Requester
- Memory
- Printers

## 5.4.1 SPM/2 GUI

The following section shows how to use the old SPM/2 go capture data. In 5.4.2, "NetFinity SPM Monitor" on page 201 we show the new interface. You would not run the SPM/2 GUI and the NFSPMMON interface at the same time.
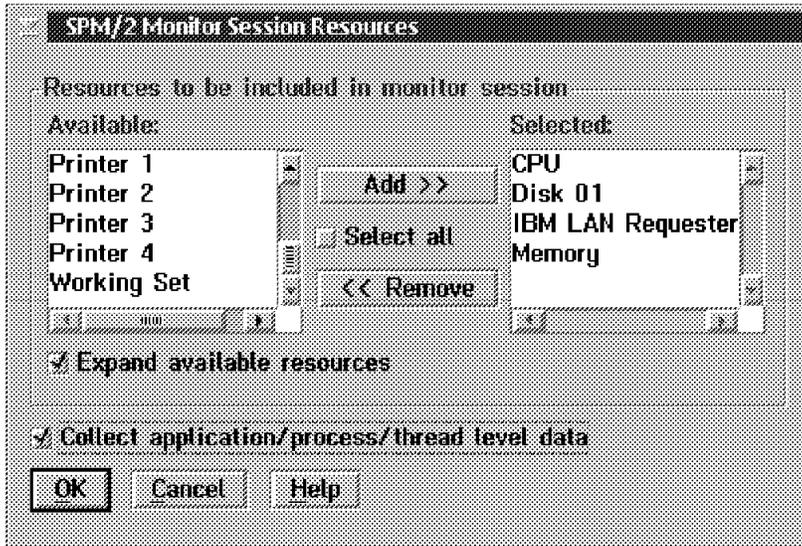
*Figure 182. SPM/2 Resources to Monitor*

You can capture SPM/2 information for individual workstations or you can capture it for many different workstations. You will need a NetBIOS connection in order to see the other workstations. In this example, we installed SPM/2 locally on BOWD1271 and captured information about it. By selecting the **Query** button we can do a NetBIOS broadcast and discover other SPM/2-enabled workstations.
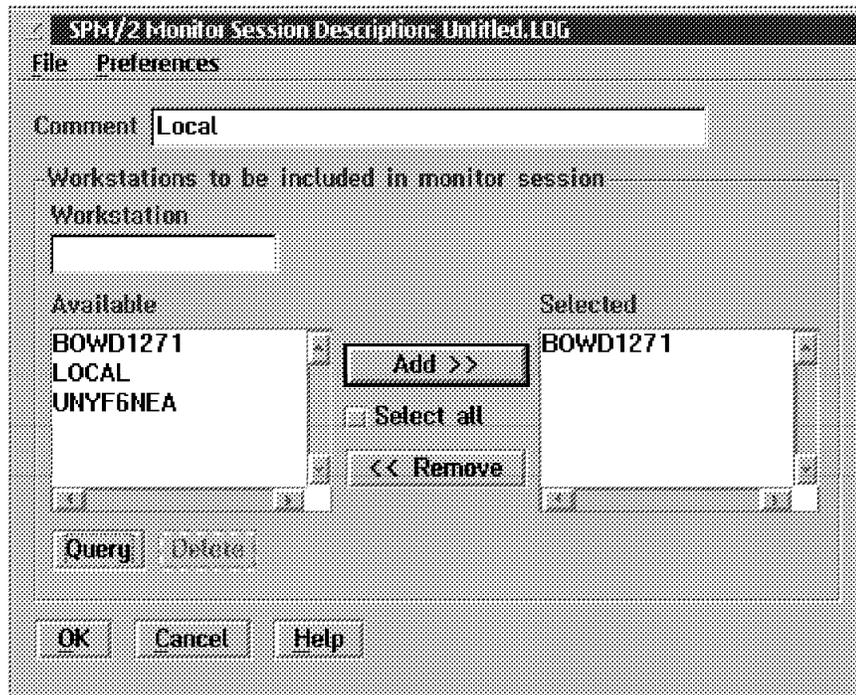


*Figure 183. NetBIOS Workstations Discovered*

So that you don't overload your system with monitoring information, you should specify a collection frequency that is not too small.
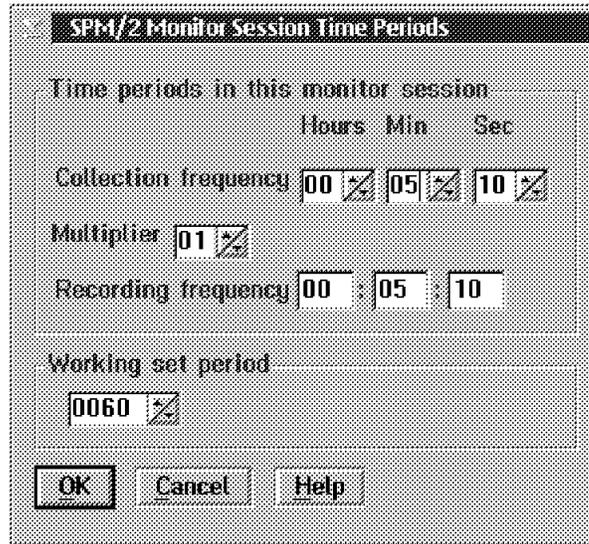


*Figure 184. Collection Frequency*

After setting up SPM/2 you will need to indicate where the information will be logged.
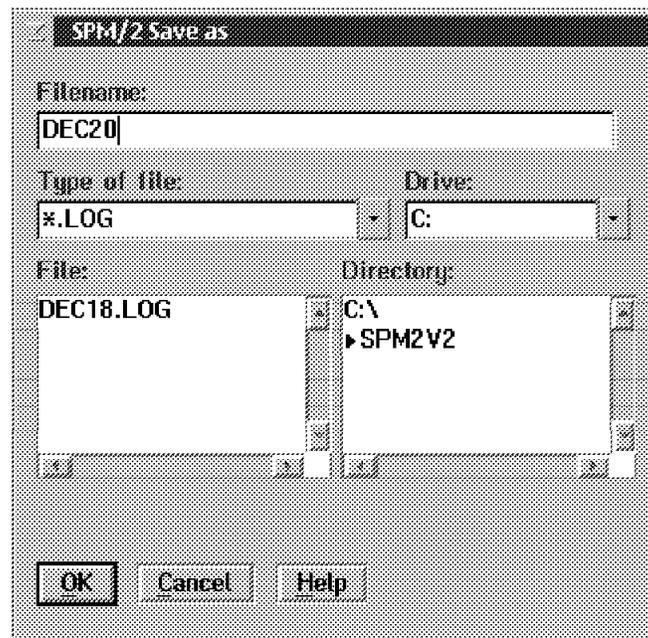


*Figure 185. Log Location*

You can graph (real-time) the information that is being collected. In addition, you may choose to record the information for later analysis.
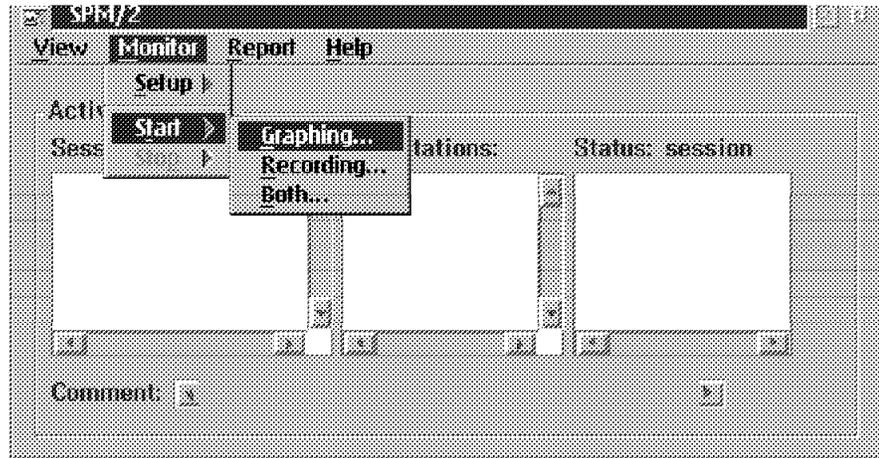
Figure 186. Graphing and Recording of SPM/2 Data

After indicating that you wish to record the data, you have to indicate where it will be logged.
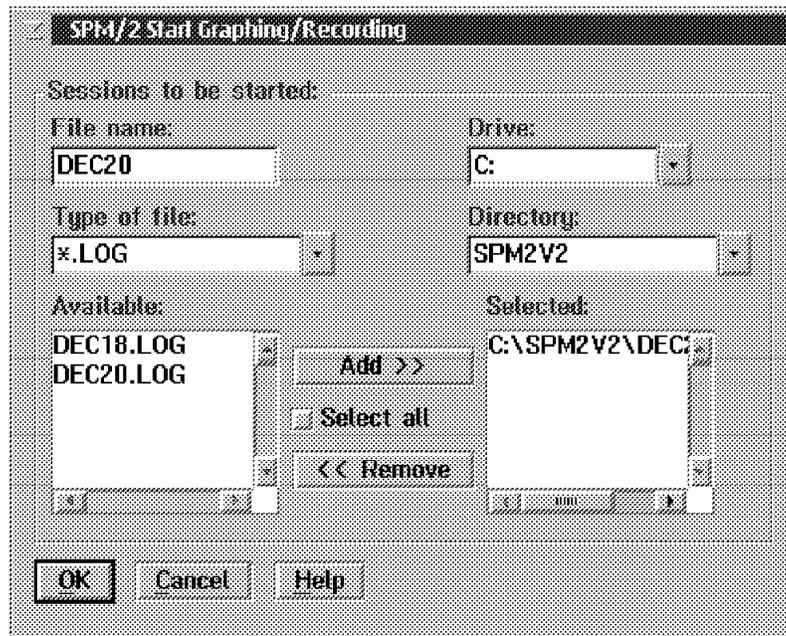


Figure 187. Start Recording
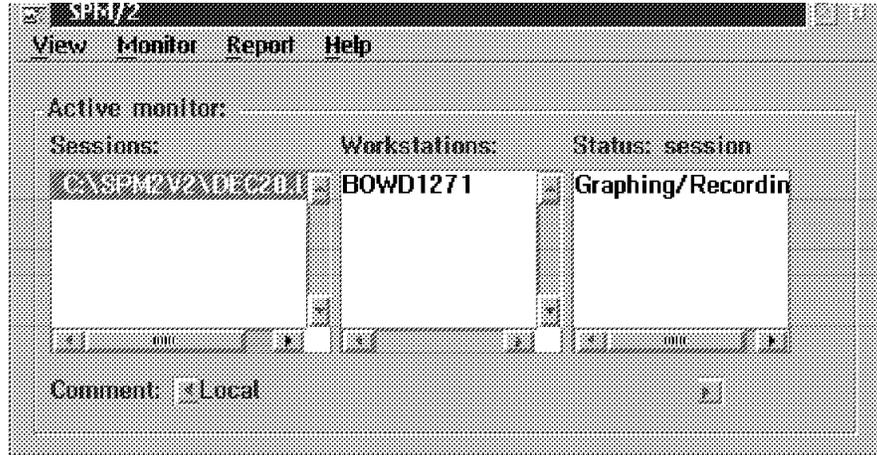
Figure 188. Status of Graphing/Recording

You can generate some reports after you have recorded some of the data.
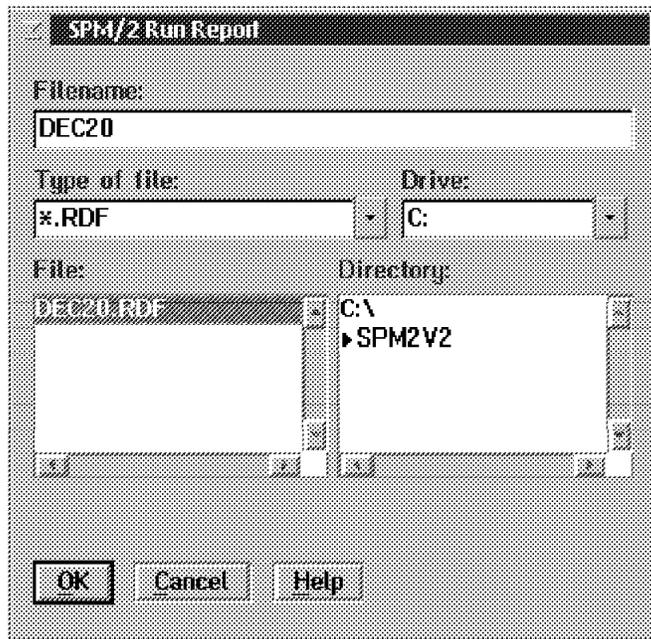


Figure 189. SPM/2 Reports

*Figure 190. NetFinity 5.0 Report Created*

## 5.4.2 NetFinity SPM Monitor

There is a new executable that comes with NetFinity 5.0 called NFSPMMON.EXE. This program will interface with the SPM/2 application. An example of the control file that it uses follows:

```
 HDR local c:\netfin\spm.log c:\netfin\spm.slg 5 5 5 2000 0 - y
#
CPU * 10
#
PID * 5 1
#
DSK 1 5
#
MEM * 5 5 5 5 5
#
```

*Figure 191. NetFinity 5.0 SPM/2 Configuration File*

After you start NFSPMMON.EXE you will see a new window open up that looks similar to the following:

```
NFSPMMON.EXE
IBM
(C) Copyright IBM Corp. 1996. All rights reserved.


NSM5635I Querying NetFinity SPM Manager service
NSM5635I Querying NetFinity Alert Manager service

Invoking:  SPMLOGF.EXE
Arg string: C:\NETFIN\SPM.LOG PF2ILOG 30.0 1 30.0 3 4 5 10 15  LOCAL

SPMLOGF:  FileName............. C:\NETFIN\SPM.LOG
SPMLOGF:  Comment.............. PF2ILOG
SPMLOGF:  SampleInterval....... 30
SPMLOGF:  SamplesPerRecord..... 1
SPMLOGF:  WorkingSetInterval... 30
SPMLOGF:  Selected GroupNumber. 3
SPMLOGF:  Selected GroupNumber. 4
SPMLOGF:  Selected GroupNumber. 5
SPMLOGF:  Selected GroupNumber. 10
SPMLOGF:  Selected GroupNumber. 15
SPMLOGF:  Selected NodeName.... LOCAL
11:19:10 Call SPMAPIInit
11:19:11 Call SPMAPIQuery
11:19:16 Call SPMAPIOpen
11:19:17 00001 GetData in progress
```

*Figure 192. NFSPMMON Startup*

The help file for nfspmmon.exe is called NFSPM.INF and it shows the format of
the control file.  In our control file, we set up some monitoring characteristics so
that we would get alerts for CPU utilization, memory usage, working set size,
disk utilization and used memory.  The monitoring that we used is only a subset
of what can be set up.  As with all NetFinity 5.0 alerts, you can then use the Alert
Editor to take some specific action as a result of the alert flowing to NetFinity 5.0.
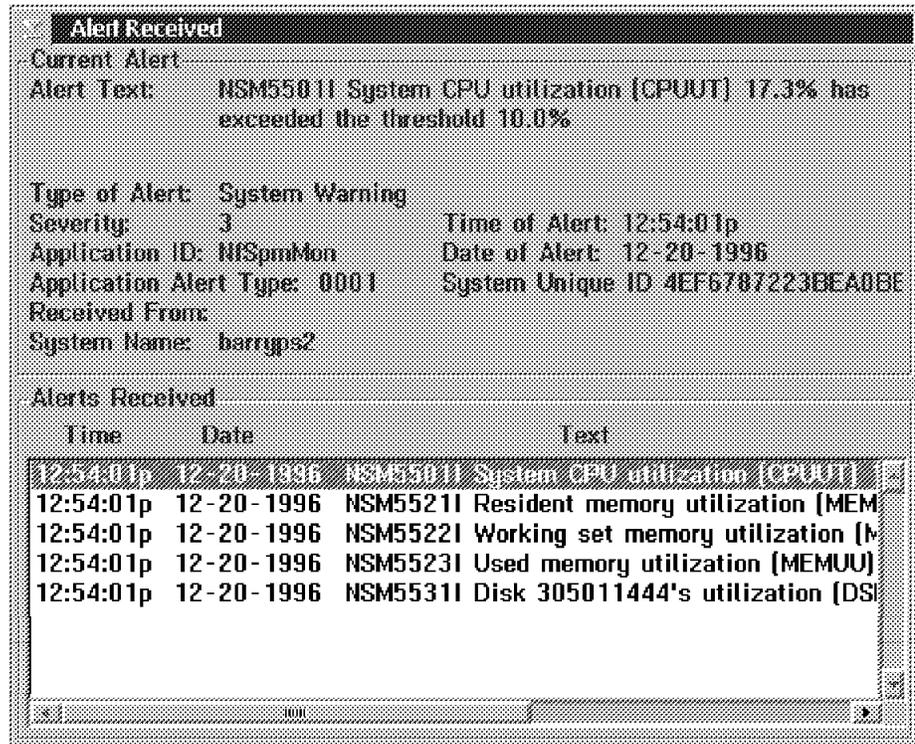
*Figure 193. SPM/2-Related Alerts*

### 5.4.3 SPM/2 and Database Queries

You can also specify that SPM/2-related information get stored in your DB2 database. We used DB2 Server V4.0. One of the functions of DB2 is the capability to view charts and tables using a GUI. The application that we used was called Visualizer. The first thing that we did was to drag a copy of Visualizer from the DB2 folder onto a separate place on the desktop and give it a new name. We called it LMU Flight. We then pointed the Visualizer to our SPM/2 database, which was called NFSPM, as is shown in the following figure:
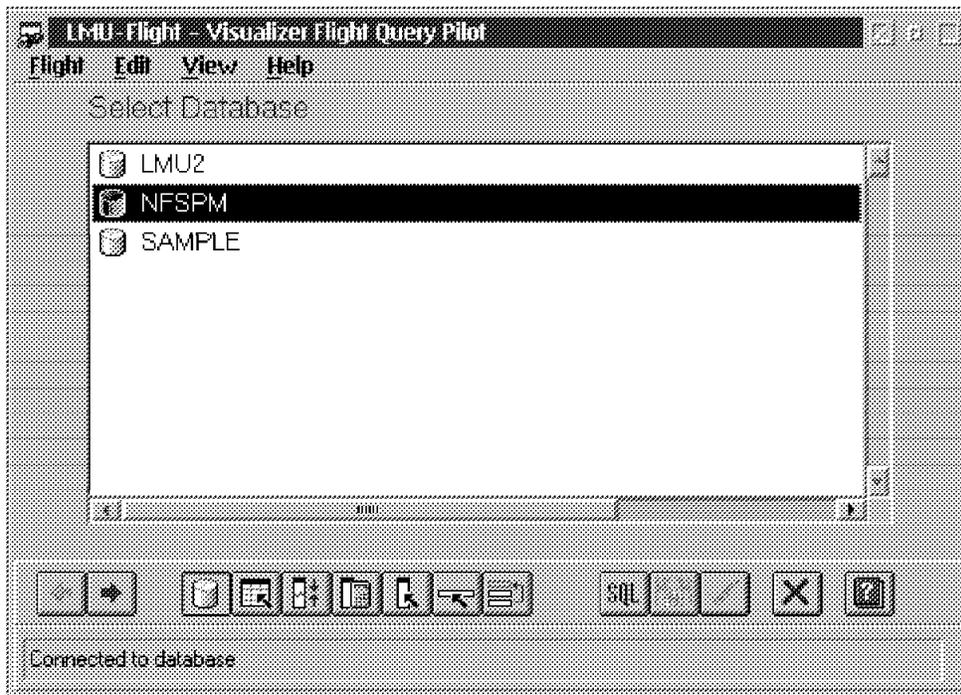
*Figure 194. Flight Visualizer*

You can specify the data collection points in the database you wish to represent in a window. The definition of all the fields is done for you by NetFinity 5.0. You will only need to specify which fields you want to see in the report.
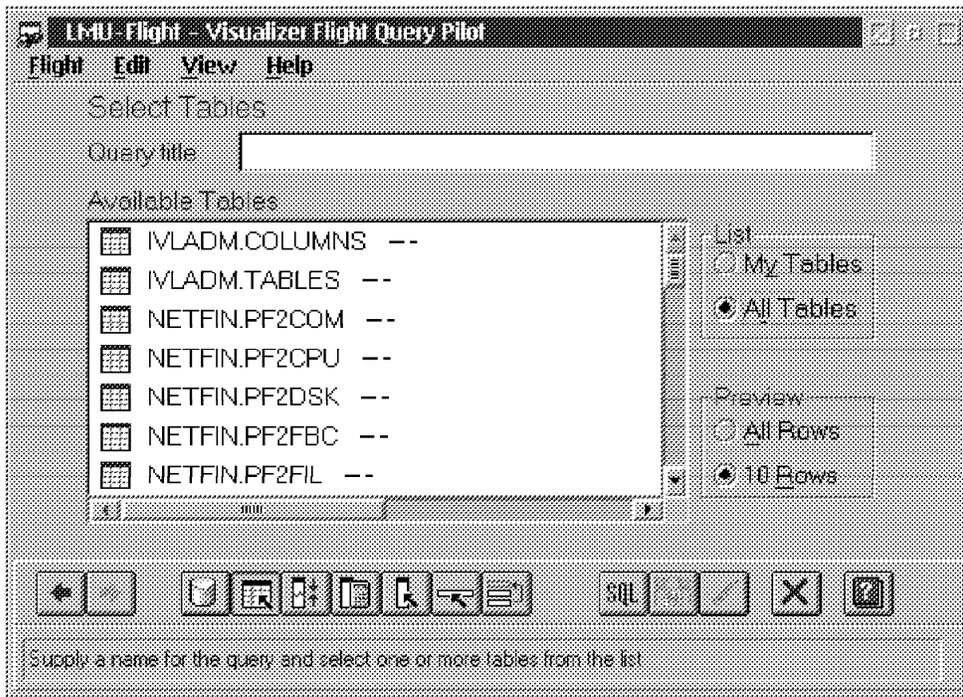
Figure 195. Available Tables

You need to specify the columns.
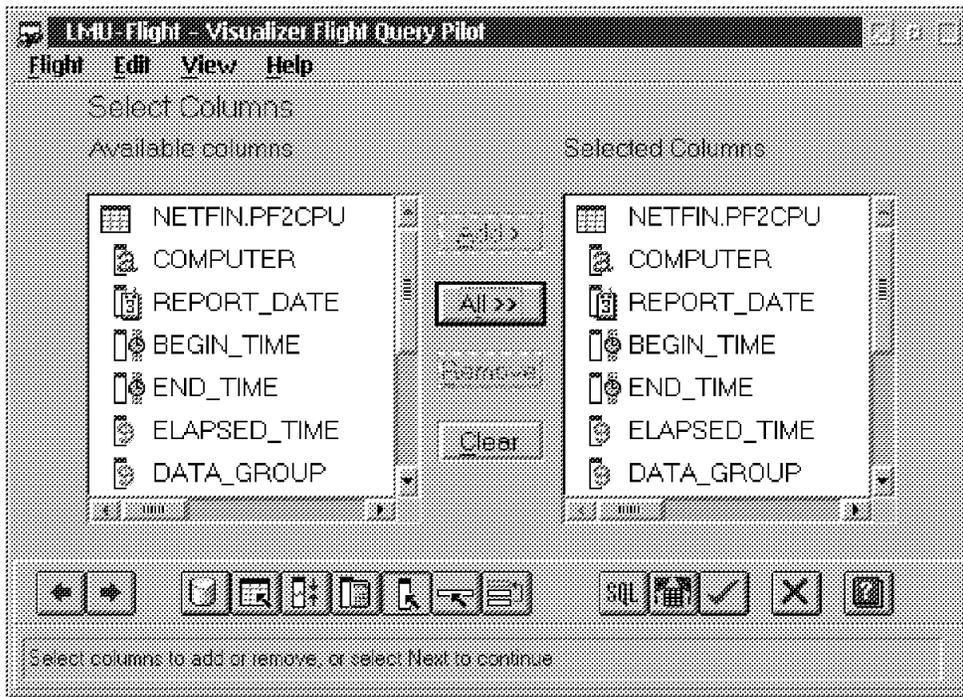


Figure 196. SPM/2 Database Columns

We wanted to graph CPU utilization by time, so we need to specify an X and a Y axis for the graph.
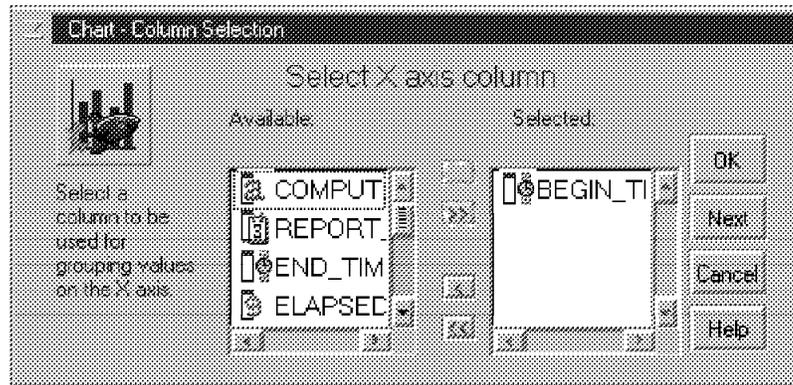


Figure 197. X Axis
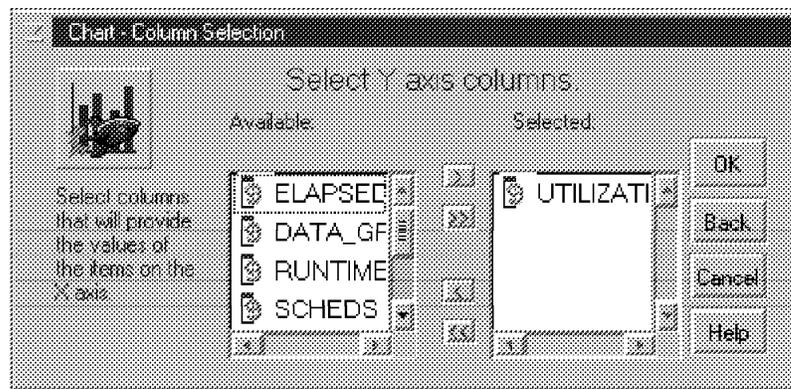


Figure 198. Y Axis

After specifying the begin time and the utilization fields we clicked on **OK** and the following chart appeared:
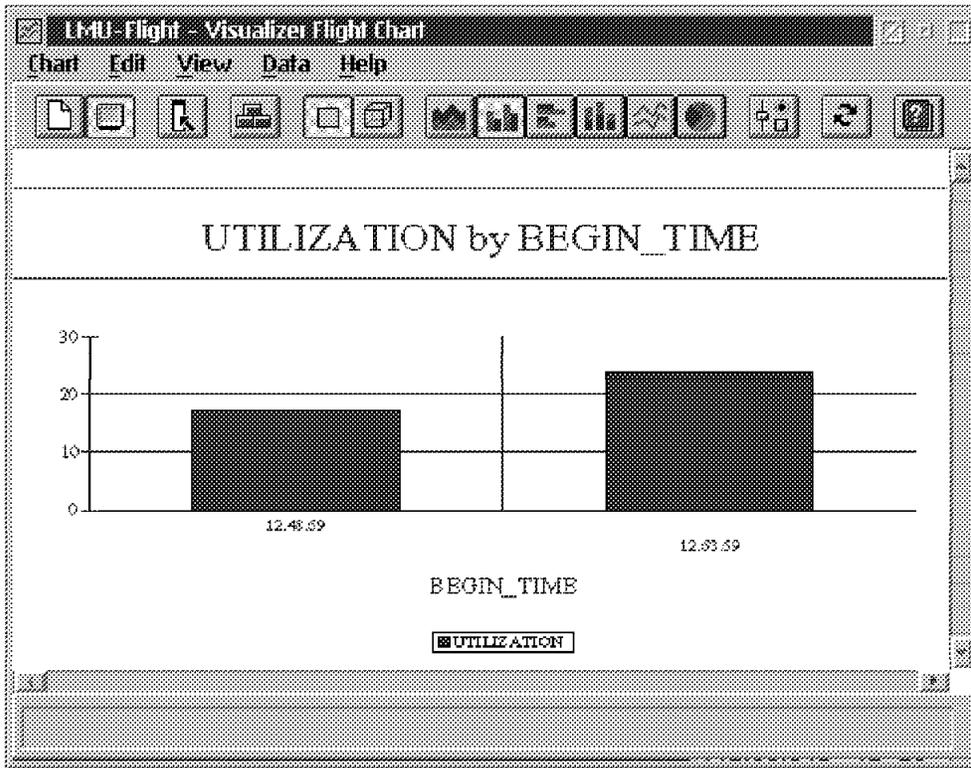
*Figure 199. CPU Utilization*

In addition to using charts with Visualizer, you can also present a table view of your data. From the mail Visualizer Pilot window you should click on **Table View**.
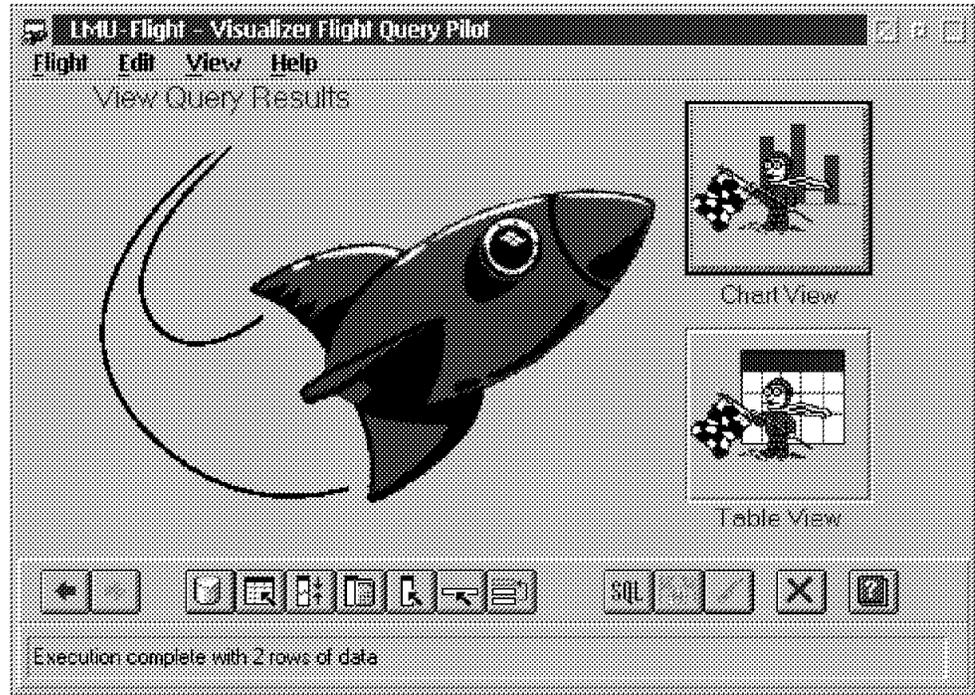
*Figure 200. Visualizer Flight Query Pilot*

We selected a report for all the time that we were gathering data and for a few of the fields. The following two figures represent the data that was collected. Since the table is very wide, we captured it in two segments.

*Figure 201. Left Side of Flight Table View*

*Figure 202. Right Side of Flight Table View*

The following two screens are from the SPM.SLG log file that was previously specified.

```
****** NFSPMMON alert check      for barryps2 at 1996-12-20 12.53.59
Alert generated (period 12.48.59 to 12.53.59), NSM5501I System CPU utilization (CPUUT) 17.3% has exceeded the threshold 10.0%
Alert generated (period 12.52.59 to 12.53.59), NSM5521I Resident memory utilization (MEMRU) 32.6% has exceeded the threshold 5.0%
Alert generated (period 12.52.59 to 12.53.59), NSM5522I Working set memory utilization (MEMWU) 48.6% has exceeded the threshold 5.0%
Alert generated (period 12.52.59 to 12.53.59), NSM5523I Used memory utilization (MEMUU) 98.8% has exceeded the threshold 5.0%
Alert generated (period 12.48.59 to 12.53.59), NSM5531I Disk 305011444's utilization (DSKUT) 14.8% has exceeded the threshold 5.0%
****** NFSPMMON database records for barryps2 at 1996-12-20 12.53.59, block size 5810 sent to SPM Manager
****** NFSPMMON summary analysis for barryps2 from 1996-12-20 12.48.59 to 12.53.59 ********
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
CPU analysis on 1996-12-20 12.48.59
   Units ->            sec            sec             pct      pct       pct      msec
   Start    Stop    Elapsed  #Samples Runtime  Scheds  Util     Exe Util Int Util Timeslic
   _____  _____  _____  _____  _____  _____  _____    _____ _____ _____
   12.48.59 12.53.59 0299.651      10  0051.874 00024281 0017.312 0013.891 0003.421 0002.036
                                                         0007.402 0006.737 0000.666 0000.325
```

```
Process analysis on 1996-12-20 12.48.59 (process utilization >= 0.00 %)
Units ->            sec              sec            pct     msec   flt/sec msec/flt
Start    Stop     Elapsed  Samp/PID Runtime  Scheds     Util    Timeslic Flt Rate Flt Serv Name
─────    ────     ───────  ──────── ───────  ──────     ────    ──────── ──────── ──────── ────
12.48.59 12.53.59 0299.651   11/00B1 0000.221 00000060 0000.074 0003.965 0000.042 0000.120 C:\SPM2V2\SPMDCF.EXE
0000.014 0001.103 0000.013 0000.001
12.48.59 12.53.59 0299.651   40/00B0 0000.191 00000157 0000.064 0001.101 0000.033 0000.120 C:\SPM2V2\SPMILOG.EXE
0000.011 0000.630 0000.000 0000.000
12.48.59 12.53.59 0299.651   10/0034 0000.003 00000011 0000.001 0000.291 0000.000 0000.000 C:\IBMLAN\SERVICES\PEER.EXE
0000.001 0000.115 0000.000 0000.000
12.48.59 12.53.59 0299.651   20/0033 0000.032 00000043 0000.011 0000.559 0000.000 0000.000 C:\IBMLAN\SERVICES\PEER.EXE
0000.004 0000.329 0000.000 0000.000
12.48.59 12.53.59 0299.651   16/002E 0000.097 00000131 0000.032 0000.561 0000.000 0000.000 C:\IBMLAN\SERVICES\WKSTAHLP.EXE
0000.015 0000.248 0000.000 0000.000
12.48.59 12.53.59 0299.651   22/002D 0000.457 00000263 0000.152 0002.037 0000.033 0000.118 C:\IBMLAN\SERVICES\WKSTA.EXE
0000.028 0000.903 0000.000 0000.000
12.48.59 12.53.59 0299.651   10/0025 0000.055 00000297 0000.018 0000.187 0000.000 0000.000 C:\OS2\EPWMP.EXE
0000.001 0000.012 0000.000 0000.000
12.48.59 12.53.29 0269.684    5/001E 0000.002 00000006 0000.001 0000.300 0000.000 0000.000 C:\SPM2V2\SPMSNAPL.EXE
0000.001 0000.130 0000.000 0000.000
12.52.59 12.53.29 0029.964    1/001C 0000.000 00000001 0000.001 0000.349 0000.000 0000.000 C:\SPM2V2\SPMNBL.EXE
0000.000 0000.000 0000.000 0000.000
12.48.59 12.53.59 0299.651   10/0018 0000.003 00000018 0000.001 0000.169 0000.000 0000.000 C:\IBMCOM\PROTOCOL\LANDLL.EXE
0000.000 0000.017 0000.000 0000.000
12.48.59 12.53.59 0299.651   55/000F 0001.998 00008788 0000.667 0000.295 0000.000 0000.000 C:\MPTN\BIN\CNTRL.EXE
0000.066 0000.181 0000.000 0000.000
12.48.59 12.53.59 0299.651   12/0001 0000.393 00000074 0000.131 0000.839 0000.000 0000.000
0000.197 0001.080 0000.000 0000.000
12.48.59 12.53.59 0299.651   74/001D 0009.073 00004200 0003.028 0001.108 0000.200 0129.607 C:\OS2\PMSHELL.EXE
0000.660 0001.056 0000.156 0200.298
12.49.29 12.53.59 0269.689    9/00AE 0000.190 00000019 0000.070 0010.042 0000.095 0000.293 C:\NETFIN\NFSPMMON.EXE
0000.007 0000.702 0000.061 0000.090
12.48.59 12.52.59 0239.719    6/0057 0000.532 00000137 0000.222 0004.265 0000.254 0000.072 C:\NETFIN\NETFIN.EXE
0000.285 0002.223 0000.163 0000.039
12.48.59 12.53.59 0299.651   12/0055 0000.008 00000033 0000.003 0000.270 0000.000 0000.000 C:\SQLLIB\BIN\DB2SYSC.EXE
0000.000 0000.064 0000.000 0000.000
12.49.29 12.52.29 0179.791    3/004D 0000.133 00000067 0000.074 0002.061 0000.389 0012.519 C:\NETFIN\ALERTMGR.EXE
0000.142 0000.305 0000.475 0016.340
12.49.29 12.49.59 0029.965    1/004C 0000.059 00000004 0000.198 0014.846 0000.167 0000.260 C:\TCPIP\BIN\SENDMAIL.EXE
0000.000 0000.000 0000.000 0000.000
12.48.59 12.49.29 0029.959    1/0047 0000.000 00000001 0000.001 0000.154 0000.000 0000.000 C:\NETFIN\SCHBASE.EXE
0000.000 0000.000 0000.000 0000.000
12.48.59 12.53.59 0299.651   44/0043 0000.477 00000518 0000.159 0000.965 0000.050 0000.259 C:\NETFIN\NETFBASE.EXE
0000.031 0000.568 0000.017 0000.003
12.48.59 12.53.59 0299.651   10/0052 0000.094 00000150 0000.031 0000.628 0000.000 0000.000 C:\NETFIN\WEBFIN.EXE
0000.002 0000.024 0000.000 0000.000
12.48.59 12.53.59 0299.651   10/004E 0001.727 00000198 0000.576 0008.780 0000.617 0000.260 C:\NETFIN\MONBASE.EXE
0000.016 0000.579 0000.017 0000.005
12.48.59 12.53.59 0299.651   20/0042 0000.114 00000079 0000.038 0000.988 0000.000 0000.000 C:\IBMLAN\NETPROG\NETMSG.EXE
0000.019 0000.861 0000.000 0000.000
12.48.59 12.53.59 0299.651   47/0029 0007.658 00002206 0002.556 0005.049 0000.353 0075.854 C:\OS2\PMSHELL.EXE
0000.755 0005.618 0000.285 0097.061
12.49.29 12.50.59 0089.897    2/0021 0000.002 00000003 0000.002 0000.631 0000.000 0000.000 C:\OS2\SYSTEM\HARDERR.EXE
0000.001 0000.001 0000.000 0000.000
12.48.59 12.53.29 0269.684    5/0006 0000.001 00000005 0000.000 0000.159 0000.000 0000.000 C:\NETFIN\EQNKCTKB.EXE
0000.000 0000.012 0000.000 0000.000
Process summary (pct activity brackets):     0.0    0 - 1%   1 - 5%     > 5%    Total
39       26        0        0       65
```

```
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Disk no. 1 analysis on 1996-12-20 12.48.59
Units ->             sec            sec          pct        KB   KB/sec   KB/sec   msec/KB
Start    Stop     Elapsed  #Samples Disk Tm RdWr Cnt Disk Ut RdWr Byt Rd ThruP Wr ThruP Cb Index
_____  _____  _____  _____  _____
12.48.59 12.53.59 0299.651       10 0044.349 00000925 0014.800 003191.5 0005.199 0005.452 0034.270
0007.108          0006.290 0006.037 0034.489
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Working set analysis on 1996-12-20 12.48.59
Units ->             sec          pages      MB       MB      pct       MB      pct       MB      pct
Start    Stop     Elapsed  #Samples Tot Mem Tot  Mem Resident Resident Work Set Work Set Used Mem Used Mem
_____  _____  _____  _____  _____
12.48.59 12.53.59 0299.651       10 00009120 0035.625 0011.529 0032.362 0017.985 0050.485 0034.364 0096.459
0000.101 0000.284 0000.863 0002.421 0000.979 0002.749
Memory analysis on 1996-12-20 12.48.59
Units ->             sec                         pg/sec            flt/sec
Start    Stop     Elapsed  #Samples Page In  Page Out Pg Rate  Faults   Flt Rate
_____  _____  _____  _____  _____
12.48.59 12.53.59 0299.651       10 00000470 00000000 0001.568 00001281 0004.275
0002.189          0005.004
```

## 5.4.4  Watching Processes

LMU comes with the function APPWATCH.EXE.  In order to control an application you have to edit the APPWATCH.TAB control file.

```
# Sample application table
#
#  Appl Name      Minimum     Maximum     Threshold    Reset/Accumulate
# -----------     -------     -------     ---------    ----------------
  APPWATCH.EXE
  LMUSRV.EXE
  LMUCLI.EXE         *           *            1        A
  LMUIPL.SYS         *           1            *        *
  AUEPITCH.EXE       1           2            *        R
  KLONDIKE.EXE       0           0            2        A
#
  ACSNETB.DLL
  NWCALLS.DLL
  OS2DASD.DMD
  OS2SCSI.DMD
  HELV.FON
  HPFS.IFS
  NWIFS.IFS
# Where:
#  Appl Name            Application name:
#                          - For OS/2 1.x this is the internal process name
#                            of the application (1 to 8 characters);
#                          - For OS/2 2.x this is the file name (including
#                            extension) of the application (1 to 12 characters)
#  Minimum              Minimum instances of application; range is 0 to 256;
#                       default is 1
#  Maximum              Maximum instances of application; range is 0 to 256;
#                       default is 1
#  Threshold            Number of violations (process instances not between
#                       minimum and maximum) before an alert is sent;
#                       range is 1 to 128; default is 1
#  Reset/Accumulate     R·ESET" indicates that the violation count should
#                       be reset to zero after an alert is sent;
#                       A·CCUMULATE" indicates that the violation count
#                       should not be reset after an alert is sent;
#                       default is RESET
# Notes:
#   - Only application name is required
#   - Use an asterisk (*) as a place holder (default value of entry is used)
#   - Table entries are order dependent but not column dependent
#   - Separate table entries with one or more blanks
#   - A non-EXE (device driver, dynamic link library, font, etc.) that
#     has been loaded constitutes one instance; otherwise the number of
#     instances is 0
```

*Figure 203. APPWATCH.SMP*

You can watch if a process is running. According to definitions an alert will be created. As an example we started APPWATCH with the above table while KLONDIKE.EXE was up. Here is what you get:

```
[C:\lmu2]appwatch /I2 /R /S1 /Tappwatch.tab
IBM LAN NetView Management Utilities
5622-153 (C) Copyright IBM Corp. 1991, 1996. All rights reserved.

Iteration 1 of 2 at 1996-10-23 10.14.00

    Name       Instances  Minimum  Maximum  Threshold  Reset  Violations  Alert
------------  ---------  -------  -------  ---------  -----  ----------  -----
APPWATCH.EXE          1        1        1          1  YES             0  NO
LMUSRV.EXE            1        1        1          1  YES             0  NO
LMUCLI.EXE            1        1        1          1  NO              0  NO
LMUIPL.SYS            1        1        1          1  YES             0  NO
AUEPITCH.EXE          1        1        2          1  YES             0  NO
KLONDIKE.EXE          1        0        0          2  NO              1  NO*
ACSNETB.DLL           1        1        1          1  YES             0  NO
NWCALLS.DLL           1        1        1          1  YES             0  NO
OS2DASD.DMD           1        1        1          1  YES             0  NO
OS2SCSI.DMD           1        1        1          1  YES             0  NO
HELV.FON              1        1        1          1  YES             0  NO
HPFS.IFS              1        1        1          1  YES             0  NO
NWIFS.IFS             1        1        1          1  YES             0  NO

Sleeping for 1 minute(s)

Iteration 2 of 2 at 1996-10-23 10.15.01

    Name       Instances  Minimum  Maximum  Threshold  Reset  Violations  Alert
------------  ---------  -------  -------  ---------  -----  ----------  -----
APPWATCH.EXE          1        1        1          1  YES             0  NO
LMUSRV.EXE            1        1        1          1  YES             0  NO
LMUCLI.EXE            1        1        1          1  NO              0  NO
LMUIPL.SYS            1        1        1          1  YES             0  NO
AUEPITCH.EXE          1        1        2          1  YES             0  NO
KLONDIKE.EXE          1        0        0          2  NO              2  YES
ACSNETB.DLL           1        1        1          1  YES             0  NO
NWCALLS.DLL           1        1        1          1  YES             0  NO
OS2DASD.DMD           1        1        1          1  YES             0  NO
OS2SCSI.DMD           1        1        1          1  YES             0  NO
HELV.FON              1        1        1          1  YES             0  NO
HPFS.IFS              1        1        1          1  YES             0  NO
NWIFS.IFS             1        1        1          1  YES             0  NO

Watch complete
```

What you can see is that, as defined in APPWATCH.TAB, after APPWATCH found
KLONDIKE.EXE running for the second time, an alert is generated. If you have
the GUI up and running, the machine's icon(s) starts blinking to indicate that an
event was added to the log. You can also find this alert in the LMU.LOG:

```
1996-10-23 10.15.03  AUE0150I Sending user alert; code=21, application =
APPWATCH, message = "LMU4790W    1 instance(s) of KLONDIKE.EXE", name = ""
```

### 5.4.4.1  NetFinity 5.0

Watching for processes in NetFinity 5.0 Process Manager is similar. The only
difference is that you cannot define an alert that is generated when a process is
found running for the second time. (That is, you cannot accumulate violations of
rules you set up for the process.) You can define an alert to be generated when
a certain process runs or when it does not run or even when it stops executing.
If you want to generate an alert if FREECELL.EXE runs on a Windows NT
machine, you can set this up with the following command:

```
[C:\netfin]nfproccl /n:NETBIOS::WTR05185 /addmon:"c:
\winnt35\system32\freecell.exe" /sev:0 /onstart:enabled /addnotify:local
NetFinity Process Manager Command-Line Tool
Copyright (C) IBM Corp. 1996.
TAG=0x0A1A8AA8
```

If someone starts Freecell on the system WTR05185, you get an alert action according to what is defined.

```
┌─────────────────────────── Alert Received ───────────────────────────┐
│ ┌─Selected Alert──────────────────────────────────────────────────┐  │
│ │ Alert Text :      Process 'C:\WINNT35\SYSTEM32\FREECELL.EXE' has │  │
│ │                   been started.                                 │  │
│ │                                                                 │  │
│ │ Type of Alert :       Application Information                   │  │
│ │ Severity :      0                    Time of Alert :   01:11:09p│  │
│ │ Application ID :    ProcMgr          Date of Alert :  10/23/1996│  │
│ │ Application Alert Type :        0900                            │  │
│ │ Received From :                                                │  │
│ │ System Name :     TME05185                                     │  │
│ └────────────────────────────────────────────────────────────────┘  │
│ ┌─Alerts In Log─────────────────────────────────────────────────┐   │
│ │  Time       Date              Text                           │   │
│ │ 01:11:00p 10/23/1996 User ID 'MAKLEIN' from Address 'WTR05144'│   │
│ │ 01:11:09p 10/23/1996 Process 'C:\WINNT35\SYSTEM32\FREECELL.E  │   │
│ └────────────────────────────────────────────────────────────────┘  │
└─────────────────────────────────────────────────────────────────────┘
```

*Figure 204. Alert Pop-Up*

┌─ **Remark** ──────────────────────────────────────────────────────┐
│                                                                   │
│ For a complete set of functions and syntax of NFPROCCL.EXE see 2.1.7, │
│ "Process Manager - NFPROCCL.EXE" on page 38.                      │
│                                                                   │
└───────────────────────────────────────────────────────────────────┘

## 5.4.5 System Performance Data

The LAN Network Management Utilities for OS/2 have some command line functions to watch the system performance. These functions are:

1. DRWATCH.EXE: Watch Diskspace

2. IPXWATCH.EXE: Watch IPX Traffic

3. NBWATCH.EXE: Watch NetBIOS

4. OSRWATCH.EXE: Monitor Adapter

5. RUWATCH.EXE: Measure Ring Utilization

6. SRVWATCH.EXE: Monitor a LAN Server

7. PERFCAP2.EXE: Get data from System Performance Monitor for OS/2

Most of these functions can be replaced by the new NetFinity 5.0 command line interfaces. We have provided some examples of how to perform the functions using NetFinity 5.0.

If you run DRWATCH without any parameters, it checks all drives from C to Z and creates an event for every drive whose percentage full is greater than 75%. But you can also run it with some parameters specified. For example, provide a new threshold for the event generation or specify a list of drives to be checked.

```
IBM LAN NetView Management Utilities
5622-153 (C) Copyright IBM Corp. 1991, 1996. All rights reserved.

LMU7756I Iteration 1 of 1 at 1996-10-23 15.29.29
LMU7766I Drive C is 66% full
LMU7767W Drive D not attached
LMU7767W Drive E not attached
LMU7767W Drive F not attached
LMU7767W Drive G not attached
LMU7767W Drive H not attached
LMU7767W Drive I not attached
LMU7767W Drive J not attached
LMU7767W Drive K not attached
LMU7770W Drive L ignored; not local
LMU7767W Drive M not attached
LMU7767W Drive N not attached
LMU7767W Drive O not attached
LMU7767W Drive P not attached
LMU7767W Drive Q not attached
LMU7767W Drive R not attached
LMU7767W Drive S not attached
LMU7767W Drive T not attached
LMU7767W Drive U not attached
LMU7770W Drive V ignored; not local
LMU7770W Drive W ignored; not local
LMU7767W Drive X not attached
LMU7767W Drive Y not attached
LMU7770W Drive Z ignored; not local
```

*Figure 205. Output for a DRWATCH /L*

In NetFinity 5.0 you can use the command line support for System Monitor to perform the same functions. You can also use it in REXX execs so that you can make your actions more dynamic, based upon operational criteria. You can get an overview on all local drives. In addition, if you have a MIB browser or only SNMP get support, you can extract all of the current monitor values into a file.

If you don't have access to a MIB browser, but you do have SNMP support, you can still use the command line to get the same information. We wrote a small AIX shell to get the same values:

```
snmpget -c public barryps2 .1.3.6.1.4.1.2.6.71.200.2.1.3.1.1.1.1663545570
snmpget -c public barryps2 .1.3.6.1.4.1.2.6.71.200.2.1.3.1.2.1.1663545570
snmpget -c public barryps2 .1.3.6.1.4.1.2.6.71.200.2.1.3.1.3.1.1663545570
snmpget -c public barryps2 .1.3.6.1.4.1.2.6.71.200.2.1.3.1.4.1.1663545570
snmpget -c public barryps2 .1.3.6.1.4.1.2.6.71.200.2.1.3.1.5.1.1663545570
snmpget -c public barryps2 .1.3.6.1.4.1.2.6.71.200.2.1.3.1.6.1.1663545570
```

*Figure 206. Get SNMP Values*

The output of the above commands follows. Note, that you can also write a small routine to parse the output and only get the information that you need. The snmpget returns the variable as well as the result at the end of each line.

```
ibm.ibmProd.netFinity.DmiMibs.NetFinityServicesMIB.dmtfGroups.tTme10NetfinityMon
ibm.ibmProd.netFinity.DmiMibs.NetFinityServicesMIB.dmtfGroups.tTme10NetfinityMon
ibm.ibmProd.netFinity.DmiMibs.NetFinityServicesMIB.dmtfGroups.tTme10NetfinityMon
ibm.ibmProd.netFinity.DmiMibs.NetFinityServicesMIB.dmtfGroups.tTme10NetfinityMon
ibm.ibmProd.netFinity.DmiMibs.NetFinityServicesMIB.dmtfGroups.tTme10NetfinityMon
ibm.ibmProd.netFinity.DmiMibs.NetFinityServicesMIB.dmtfGroups.tTme10NetfinityMon
```

*Figure 207. Output of snmpget*

You can also turn monitoring on for the disk drive in NetFinity 5.0 using the command line:

```
[C:\netfin]nfsmoncl /getmon /monname:"Drive C: % Space Used"
NetFinity System Monitor Command-Line Tool
Copyright (C) IBM Corp. 1996.
{ MONITOR_ID=0x6327B1E2, NAME="Drive C: % Space Used",
  SAMPLE=300000, UNITS_LBL="Percent Used", RECORDING=DISABLED,
  MINVAL=0.000000, MAXVAL=100.000000, VALUE=66.669783
}
```

In addition, you can get a lot of information about all of the hard drives by looking at the output of the NetFinity 5.0 System Information tool. In the Disk Information section you get a list of all logical drives, including all attached remote drives.

```
*********** Logical Drive Information ***********
C: Fixed Disk Drive
Volume            OS2
File system name  HPFS
Drive type        Local drive
Sectors per cluster 1
Disk size         529,392 Kilobytes
Available space   171,323 Kilobytes
Bytes per sector  512
*********** Logical Drive Information ***********
L: Remote Disk Attached by LAN
Volume               SYS
File system name     NETWARE
File system attach name
Drive type           Remote drive attached to FSD
Sectors per cluster  64
Disk size            204,800 Kilobytes
Available space      0 Kilobytes
Bytes per sector     512
*********** Logical Drive Information ***********
V: Remote Disk Attached by LAN
Volume               IBM23MN465
File system name     LAN
File system attach name \\WTRDC\LANCMDS
Drive type           Remote drive attached to FSD
Sectors per cluster  1
Disk size            162,800 Kilobytes
Available space      143,443 Kilobytes
Bytes per sector     512
*********** Logical Drive Information ***********
W: Remote Disk Attached by LAN
Volume               IBM23MN465
File system name     LAN
File system attach name \\WTRDC\LANDLLS
Drive type           Remote drive attached to FSD
Sectors per cluster  1
Disk size            162,800 Kilobytes
Available space      143,443 Kilobytes
Bytes per sector     512
*********** Logical Drive Information ***********
Z: Remote Disk Attached by LAN
Volume               F_DRIVE
File system name     LAN
File system attach name \\WTRAS1X\NT_CODE
Drive type           Remote drive attached to FSD
Sectors per cluster  1
Disk size            2,252,784 Kilobytes
Available space      443,821 Kilobytes
Bytes per sector     512
```

*Figure 208. Disk Drive Information in NetFinity 5.0 System Information*

The only watch function that comes with LMU that has some parallels in NetFinity 5.0 is the SRVWATCH. This function watches a LAN Server and sends an alert for the following conditions:

**SAVRms**[**,cnt**] Server Average Response Time Exceeds ms Milliseconds After cnt Exceptions.

**SRBFn** Regular Buffer Allocation Failures >= n.

**SBBFn**    Big Buffer Allocation Failures >= n.

**TSESn**    Number of Sessions at this Server >= n.

**TCONn**    Number of Connections at this Server >= n.

**TOPNn**    Number of File Opens at this Server >= n.

**SSESn**    Number of Sessions for a Single User >= n.

**SCONn**    Number of Connections for a Single User >= n.

**SOPNn**    Number of File Opens for a Single User >= n.

**SIDLn**    Session Idle Time for a Single User >= n.

The corresponding NetFinity 5.0 System Monitors would be the OS/2 LAN Server
Monitors:

- Sessions $\rightarrow$ TSESn
- Connections $\rightarrow$ TCONn
- Opens $\rightarrow$ TOPNn
- Shares
- Bytes Sent
- Bytes Received
- Response Time $\rightarrow$ SAVRms[,cnt]
- Request Buf Shortage $\rightarrow$ SRBFn
- Big Buf Shortage $\rightarrow$ SBBFn

## 5.5 Fault Management

We next try to compare the fault management functions of NetFinity 5.0 with
LMU. The following table shows the functions implemented in the different
programs. You see some LMU functions are not supported or not yet available
in NetFinity 5.0.

```
| LMU                       | Netfin 4.1
|---------------------------+-------------------------
| AUEUSRGA - Generate Alert | GENALERT
| AUEREXGA                  |
| LMUEVENT                  |
| AUEVERTB - Verify Al.Tbl. |
| DOSVIRGA - Gen. VirAlert  | GENALERT
| VIRALERT                  | GENALERT
| EPWRT003 - FFST Fehler    |
| LMUPOPUP - Popup Msg.     | Alert Action
| LMUPOPW                   |
| LMUPAGER / LMUPAGE        | Alert Action
| Forwarding of Alerts      | Alert Action
| Alert automation          | Alert action
| NMVT creation             | << New alert action
| MSM support               | << 3/97
```

We have a closer look at some of the functions now.

## 5.5.1 Generate Alerts

Let's first look at the alert generation function(s). In LMU you have two DLLs that provide you with the necessary routines to write a C or REXX program to generate the alert. The AUEUSRGA.DLL is for C programs while the AUEREXGA is for REXX programs. Both DLLs are for OS/2 machines only. The GENALERT function is available on all platforms for NetFinity 5.0. Here is the syntax for the genalert command:

```
GENALERT /T:"text" < /APP:id_name > < /PRI:<0..7> > </TYPE:sssttt >
   < /ATYPE:hexnum >, where:
  /T:"text"      - Defines the text message describing the alert
  /APP:id_name  - Defines the application ID for the alert (1-8 characters)
  /SEV:<0..7>    - Defines the severity of the alert (0 = max, 7=min)
  /TYPE:sssttt  - Defines the standard type of alert.  The 'sss'
                    field describes the ID of the alert:
                        UNK  - Unknown
                        SYS  - System
                        DSK  - Disk or DASD
                        NET  - Network
                        OS_  - Operating System
                        APP  - Application
                        DEV  - Device
                        SEC  - Security
                    The 'ttt' field describes the class of the alert:
                        UNK  - Unknown
                        FLT  - Fault or Failure
                        ERR  - Error
                        WRN  - Warning
                        INF  - Information
  /ATYPE:hexnum - Defines the application-specific alert type as a
                    hexadecimal value between 0000 and FFFF.
```

The command genalert /T:"test alert" /APP:"myapp" /SEV:3 /type:sysinf /atype:ffff results in the following pop-up if the pop-up is specified as an appropriate action for that alert (by default, it is).
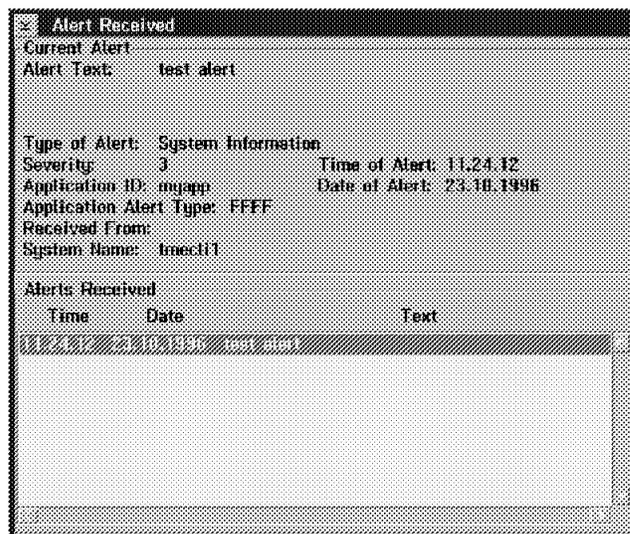


Figure 209. Genalert Sample Output

The LMUEVENT command is another way to send alerts to either the GUI or to the SNMP Proxy Agent. This command runs on OS/2 machines only. You can specify a target system to receive the event. If you don't provide any information about that, the address is taken from the lmu.ini file (keys GRAPHICAL_USER_INTERFACE and SNMP_PROXY_AGENT).

```
   APP(LMU_UTILITY),
      KEY(GRAPHICAL_USER_INTERFACE),
      ASCIIZ(WTR05144);

   APP(LMU_UTILITY),
      KEY(SNMP_PROXY_AGENT),
      ASCIIZ(WTR05144);
```

Figure 210. LMU.CTL Values

Sending the command lmuevent /G wtr05188 "LMUEVENT test alert" results in a flashing GUI and when you open the event list for the managed machine with the hashmarks you see the event text.
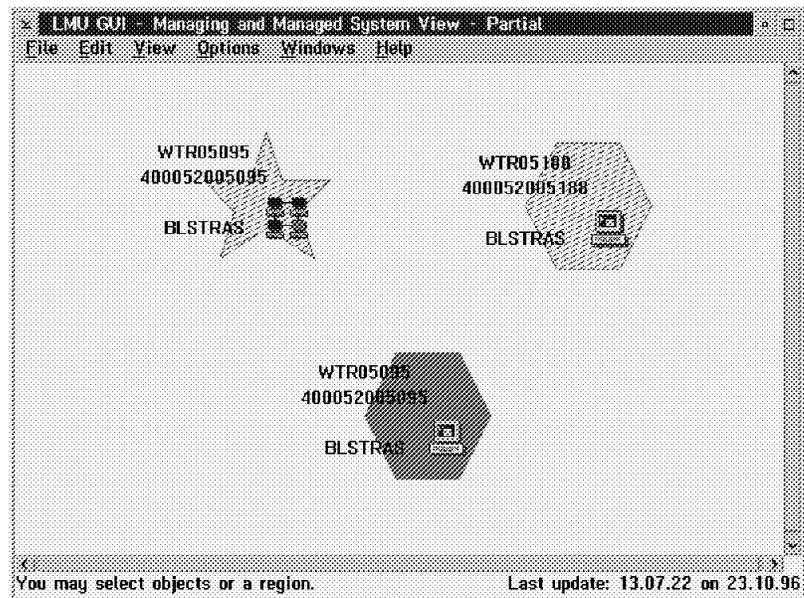


Figure 211. LMU GUI Alert Arrived

To see the event table, double-click with the right mouse button on the icon of the hashed managed machine.
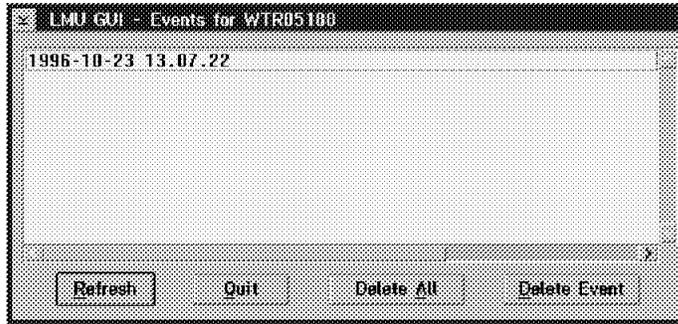
*Figure 212. LMU GUI Event Table - Left Side*

Remember to scroll to the right to see the text.
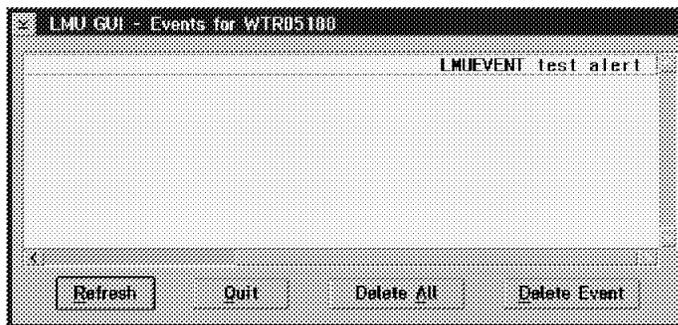


*Figure 213. LMU GUI Event Table - Right Side*

## 5.5.2 Virus Warnings

The antivirus alerts DOSVIRGA and VIRALERT are not implemented in NetFinity 5.0. These executables generate a virus alert. The LMU version of this alert only provides fault reporting. You could just as easily use the antivirus reporting mechanism to issue a genalert when a virus is detected. The virus detection has to be done by an antivirus program of your choice (for example, IBM Antivirus). These programs normally return a 0 when no virus is found and a non-zero return code if a virus is found. Start the antivirus software from a batch file and check for the return code after completion of the virus check. If the return code is not 0, start GENALERT for NetFinity 5.0 with the appropriate parameters (for example, genalert /T:″Possible Virus detected″ /APP:″ANTIVIR″ /SEV:0 /TYPE:secwrn). This command displays the following window:
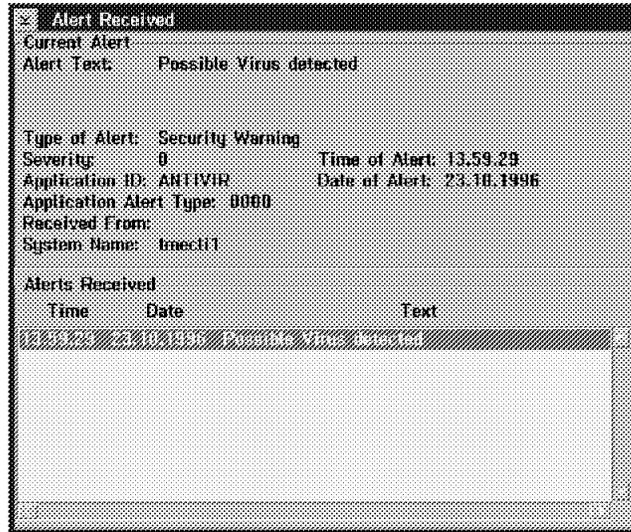
*Figure 214. NetFinity 5.0 Antivirus Sample*

With LMU use the appropriate command for your operating system such as VIRALERT for OS/2 or DOSVIRGA for DOS. The command VIRALERT results in a flashing GUI on the LMU manager and when you double-click with the right mouse button on the icon of the hashed managed machine you see the alert like in the following picture.

Viralert generates an alert that the fault manager can automatically respond to. Since it can respond to it by calling an executable or a REXX exec, you can customize its actions. You can also forward the alert to the GUI.
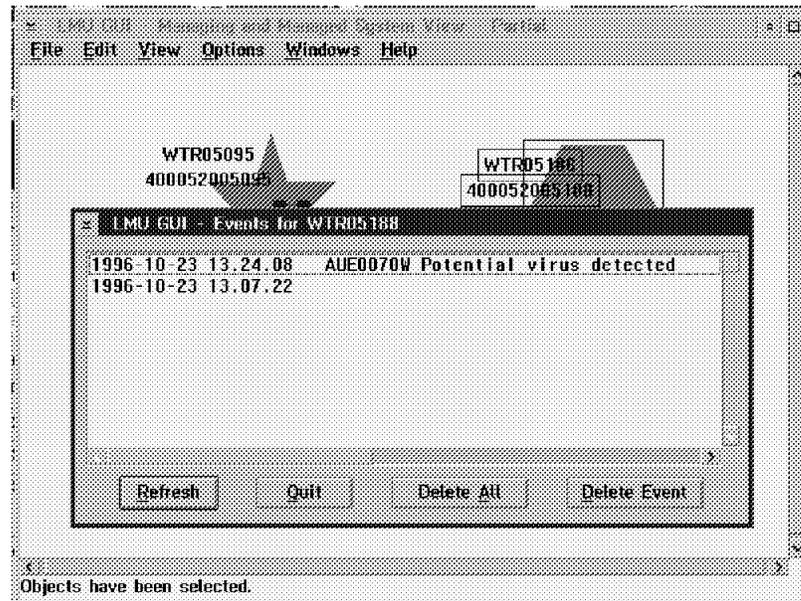


*Figure 215. LMU Antivirus Sample*

The following sample program shows how to automate the sending of a virus alert in a NetFinity 5.0 environment. Make sure the managed system has an alert action to forward the alert through the network to the managing system. See 4.3, "The Alert Manager" on page 137 for details on how to set this action for a complete group. The sample program has do be in the same directory as

the antivirus software is installed or the path statement has to contain the path to the software. In our sample we use IBM Antivirus software on an OS/2 system. The program is fairly simple. We start the IBMAVSP command, which is the stand-alone utility for virus checking under OS/2. There are a lot of parameters for that utility and we recommend that you look into the online documentation of the antivirus software. After the checking is done we get a return code (rc). We only check a few of the possible return codes here in our sample. If you want to check other rcs, see the online documentation for a description of the rcs.

Here is the listing:

```
/**/
'ibmavsp * -Programs -copenerr -nlmsg'
select
when rc=30 then
   'genalert /t:"The program was interrupted by the user.",
   /APP:"IBMAVSP" /SEV:0 /TYPE:SYSERR /ATYPE:' rc

when rc= 40 then
   'genalert /t:"A severe error occurred during scanning.",
   /APP:"IBMAVSP" /SEV:0 /TYPE:SYSERR /ATYPE:' rc

when rc= 60 then
   'genalert /t:"An internal error occurred in the program.",
   /APP:"IBMAVSP" /SEV:0 /TYPE:SYSERR /ATYPE:' rc

when rc= 70 then
   'genalert /t:"The program has been modified, possibly by a virus.",
   /APP:"IBMAVSP" /SEV:0 /TYPE:SYSERR /ATYPE:' rc

when rc= 100 then
   'genalert /t:"Viruses were found on a fixed disk, but not in memory. See log file.",
   /APP:"IBMAVSP" /SEV:0 /TYPE:SYSERR /ATYPE:' rc

when rc= 105 then
   'genalert /t:"Viruses were found on a diskette, but not in memory or on fixed disks.,
   see log file",
   /APP:"IBMAVSP" /SEV:0 /TYPE:SYSERR /ATYPE:' rc

when rc= 110 then
   'genalert /t:"Viruses were found on a fixed disk, but not in memory.  See log file.",
   /APP:"IBMAVSP" /SEV:0 /TYPE:SYSERR /ATYPE:' rc

when rc= 120 then
   'genalert /t:"A virus was found in memory, and is probably still active in the system.",
   /APP:"IBMAVSP" /SEV:0 /TYPE:SYSERR /ATYPE:' rc

otherwise nop
end
```

*Figure 216. Genalerts for Antivirus*

### 5.5.3 Popping Up a Window

The ability to pop up a window to notify a user exists in both LMU and in NetFinity 5.0. LMU has a command called LMUPOPUP (LMUPOPW is the Windows version), while NetFinity 5.0 lets you pop up a window using an action in the alert editor. The action in the alert editor is called Notify user with pop-up. You can't define any text or buttons in the NetFinity 5.0 version of the pop-up. The screen below shows the NetFinity 5.0 screen for defining the action.
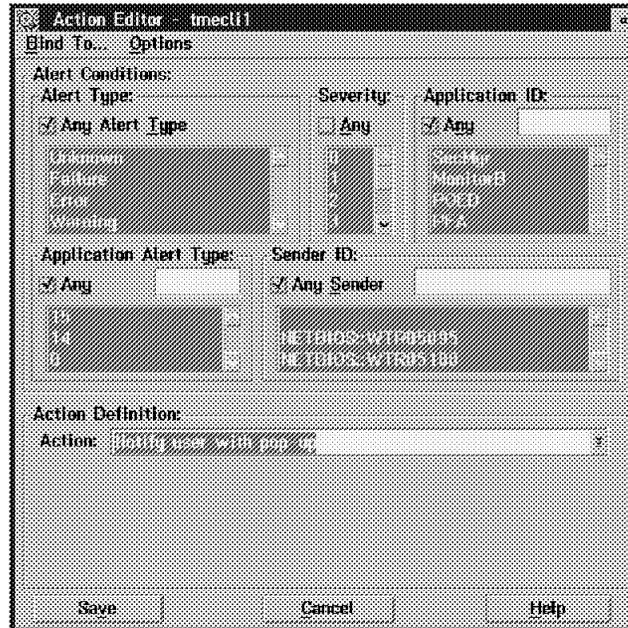


*Figure 217. Action Editor - Notify User with Pop-Up*

The LMUPOPUP has to be called using a command while the NetFinity function is started when an alert arrives and the action is specified in the Alert Manager. See the help for commands by entering LMUPOPUP ? for example.

### 5.5.4 Contact a Pager

There is a pager function in LMU. The LMUPAGE command triggers the LMUPAGER which actually contacts the person's pager with the TAP, IXO, Peopfind and Skytel protocols. In NetFinity 5.0 this is done with an Alert Manager action. Below you see an example screen of this action.
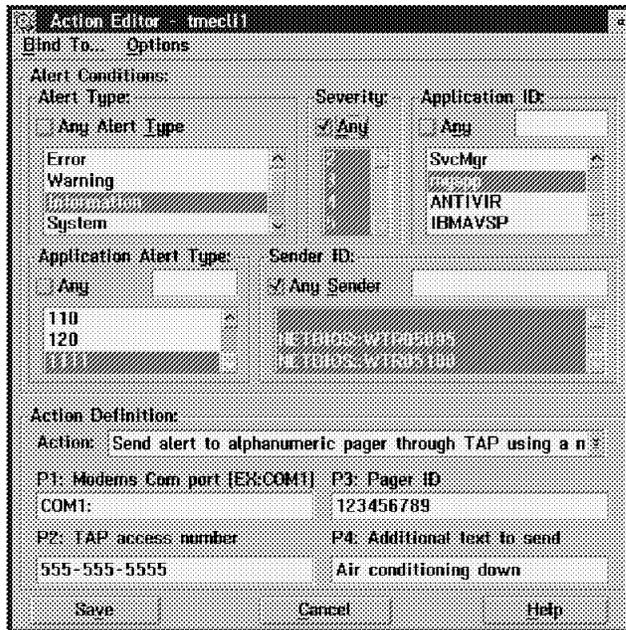
*Figure 218. Action Editor - Send Alert to Alphanumeric Pager through TAP*

> **Note**
>
> In NetFinity 5.0 there is also the ability to define a digital pager for numerical messages only.

In addition, with NetFinity 5.0 you can automatically establish a serial connection with another system to notify a NetFinity 5.0 manager that some problem resolution may be necessary. You can also go the other way and have the NetFinity 5.0 manager automatically dial the client as a result of some alert flowing into the manager.

### 5.5.5 Forwarding Alerts

The forwarding of alerts is done by three LMU components that work together. The first component is called AUECATCH. It uses a user-written fault table and it has an environment variable that is called AUE_DEFAULT. AUECATCH can forward alerts to, for example, NetView for MVS using the OS/2 Communications Manager. The fault table holds the entries for the actions to be taken at specific alerts and indicates if the alert is to be forwarded or not. The default if no entry exists for that alert is to forward it. To change the default you need to modify the environment variable AUE_DEFAULT. If you set this variable to block, the alerts without an entry in the fault table are not forwarded. LMU can forward alerts to three different destinations. One is NetView for MVS which we mentioned before. Another destination can be the IBM LAN Manager or the IBM LAN Network Manager over the IEEE 802.2 interface. The third choice is to operate in LAN Network Manager compatibility mode. The alert is forwarded to a LAN management functional address using the 802.2 interface.

In NetFinity 5.0 there are several ways to forward the alerts. The alerts can also be forwarded to different managers. You can forward the alert to an SNMP Manager. Some examples of that are:

- TME 10 NetView for AIX

You can forward the alerts to NetView for MVS by using the FFST function of MVS and Communications Manager. See *LAN Management Processes Using NetFinity*, SG24-4517 for examples of that.

- NetView for OS/2

- TME 10 Distributed Monitor (Sentry)

- HP OpenView

- NWAYS for Windows

## 5.6 System Variables

Both LMU and NetFinity 5.0 provide a set of informational variables that you can use to set up commands to use in REXX execs.

### 5.6.1 LMU Environment Variables

> **Remark**
>
> The following table is taken from the *LAN NetView Management Utilities for OS/2 V1.21 User's Guide*.

LMU provides a REXX external function to enable REXX programs to have easy access to many LMU-specific characteristics regarding the workstation on which it is running. LMU2RX is the function name and the name of the dynamic link library. LMU2REXX is the name of the procedure. After the REXX function RXFUNCADD has been called, the following variables can be used by the REXX program:

*Table 2 (Page 1 of 2). REXX Accessible Variables*

| Variable | Description |
|---|---|
| lmu_boot_drive | The boot drive of the workstation. |
| lmu_machine_name | The computer name from the IBMLAN.INI file in the workstation for an IBM OS/2 LAN Requester. The internetwork address of the workstation for a Novell NetWare Requester. |
| lmu_domain | The default domain of the workstation for an IBM OS/2 LAN Requester, identified in the workstation's IBMLAN.INI file. |
| lmu_logon_domain | The logon domain of the workstation for an IBM OS/2 LAN Requester. |
| lmu_nw_network_id | The network number for a Novell NetWare Requester. |
| lmu_nw_node | The node address of the workstation for a Novell NetWare Requester. |
| lmu_message_log | The name and location of the LMU message log file for the workstation, identified in the workstation's LMU.INI file. |

| Table 2 (Page 2 of 2). REXX Accessible Variables | |
|---|---|
| **Variable** | **Description** |
| lmu_managing_system | The LMU managing system for the workstation, identified in the workstation's LMU.INI file. |
| lmu_managing_system_with_database | The workstation's LMU managing system with database, identified in the workstation's LMU.INI file. |
| lmu_fault_manager | The workstation's LMU fault manager, identified in the workstation's \ LMU.INI file. |
| lmu_command_source | The computer name of the station that issued the LMUCMD command for an OS/2 LAN Requester. For a Novell NetWare Requester for OS/2, lmu_command_source is the internetwork address of the station that issued the LMUCMD command. This contains a value only if the REXX program running LMU2REXX is initiated through an LMUCMD command (LMUCMD target yourrexx). |
| lmu_command_domain | The current domain of the station that issued the LMUCMD command. If the station that issued the LMUCMD command used the NetBIOS interface to send the command to the target or, in the case of broadcast, to the managing system for delivery to the target, lmu_command_domain contains a domain name. This is true even if the workstation running LMU2REXX is a NetWare Requester. This contains a value only if the REXX program running LMU2REXX is initiated through an LMUCMD command (LMUCMD target yourrexx). |
| lmu_command_userid | The logon ID of the station that issued the LMUCMD command. for a NetWare Requester for OS/2, it is the user name that is logged in to a NetWare server; for an OS/2 LAN Requester, it is the user ID that is logged on to the domain controller. This contains a value only if the REXX program running LMU2REXX is initiated through an LMUCMD command (LMUCMD target yourrexx). |

---

**Note**

LMU2REXX can retrieve the variables lmu_command_source, lmu_command_domain, and lmu_command_userid if LMU2REXX is called from LMUCMDX.CMD. (LMUCMDX.CMD is run automatically from the LMUCMD command.) You can modify LMUCMDX.CMD to call LMU2REXX and to include appropriate processing for these variables, such as performing security checks, providing an audit trail, and other functions.

To use LMU2REXX:

1. Place these lines somewhere in your REXX program before you issue the call to LMU2REXX:

```
if Rxfuncquery('lmu2rx') <> 0 then
    rcy = Rxfuncadd('lmu2rx', 'lmu2rx', 'LMU2REXX')
```

2. Call LMU2REXX with this statement:

```
call lmu2rx
```

> **Note**
>
> The workstation must have access to LMU2RX.DLL.

## 5.6.2 NetFinity 5.0 Command Strings

NetFinity 5.0 also has some variables that can be used on the alert actions: Execute command ' < P 1 > ' and Execute minimized command ' < P 1 > '. These variables are transferred with the command as parameters. The variables are case-sensitive. Below you see an alert action example with these parameters.
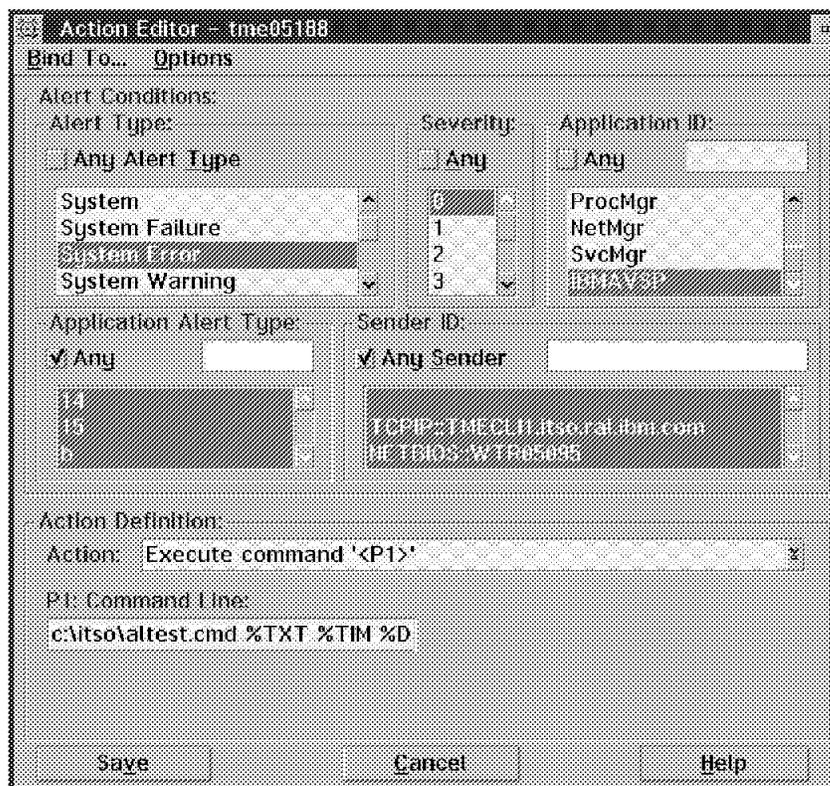


Figure 219. Alert Manager Action - Variables

| Table 3 (Page 1 of 2). NetFinity 5.0 Command Strings | |
|---|---|
| **Command String** | **Embedded Information** |
| %TXT | Alert Text |
| %TIM | Alert Time |
| %DAT | Alert Date |
| %SEV | Alert Severity |

| Table 3 (Page 2 of 2). NetFinity 5.0 Command Strings | |
|---|---|
| **Command String** | **Embedded Information** |
| %SND | Alert Sender |
| %TYP | Alert Type |
| %APP | Alert Application ID |
| %AT | Alert Application Specific Type |
| %P1-%P9 | Alert-specific text strings, that are embedded in the alert text. The content of these parameters is dependant on the alert itself. |

# Appendix A.  Object REXX for Windows 95 Installation

Download Object REXX from the Object REXX home page
http://www2.hursley.ibm.com/orexx/ and save it to a temporary directory (for
example, C:\TEMP).

From your Win95 menu bar select **Start** then **Run**.  Type in your source drive,
path and SETUP.EXE as in the example shown below.  Click on **OK**.
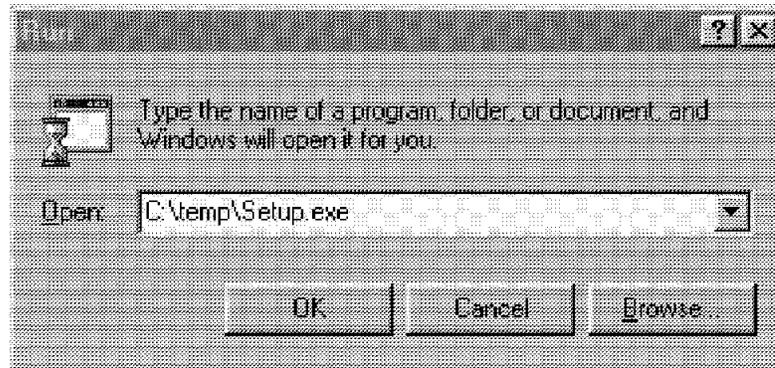


*Figure  220.  Source Path for Object REXX Installation*

Click on **Next** on the Welcome screen.

Read and agree to the license agreement by clicking on the **Yes** button.

The next window lets you define your target directory.  Accept the default or
enter a different one, then click on **Next**.
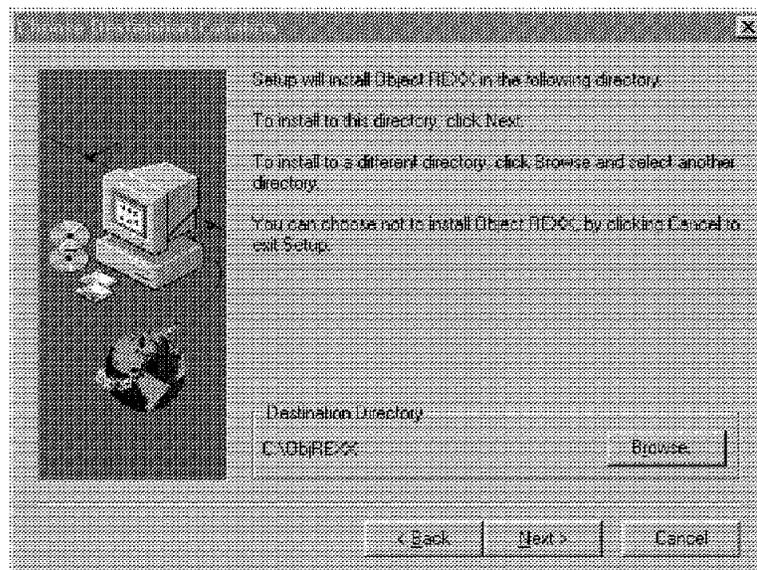


*Figure  221.  Target Directory*

**231**

On the Setup Type window you can define the components to be installed. Use **Typical** unless you want the C++ API Support installed. If you want the API, select **Custom**. Click on **Next**.

The setup program now asks for a program folder. You can use the default or define a different one. Then click on **Next**.
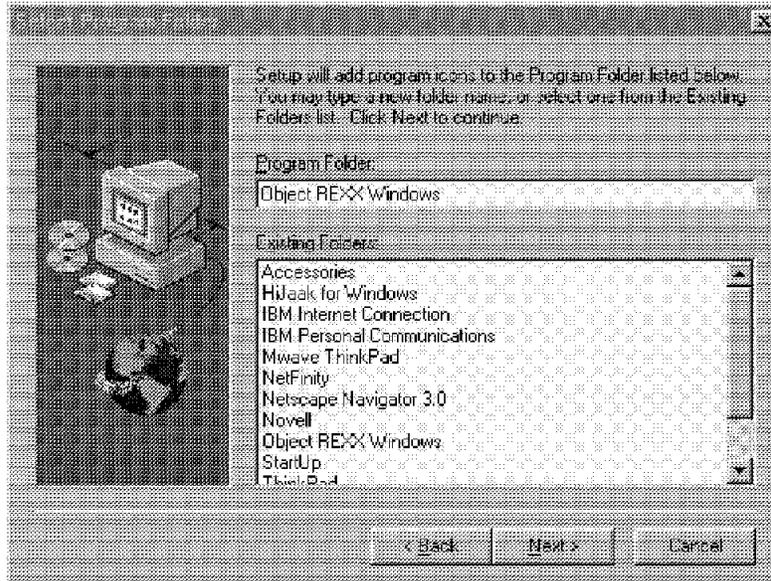


*Figure 222. Program Folder*

This window shows all the options you have selected. If you agree click on **Next** otherwise click on **Back** and do the necessary changes.
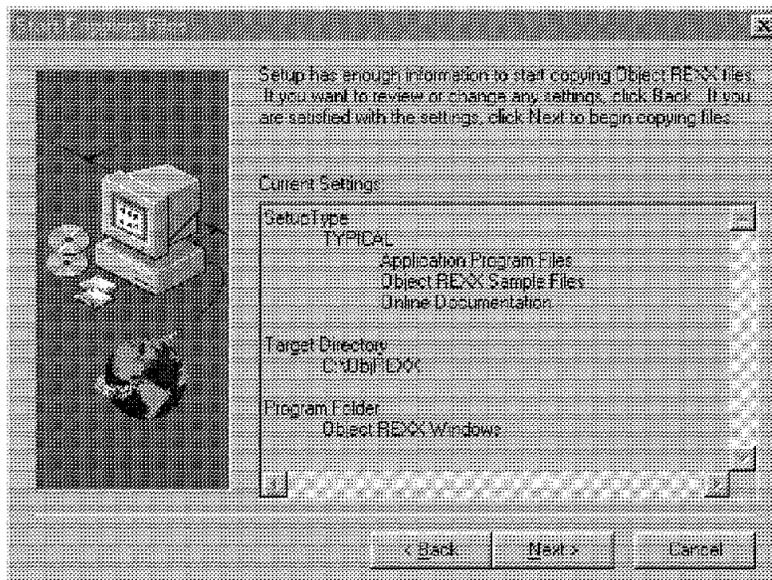


Figure 223. Selections Made

Allow the setup to restart your system so that changes become active. Click on **OK**.
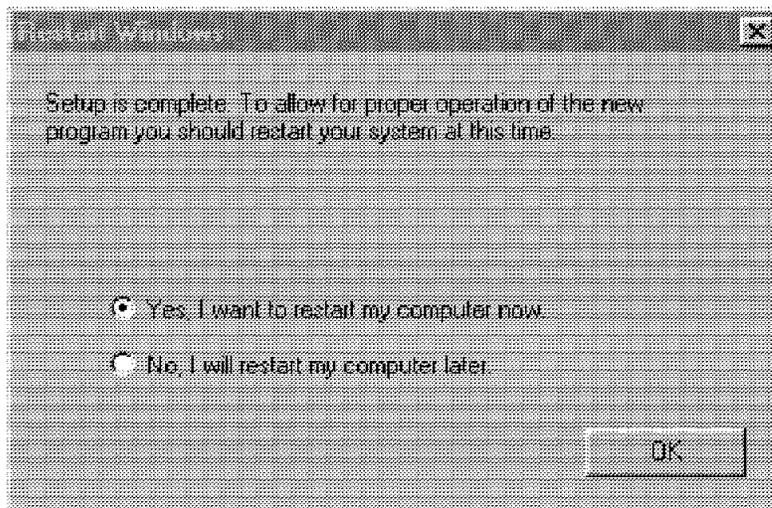


Figure 224. Restart Window

If you need more information about Object REXX, see the home page or use the online documentation. To use the online documentation select **Start** from your menu bar and then **Run**. Enter the drive and path where your Object REXX is installed (see Target Directory) and click on **OK**.
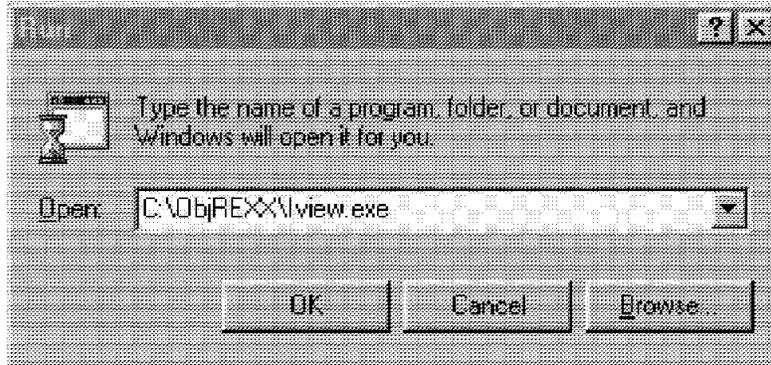


*Figure 225. Run the Online Doc Viewer*

Select the **Books** directory by double-clicking on it.
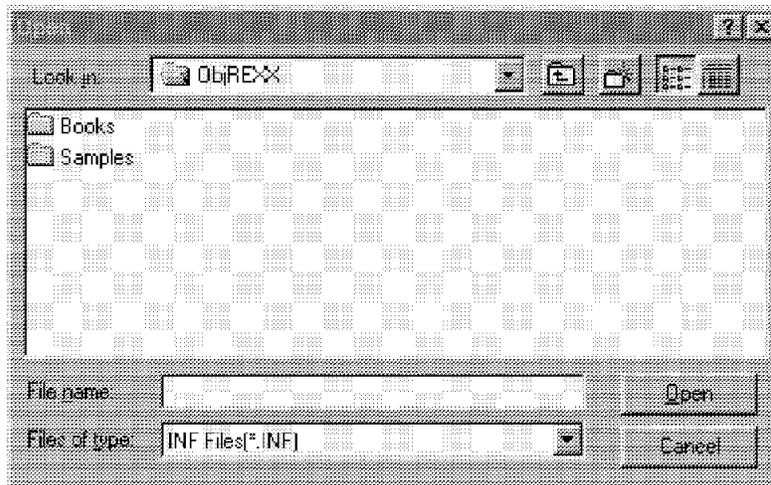


*Figure 226. Books Directory*

Select the appropriate book by simply double-clicking on it. OBJREXXW.INF is the reference book with all the commands, classes, methods and so on. RXGUIDEW.INF is a guide to Object REXX. It also has a part for standard REXX included.
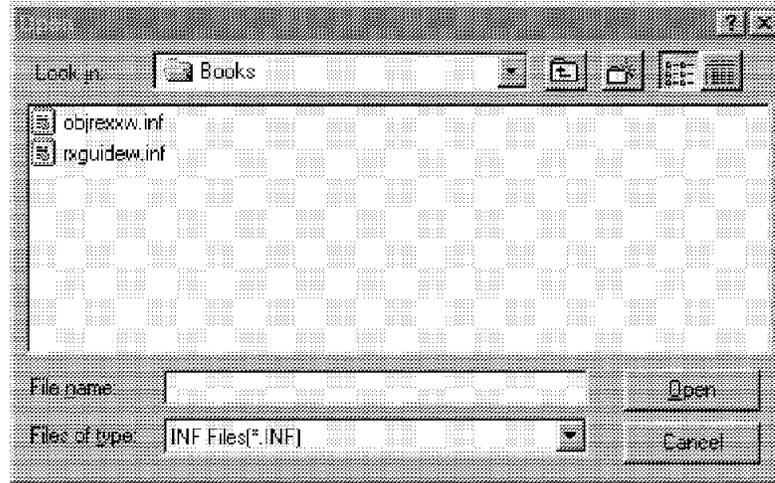


*Figure 227. Select the Book*

# Appendix B.  New Command Line Functions Specifications for NetFinity 5.0

This appendix provides detailed specifications on the new NetFinity command-line interface.

## B.1  Variables

In this appendix the variables used for the syntax diagrams are defined.

| Table 4 (Page 1 of 3).  Variables Used for the Syntax Diagrams | | | RSYS | SEC | PROF | ALRT | SMON | CRTF | PROC | SYSI |
|---|---|---|---|---|---|---|---|---|---|---|
| **Variable** | **Type** | **Description** | | | | | | | | |
| grpname | alphanumeric in quotes | Name of the selected group | X | | | | | | | X |
| sysname | alphanumeric in quotes | Name of the selected system | X | | | | | | | X |
| grouptag | hex(8) | Tag to select a group | X | | | | | | | |
| systag | hex(8) | Tag to select a system | X | | | | | | | |
| proftag | hex(8) | Tag to select a profile | | | | X | | | | |
| logtag | hex(8) | Tag to select a log file | | | | X | | | | |
| MonID | hex(8) | Monitor ID | | | | | X | | | |
| proctag | hex(8) | Tag to select a process | | | | | | | | X |
| protocol | NetBIOS, IPX, TCPIP, SERIPC, SNA_APPC, ... | Protocol to be used for a system | X | | | | | | | |
| sev | 0 ... 7 | Severity of the alert | X | | | X | X | X | X | |
| Action | Any NetFinity CL procedure | Start any NetFinity 5.0 CL function on the remote system | X | | | | | | | |
| Address | Protocol address | Address for a system, according to the used protocol | X | X | | | | | | |
| ping_int | Number in seconds | Interval for pinging remote systems | X | | | | | | | |
| keyword | alphanumeric | Keyword to group systems | X | | | | | | | |
| os_type | OS2, WINDOWS_NT, WINDOWS, NOVELL, ... | Operating system type | X | | | | | | | |
| int_min | number | Interval in minutes | X | | | | | | X | |
| int_sec | number | Interval in seconds | | | | | X | | | |

**237**

| Variable | Type | Description | RSYS | SEC | PROF | ALRT | SMON | CRTF | PROC | SYSI |
|----------|------|-------------|------|-----|------|------|------|------|------|------|
| uid | alphanumeric in quotes | User ID | | X | | | | | | |
| pwd | alphanumeric in quotes | Password | | X | | | | | | |
| srvc | alphanumeric | NetFinity 5.0 Service name | | X | | | | | | |
| attrib | alphanumeric | Attribute to add in system profile | | | X | | | | | |
| value | depends on attrib | Value for a given attribute in system profile | | | X | | | | | |
| path | valid path | Fully qualified path to a file | | | X | | | | X | |
| filename | valid filename | File name without path | | | | | | | X | X |
| handler_id | alphanumeric in quotes | Handler ID for the different alert actions | | | | X | | | | |
| actionname | alphanumeric in quotes | Name for an alert action | | | | X | | | | |
| parm_val | alphanumeric in quotes | Value for the parameter of an alert action | | | | X | | | | |
| profname | alphanumeric in quotes | Name of a alert profile | | | | X | | | | |
| alerttype | alphanumeric(6) | Six letter abbreviation of the alert type | | | | X | | | | |
| app_id | alphanumeric in quotes | ID of the alert sending application | | | | X | | | | |
| appl_alerttype | hex(4) | Application alert type | | | | X | | | | |
| sender | protocol::sysname | Sender ID network name | | | | X | | | | |
| handname | alphanumeric in quotes | Name of the handler | | | | X | | | | |
| monname | alphanumeric in quotes | Name of the monitor | | | | X | | | | |
| errval | numeric | Threshold for a monitor | | | | | X | | | |
| thrname | alphanumeric in quotes | Name for a threshold definition | | | | | X | | | |
| command string | alphanumeric in quotes | Command line procedure to be executed | | | | | | | X | |
| procname | alphanumeric in quotes | Name for a process | | | | | | | X | |
| dbi_name | DB2OS2CI, DB2WINFI, ... | Name of the database interface | | | | | | | | X |

*Table 4 (Page 2 of 3). Variables Used for the Syntax Diagrams*

| Table 4 (Page 3 of 3). Variables Used for the Syntax Diagrams | | | RSYS | SEC | PROF | ALRT | SMON | CRTF | PROC | SYSI |
|---|---|---|---|---|---|---|---|---|---|---|
| Variable | Type | Description | | | | | | | | |
| db_name | alphanumeric in quotes | Name of the database | | | | | | | | X |

## B.2 NetFinity 5.0 Specifications

All NetFinity CLI tools conform to the following specifications:

1. Tools are text-mode executables which direct their outputs to stdout and stderr, and accept all inputs from command-line parameters and (optionally) a textual response file. On Win95 and WinNT, this means that the tools are Win32 Console apps. On OS/2, this means that the tools are VIO apps. Win16 apps are not supported.

2. CLI tools return errorlevels conforming to the following definitions, as appropriate:

   rc=0 (NFCMDLINE_SUCCESS) Operation requested was completed successfully
   rc=1 (NFCMDLINE_SYSTEM_UNAVAILABLE) Target system is unavailable or unreachable
   rc=2 (NFCMDLINE_SERVICE_DOES_NOT_EXIST) Needed service(s) on target system is not available or defined
   rc=3 (NFCMDLINE_NOT_AUTHORIZED) Authorization for needed service(s) on target system is not available
   rc=4 (NFCMDLINE_TIMEOUT) Communication with target system failed due to time-out
   rc=5 (NFCMDLINE_COMMUNICATION_ERROR) Communication with target system failed (non-timeout)
   rc=6 (NFCMDLINE_SERVICE_BUSY) Needed service(s) on target system are available, but currently not available for use
   rc=7 (NFCMDLINE_SVC_NODE_CREATE_FAILED) Attempt to create ServiceNode failed (probably missing DLLs)
   rc=8 (NFCMDLINE_SVC_START_FAILED) Request to start required service(s) on target system failed
   rc=9 (NFCMDLINE_SVC_STOP_FAILED) Request to stop required service(s) on target system failed
   rc=10 (NFCMDLINE_BAD_COMMAND_LINE) Command-line syntax error or invalid input
   rc=11 (NFCMDLINE_FUNCTION_NOT_SUPPORTED) Requested operation is not supported on the service(s) on the target system
   rc=12 (NFCMDLINE_FILE_NOT_FOUND) Requested operation failed due to not finding needed local file
   rc=13 (NFCMDLINE_FILE_OUTPUT_ERROR) Requested operation failed due I/O error to

3. Data output for CLI tools is intended for programmatic processing by other tools. To support this, informational outputs from the CLI tools are as follows:

   a. All information outputs (as opposed to error messages, help messages, logo messages, copyright messages, and other human-oriented outputs) are directed to 'stdout'. All other outputs are directed to 'stderr'.

   b. Information outputs consist of keywords or keyword/value pairs, i.e. MANAGER, ATTRIBNAME=VALUE, TOKEN="string"

   c. Grouped outputs are formatted by listing keyword and keyword/values pairs, separated by commas, spaces, or new lines or consecutive lines, with a blank-line marking the end of each record. Each group of data is enclosed in '{' and '}' brackets. Groups of data can be nested, for example

           { TAG=1234, NAME='xxx', MANAGER, MAC=4000123677 }
           { TAG=1236, NAME='yyy', MAC=4000123678 }
       or
           { TAG=1234
             NAME='xxx'
             MANAGER
             MAC=4000123677 }
           { TAG=1236
             NAME='yyy'
             MAC=4000123678 }
       or
           { TAG=5555, GROUPNAME='aaa'
             SYSTEMS={ { TAG=1234, NAME='xxx', MANAGER, MAC=4000123677 }
                       { TAG=1236, NAME='yyy', MAC=4000123678 } } }

   d. Values in keyword/value pairs are formatted as follows:
      keyword=0xNNNN for hexadecimal values (capital letters for A-F)
      keyword=NNNN for decimal values (a leading '-' or '+' is allowed)

```
            keyword=AAAAAA for keyword-type values (i.e. ON, OFF, ENABLED)
            keyword="string" for string-like values (use two double-quotes for double-quote
            characters within the string. i.e. "")
            keyword=NNN.NNN for floating-point decimal values
            keyword=dd/mm/yyyy for dates (ISO ordering, not local)
            keyword=hh:mm:ss for times (always 24 hour)
            keyword=dd/mm/yyyy:hh:mm:ss for full local timestamp
            keyword=dd/mm/yyyy:hh:mm:ss+off for full universal timestamp (off = minutes from
            UT ("timezone"), use - instead of + if negative delta)
            keyword={ other_keywords_or_values } - for collections of values or attributes
      Only "string" fields contain non-ASCII characters (as well as translated or country-local
      values).
      e. Keywords start with an ASCII letter ('A'-'Z', 'a'-'z', '_'), followed by ASCII
         alphanumeric characters ('0'-'9', 'A'-'Z', 'a'-'z', '_').
      f. Commas, spaces, new-lines (CR, LF), and tab characters, when not enclosed in
         "string" fields, are treated as interchangeable separators.
      g. "String" fields are expressed in the local codepage of the system running the CLI tool.
         The caller of the tool is expected to provide any required codepage mapping of this
         data.
   4. Command-line parameters are structured as follows:
      a. Each command-line parameter must have a leading '/' or '-' character, followed by a
         case-insensitive ASCII keyword.  If a value is to be provided with the keyword, the
         keyword will be followed by a colon and the value (which may be enclosed in
         double-quotes, as needed).
         For example:
         /ADDSYS, /DELSYS:name, -REMSYS:"name with spaces"
      b. Textual values on the command line must be provided in the local codepage of the
         command-line tool.  The caller of the tools is expected to provide any required
         codepage mapping of these inputs.
      c. All CLI tools support the following command-line values, with the following default
         behaviors if the parameter is not provided:
         1. /N:netpath - Provides NetFinity-style network path to target system.  For example:
            /N:NETBIOS::PRIMM,
            or
            /N:TCPIP::primm.raleigh.ibm.com::NETBIOS::NetFin
            If not provided, local system is target system.
         2. /S:"sysname" - Provides optional system name for target system.  For exapmle:
            /S:"Mike's system"
            No optional system name, if not provided.
         3. /? - Print command-line help for tool (if any)
         4. Standard parameters are not order sensitive, and tool-specific parameters are not
            order-specific.
         5. The presence of any unsupported or inappropriate parameters results in no action,
            with a return code of NFCMDLINE_BAD_COMMAND_LINE.
         6. Response file support may optionally be provided by a tool.  The response file must
            be provided with a leading '@'.  The contents of the response file are processed as
            if they were substituted for the response file option on the command line (with
            new-line and tab characters replaced with spaces).
         7. Command-line help, logos, copyright statements, error messages, and other
            human-oriented outputs are directed to 'stderr', not ' stdout' (to prevent any
            difficulties in parsing the informational output directed to 'stdout', particularly due
            to NLS issues and the like).
o NFRSYSCL (CLI tool for Remote System Manager)
o NFSECCL (CLI tool for Security Manager)
o NFPROFCL (CLI tool for System Profile)
o NFALRTCL (CLI tool for Alert Manager)
o NFMONCL (CLI tool for System Monitor)
o NFCRTFCL (CLI tool for Critical File Monitor)
o NFPROCCL (CLI tool for Process Manager)
o NFSYSICL (CLI tool for System Information Tool)
o NFREPLCL (CLI tool for Service Configuration Manager)
NFRSYSCL /? - Command line help
<grpsel> = one of the following parameters
        /ALL - Select all groups
        /GRP:"groupname" - Select group(s) with the given name
        /GRPTAG:NNNNNNNN - Select group with the given group tag (in hex)
<syssel> = one of the following parameters
        /ALL - Select all systems
        /GRP:"groupname" - Select systems which are in the group(s) with
                the given name
        /GRPTAG:NNNNNNNN - Select systems which are in the group(s) with
                the given group tag (in hex)
        /SYS:"sysname" - Select system(s) with the given system name
        /SYSTAG:NNNNNNNN - Select system with the given system tag (in hex)
NFRSYSCL /GETGRP <grpsel> - List selected system groups attributes
```

```
                Output:
                   For each group:
                   { GRPTAG=0xNNNNNNNN, GRPNAME="group name", COMBO=xxx,
                     KWD={ kwd, ... }, OS_MASK={os, ...}, AUTODISC=minutes,
                     PROTO_MASK={proto,...}, DEFONLN=online_note,
                     DEFOFFLN=offline_note, DEFPING=ping_int, others }
                    where GRPTAG value is hexadecimal identifier for group,
                          GRPNAME value is string name for group,
                          COMBO value is keyword combo code (ALL, ANY, ONE),
                          KWD value is bracketted list of string keyword values,
                          OS_MASK value is bracketted list of OS types to exclude
                              from discovery.  OS2, WINDOWS, NETWARE, WINDOWS_NT, AIX,
                              VINES, DOS, UNIX, OS_400, WINDOWS_95
                          PROTO_MASK value is bracketted list of protocol types
                              to exclude from discovery.  NETBIOS, TCPIP, IPX,
                              SERIPC, SNA
                          AUTODISC value is minutes between each autodiscovery, or
                              NONE if not enabled.  Attribute not returned on managers
                              which don't support feature.
                          DEFONLN value is default online notify severity (0-7,
                              DISABLED, or NONE)
                          DEFOFFLN value is default offline notify severity (0-7,
                              DISABLED, or NONE)
                          DEFPING value is default ping interval in seconds (15+,
                              or NONE)
NFRSYSCL /GETSYS <syssel> - List system attributes of selected systems
                Output:
                   For each system:
                   { SYSTAG=0xNNNNNNNN, SYSNAME="system name", PROTO=protocol,
                     ADDR=address, ONLINE=status, ONLN=online_notify,
                     OFFLN=offline_notify, PING=ping_interval, GRPLIST={ grp_tags,
                     ... }, ERRORCOND={ "error_cond_str", ... }, OS=os_type,
                     OSVER=x.xx, MANAGER, SERVER, MAC=mac_addr, SHUTDOWN,
                     POWERDOWN, WAKEONLAN, WEBMGR, others }
                    where SYSTAG value is hexadecimal identifier for system,
                          SYSNAME value is string name for system,
                          PROTO value is protocol used to communicate with system
                              (NETBIOS, TCPIP, IPX, SERIPC, SNA, others)
                          ADDR value is textual address of system on PROTO protocol
                          ONLINE value is TRUE if system is online, FALSE if offline
                          ONLN value is severity of on-line notification alert (0-7)
                              or DISABLED if not enabled
                          OFFLN value is severity of off-line notification alert
                              (0-7) or DISABLED if not enabled
                          PING value is seconds between system pings (15+)
                          GRPLIST value is bracketted list of hex tag values for
                              groups which system is member of.
                          ERRORCODE value is bracketted list of "strings" for each
                              error condition currently defined for system
                          OS value is type and version of operating system of system:
                              UNKNOWN, OS2, WINDOWS, NETWARE, WINDOWS_NT, AIX,
                              VINES, DOS, UNIX, OS_400, WINDOWS_95
                          MANAGER keyword is present if system is NetFinity manager
                          SERVER keyword is present if system is a server of some sort
                          MAC value is a hex number for the system's MAC address, or
                              UNKNOWN if not known
                          SHUTDOWN keyword is present if system supports remote
                              shutdown
                          POWERDOWN keyword is present if system supports remote
                              powerdown
                          WAKEONLAN keyword is present if system supports remote
                              wake-up on LAN
                          WEBMGR keyword is present if NetFinity Web manager is
                              active on the system
                          UNIQUE_ID value, if present, indicates a 16-digit
                              hexadecimal value provided by the client as a unique
                              system ID (the same system appearing under different
                              protocols will have the same UNIQUE_ID value)
NFRSYSCL /RUNSYS <syssel> /ACTION:rest of line - Run action on all
                         selected systems
              Output:
                   For each system:
                   { SYSTAG=0xNNNNNNNN, SYSNAME="system name", PROTO=protocol,
                     ADDR=address, ONLINE=status, OUTPUT={ action output },
                     RC=ret_code, others }
```

```
                            where SYSTAG value is hexadecimal identifier for system,
                                  SYSNAME value is string name for system,
                                  PROTO value is protocol used to communicate with system
                                      (NETBIOS, TCPIP, IPX, SERIPC, SNA, others)
                                  ADDR value is textual address of system on PROTO protocol
                                  ONLINE value is TRUE if system is online, FALSE if offline
                                  OUTPUT value is bracketted output from running tool
                                      selected by ACTION against given system.  See the spec
                                      for the executed tool for format of output
                                  RC value is errorlevel returned by running tool specified
                                      by ACTION against given system
                        Note: This operation only supports running other NetFinity
                            Manager executables which conform to this specification.  It
                            is not for running programs on the targetted systems.
                            See NFPROCCL for remote task execution.
NFRSYSCL /DELSYS <syssel> - Delete all selected systems
            Output:
                SYSTAG=0xNNNNNNNN
                 where SYSTAG value is hexadecimal tag of deleted system(s)
NFRSYSCL /ADDSYS:"systemname" /PROTO:protocol /ADDR:address
            /ONLN:online_sev /OFFLN:offline_sev /PING:ping_int
            /ADDGRP:"group name" /ADDGRPTAG:group_tag - Create new system
                with given attribs
            Input:
                where ADDSYS value is name string for system (required),
                      PROTO value is protocol to communicate with system
                          (required) Values include NETBIOS, IPX, TCPIP,
                          SERIPC, SNA, and others
                      ADDR value is protocol address for system (required)
                      ONLN value is on-line notify alert severity (0-7) or
                          DISABLED (option) Defaults to DISABLED
                      OFFLN value is on-line notify alert severity (0-7) or
                          DISABLED (option) Defaults to DISABLED
                      PING value is ping interval for system in seconds
                          (optional) Defaults to 600 seconds.
                      ADDGRP value is name of group(s) which system will be
                          added to (one ADDGRP or ADDGRPTAG required, others
                          optional).  More than one ADDGRP or ADDGRPTAG parameter
                          is allowed.
                      ADDGRPTAG value is hex tag of group which system will be
                          added to (one ADDGRP or ADDGRPTAG required, others
                          optional).  More than one ADDGRP or ADDGRPTAG parameter
                          is allowed.
            Output:
                SYSTAG=0xNNNNNNNN
                 where SYSTAG value is hexadecimal tag assigned to new system
NFRSYSCL /EDITSYS <syssel> /PROTO:protocol /ADDR:address
            /ONLN:online_sev /OFFLN:offline_sev /PING:ping_int /NEWNAME:"name"
            /ADDGRP:"group name" /ADDGRPTAG:group_tag /DELGRP:"group name"
            /DELGRPTAG:group_tag - Modify system attributes
            Input:
                where PROTO value is new protocol for system (optional)
                          Values include NETBIOS, IPX, TCPIP, SERIPC, SNA,
                          and others
                      ADDR value is new protocol address for system (optional)
                      NEWNAME value is new system name for system (optional)
                      ONLN value is new on-line notify alert severity (0-7)
                          or DISABLED (optional)
                      OFFLN value is on-line notify alert severity (0-7)
                          or DISABLED (optional)
                      PING value is ping interval for system in seconds
                          (optional)
                      ADDGRP value is name of group(s) which system will be
                          added to (optional)
                      ADDGRPTAG value is hex tag of group which system will
                          be added to (optional)
                      DELGRP value is name of group(s) which system will be
                          removed from (optional)
                      DELGRPTAG value is hex tag of group which system will
                          be removed from (optional)
            Output:
                SYSTAG=0xNNNNNNNN
                 where SYSTAG value is hexadecimal tag of each modified system
NFRSYSCL /DELGROUP <grpsel> - Delete all selected groups
            Output:
```

```
                        GRPTAG=0xNNNNNNNN
                          where GRPTAG value is hexadecimal of deleted group(s)
NFRSYSCL /ADDGROUP:"groupname" /COMBO:combo_code /ADDKWD:kwd /ADDOS:os_type
                /ADDPROTO:protocol /AUTODISC:interval /DEFONLN:online_note
                /DEFOFFLN:offline_note /DEFPING:ping_int - Create new group
                            with given attribs
            Input:
                where ADDGROUP value is name string for group (required)
                        COMBO value is keyword combination code (ALL, ANY, ONE)
                            (optional, default is ALL)
                        ADDKWD value is keyword string to add for group (optional)
                        ADDOS value is OS type to add to discovery exclusion list
                            (optional)
                        ADDPROTO value is protocol type to add to discovery
                            exclusion list (optional)
                        AUTODISC value is decimal minutes between autodiscovery
                            in group (optional, default is NONE)
                        DEFONLN value is default online notify severity
                            (0-7, DISABLED, NONE) (optional, default is NONE)
                        DEFOFFLN value is default offline notify severity
                            (0-7, DISABLED, NONE) (optional, default is NONE)
                        DEFPING value is default ping interval, in seconds
                            (15+, NONE) (optional, default is NONE)
            Output:
                GRPTAG=0xNNNNNNNN
                  where GRPTAG value is hexadecimal assigned to new group
NFRSYSCL /EDITGRP <grpsel> /COMBO:combo_code /NEWNAME:newname
                /ADDKWD:kwd /DELKWD:kwd /AUTODISC:interval /DEFONLN:online_note
                /DEFOFFLN:offline_note /DEFPING:ping_int /ADDOS:os_type
                /DELOS:os_type /ADDPROTO:protocol /DELPROTO:protocol - Modify
                    all selected group(s)
            Input:
                where COMBO value is new keyword combination code (ALL, ANY, ONE)
                            (optional)
                        ADDKWD value is additional keyword string for group
                            (optional, only allows 8 keywords total)
                        DELKWD value is keyword string to remove from group
                            (optional)
                        ADDOS value is OS type to add to discovery exclusion
                            list (optional)
                        DELOS value is OS type to remove from discovery
                            exclusion list (optional)
                        ADDPROTO value is protocol type to add to discovery
                            exclusion list (optional)
                        DELPROTO value is protocol type to remove from discovery
                            exclusion list (optional)
                        AUTODISC value is new decimal minutes between autodiscovery
                            in group. (optional)
                        DEFONLN value is new default online notify severity
                            (0-7, DISABLED, NONE) (optional)
                        DEFOFFLN value is new default offline notify severity
                            (0-7, DISABLED, NONE) (optional)
                        DEFPING value is new default ping interval, in seconds
                            (15+, NONE) (optional)
            Output:
                GRPTAG=0xNNNNNNNN
                  where GRPTAG value is hexadecimal tag of mofified group
NFRSYSCL /BOOTSYS <syssel> - Reboot all selected systems
NFRSYSCL /SHUTSYS <syssel> - Shutdown all selected systems
NFRSYSCL /PWRSYS <syssel> - Powerdown all selected systems
NFRSYSCL /WAKESYS <syssel> - Wake-on-LAN all selected systems
            Output: for each targetted system
                { SYSTAG=0xNNNNNNNN, RC=ret_code }
                  where SYSTAG value is hexadecimal tag of system,
                        RC value is standard errorlevel for action on given system
NFRSYSCL /RESETERRSYS <syssel> - Reset error conditions on all selected
                                 systems
            Output: for each targetted system
                SYSTAG=0xNNNNNNNN
                  where SYSTAG value is hexadecimal tag of system
NFRSYSCL /DODISC <grpsel> - Start discovery in all selected groups
NFRSYSCL supports the /N tag on all commands for selecting the target
            remote system manager service to interact with (i.e. "passthru"
            management)
NFRSYSCL /DOPING <syssel> - Initiate presence check on all selected systems
```

```
                          Output: for each targetted system
                               SYSTAG=0xNNNNNNNN
                               where SYSTAG value is hexadecimal tag of system
NFRSYSCL /GETSYSKWD <syssel> - List system's configured keywords
                          Output: for each targetted system
                            { SYSTAG=0xNNNNNNNN, RC=retcode, KWD={ "kwd1", "kwd2", ... } }
                            where SYSTAG values is hexadecimal tag of system,
                                   RC value is return code for operation on specific system,
                                   KWD value is list of keyword strings (if successful)
NFRSYSCL /SETSYSKWD <syssel> /ADDKWD:keywd /DELKWD:keywd - Add or
                          delete keywords from selected system
                          Input:
                              where ADDKWD value is additional keyword string for group
                                     (optional, only allows 8 keywords total)
                                    DELKWD value is keyword string to remove from group
                                     (optional)
                          Output: for each targetted system
                            { SYSTAG=0xNNNNNNNN, RC=retcode }
                            where SYSTAG values is hexadecimal tag of system,
                                   RC value is return code of action result on system
     Tool-specific return codes:
          rc=200 - Group name not defined
          rc=201 - System name not defined
          rc=202 - Group tag not defined
          rc=203 - System tag not defined
          rc=204 - Bad OS type
          rc=205 - Bad protocol type
          rc=206 - Bad ping interval
          rc=207 - Bad notification severity
          rc=208 - Bad autodiscovery interval
NFSECCL /? - Command line help
NFSECCL /LISTIN /ALL - List attribs for all inbound user-ids
NFSECCL /LISTIN /USERID:"userid" - List attribs for given user-id
            Output:
                { USERID="userid", PWD="passwd", SECMGR, SVC={"svcname",...} }
                 where USERID value is user-id string,
                       PWD value is password string ('*' for public),
                       SECMGR keyword is present if security manager access is
                           defined for given user-ID,
                       SVC value is bracketted list of strings for service IDs
                           (base servicenode names) or ALL for all services

NFSECCL /DELIN /ALL - Delete all inbound user-ids
NFSECCL /DELIN /USERID:"userid" - Delete inbound user-id
NFSECCL /ADDIN:"userid" /PWD:"password" /ADDSECMGR /ADDSVC:"svc"
                  /IGNORESVC - Add new inbound user-id
            Input:
                where PWD value is password (required)
                      ADDSECMGR adds security manager access (optional),
                          default is no SECMGR set
                      ADDSVC adds access to a service (optional, multiple OK),
                          either service ID string or ALL for all services
                      IGNORESVC ignores unknown services in ADDSVC
                          parameters (optional, default is error if unknown)
NFSECCL /EDITIN:"userid" /PWD:"password" /ADDSECMGR /DELSECMGR
              /ADDSVC:"svc" /DELSVC:"svc" /IGNORESVC - Modify inbound user-id
            Input:
                where PWD value is new password (optional)
                      ADDSECMGR adds security manager access (optional)
                      DELSECMGR removes security manager access (optional)
                      ADDSVC adds access to a service (optional, multiple OK),
                          either service ID string or ALL for all services
                      DELSVC removes access to a service (optional, multiple OK),
                          either service ID string or ALL for all services
                      IGNORESVC ignores unknown services in ADDSVC and DELSVC
                          parameters (optional, default is error if unknown)
NFSECCL /LISTOUT /ALL - List attribs for all outbound host-ids
NFSECCL /LISTOUT /HOST:"host" - List attribs for given host
            Output:
                { HOST="host", USERID="userid", PWD="passwd" }
                 where HOST value is host-id string,
                       USERID value is user-id string,
                       PWD value is password string,
NFSECCL /DELOUT /ALL - Delete all outbound host-ids
NFSECCL /DELOUT /HOST:"host" - Delete outbound host-id
```

```
NFSECCL /ADDOUT:"host" /SETUID:"userid" /PWD:"password" - Add new
                                 outbound host-id
         Input: where SETUID value is user-id (required)
                     PWD value is password (required)
NFSECCL /EDITOUT /ALL /SETUID:"userid" /PWD:"password" - Modify
                                 outbound host-id
NFSECCL /EDITOUT /HOST:"host" /SETUID:"userid" /PWD:"password" - Modify
                                 outbound host-id
         Input: where SETUID value is new user-id string (optional),
                     PWD value is new password (optional)
NFSECCL supports the /N tag on all commands for selecting the target
          security manager service to interact with.
   Tool-specific return codes:
        rc=200 - Inbound user-ID not defined
        rc=201 - Outbound node not defined
        rc=202 - Bad password
        rc=203 - Bad user-ID
        rc=204 - Bad service
NFPROFCL /? - Command line help
NFPROFCL /GETALL - Get all attributes and their values
         Output: for each attribute
            attrib="value"
                where attrib is name of attribute
                     value is string, date, or time
NFPROFCL /GET:attrib - Get given attribute and its value
         Output:
            attrib="value"
                where attrib is name of attribute
                     value is string, date, or time
NFPROFCL /SET:attrib /SETVAL:value - Set attribute to value
         Input:
            where attrib is name of attribute
                value is string with maximum 32 characters (rest will be
                          truncated), date (dd-mm-yyyy), or time (hh:mm:ss)
NFPROFCL /SETMANY /INPUT:filename - Set attributes to values in input file
         Input file format: one line for each attribute
            attrib="value"
            where attrib is name of attribute
                value is string with maximum 32 characters (rest will be
                          truncated), date (dd-mm-yyyy), or time (hh:mm:ss)
         For example:
            firstname="Vicki"
            startdate="31/12/1995"
            starttime="14:30:00"
NFPROFCL supports the /N tag on all commands for selecting the target
          system profile service to interact with.
   Tool-specific return codes:
        rc=200 - Attribute missing
        rc=201 - Attribute invalid
        rc=202 - Value missing
        rc=203 - Value invalid
        rc=204 - Input file missing
        rc=205 - Input file format error
        rc=206 - Date format error
        rc=207 - Date invalid
        rc=208 - Date out-of-range
        rc=209 - Time format error
        rc=210 - Time invalid
        rc=211 - Time out-of-range
NFALRTCL /? - Command line help
<actsel> = either /ALL, /TAG, or /HASHAND (with other /HAS* parms optional)
             where ALL selects all configured actions
                   TAG value is the hex identifier of a specific action
                   HASHAND value is handler ID of actions to select,
                   HASPARM# value is value of parm # (0-3) of actions to
                       select (optional),
                   HASPROFTAG value is hex profile tag possessed by
                       actions to select (optional, multiple allowed, with
                       only actions which contain all provided profile
                       tags being selected)
                   HASPROF value is name of profile used by
                       actions to select (optional, multiple allowed, with
                       only actions which contain all provided profile
                       names being selected)
                   HASTYPE value is standard alert type matched by
```

```
                                      actions to select (optional, multiple allowed, with
                                      only actions which match all provided
                                      types being selected (actions which match on "any"
                                      type are only selected with /HAStype:ANY))
                              HASAPP value is application IDs matched by actions to
                                      select (optional, multiple allowed, with
                                      only actions which match all provided IDs
                                      being selected (actions which match on "any"
                                      type are only selected with /HASAPP:ANY))
                              HASATYPE value is hex application
                                      alert types matched by actions to select (optional,
                                      multiple allowed, with only actions which match
                                      all provided types being selected (actions
                                      which match on "any" application alert
                                      type are only selected with /HASAtype:ANY))
                              HASSEV value is alert severity matched by actions
                                      to select (optional, multiple allowed, with only
                                      actions which match all provided severities
                                      being selected (actions which match on "any"
                                      severity are only selected with /HASSEV:ANY))
                              HASSENDER value is sender path matched by actions
                                      to select (optional, multiple allowed, with only
                                      actions which match all provided sender paths
                                      being selected (actions which match on "any"
                                      sender are only selected with /HASSENDER:ANY))
            <profsel> = one of the following
                          /ALL - Select all profiles
                          /TAG:NNNNNNNN - Select profile with given tag
                          /NAME:"profname" - Select profile(s) with given name

            <logsel> =  one of the following
                          /ALL - Select all alerts in log
                          /TAG:NNNNNNNN - Select alert with given tag

            <handsel> = one of the following
                          /ALL - Select all handlers
                          /NAME:"handname" - Select handler with given ID string

            NFALRTCL /LISTLOG <logsel> - List all selected alerts in alert log
                    Output: for each selected alert in log
                        { TAG=0xNNNNNNNN, TEXT="alert text", TIME=alert_time,
                          DATE=alert_date, SEV=severity, TYPE=alrt_type,
                          APP="app_id", ATYPE=0xNNNN, SYSNAME="system name",
                          SENDER="sender netpath", P1="parm1", others }
                          where TAG value is identifier for log entry,
                                TEXT value is string for alert text,
                                TIME value is time attribute of alert (hh:mm:ss),
                                DATE value is date attribute of alert (dd-mm-yyyy),
                                SEV value is severity of alert (0-7),
                                TYPE value is standard alert type: format is two
                                    3-character sequences (as used in GENALERT): xxxyyy
                                    where xxx is alert type:
                                        UNK  - Unknown
                                        SYS  - System
                                        DSK  - Disk or DASD
                                        NET  - Network
                                        OS_  - Operating System
                                        APP  - Application
                                        DEV  - Device
                                        SEC  - Security
                                    and yyy is alert class:
                                        UNK  - Unknown
                                        FLT  - Fault or Failure
                                        ERR  - Error
                                        WRN  - Warning
                                        INF  - Information
                                APP value is string for application ID of alert,
                                ATYPE value is hex number of application alert type,
                                SYSNAME value is string for system name,
                                SENDER value is string for alert sender's path,
                                P# value is string for alert parameter # (#=1-9)

            NFALRTCL /DELLOG <logsel> - Delete all selected alerts in alert log

            NFALRTCL /LISTHAND <handsel> - List all selected alert handlers
```

```
Output: for each selected alert handler
    { HANDLER="handler_id", NAME="handler name",
     PARMLAB0="parm label", PARMLAB1="parm label", others }
    where HANDLER value is string identifier for alert handler:
            "WEBACT/0" - Send alert as TCP/IP Web mail
            "SETERRORCONDITION" - Set error condition for
                            sending system
            "CLEARERRORCONDITION" - Clear error condition for
                            sending system
            "DMIACT/0" - Semd DMI Event through DMI Service Layer
            "ALERTLOG" - Add alert to log file
            "ALERTFORWARD" - Forward alert through network
            "ALERTPOPUP" - Notify user with pop-up
            "RUNCOMMAND" - Execute command
            "RUNMINCOMMAND" - Execute minimized command
            "PAGERACT/0" - Activate numeric pager
            "PAGERACT/1" - Send to alphanumeric pager
            "TCPIPACT/0" - Send SNMP alert
            "TCPIPACT/1" - Send TCP/IP mail
            "TCPIPACT/2" - Send mapped SNMP alert
            others: name is uppercase DLL name, followed by '/',
                    followed by entry point number (AlertHandler#).
        NAME value is string for alert label
        PARMLAB# value is string label for alert parameter (0-3)

NFALRTCL /LISTACT <actsel> - List all selected alert actions
    Output: for each selected alert action
        { TAG=0xNNNNNNNN, NAME="name", HANDLER="handler_id",
         PARM0="parm value", PARM1="parm value",
         PROFTAG={ 0xNNNNNNNN, ... }, PROF={"profname", ... },
         TYPES={ alert_type, ... }, APPS={ "app_id", ... },
         ATYPES={ atype, ... }, SEVS={ sev, ... },
         SENDERS={ "sender", ... }, others }
        where TAG value is unique identifier for action,
            NAME value is string label (on profile actions),
            HANDLER value is string identifier for alert handler
            PARM# value is string value for parameter # (0-3)
            PROFTAG value is a bracketted list of hex tag IDs for
                profiles used to match with action (if used)
            PROF value is a bracketted list of strings for
                profile names used to match with action (if used)
            TYPES value is a bracketted list of standard alert types,
                with an empty list indicating ANY.  Only present if
                profiles not used for action
            APPS value is a bracketted list of application ID strings,
                with an empty list indicating ANY.  Only present if
                profiles not used for action
            ATYPES value is a bracketted list of hex numbers for
                application alert types, with an empty list
                indicating ANY.  Only present if profiles not used
                for action
            SEVS value is a bracketted list of alert severities,
                with an empty list indicating ANY.  Only present if
                profiles not used for action
            SENDERS value is a bracketted list of sender path strings,
                with an empty list indicating ANY.  Only present if
                profiles not used for action

NFALRTCL /DELACT <actsel> - Delete all selected alert actions
    Output: for each selected action
        TAG=0xNNNNNNNN
            where TAG is unique tag for deleted action

NFALRTCL /ADDPACT:"handler_id" /NEWNAME:"name" /PARM0:"parm val"
        /PARM1:"parm val" /ADDPROFTAG:NNNNNNNN /ADDPROF:"profname" -
            Add alert action based on profile(s)
NFALRTCL /ADDACT:"handler_id" /PARM0:"parm val" /PARM1:"parm val"
        /ADDtype:xxxyyy /ADDAPP:"app_id" /ADDAtype:NNNN /ADDSEV:sev
        /ADDSENDER:"sender" - Add alert action based on matching rules
    Input:
        where ADDACT value is handler ID for action,
            ADDPROFACT value is handler ID for action,
            PARM# value is value for parameter # of action (0-3)
            NEWNAME value is string label for profile-based action
            ADDPROFTAG value is hex tag for profile to add to
```

```
                              list of profiles for action (can use more than once)
                          ADDPROF value is string name of profile to add to
                              list of profiles for action (can use more than once)
                          ADDTYPE value is standard alert type to add to
                              list of matching alert types (can use more than once)
                          ADDAPP value is application ID string to add to
                              list of matching app IDs (can use more than once)
                          ADDATYPE value is hex application alert type to add
                              to matching list (can use more than once)
                          ADDSEV value is alert severity (0-7) to add to
                              matching list (can use more than once)
                          ADDSENDER value is sender path string to add to
                              matching list (can use more than once)
                Output:
                   TAG=0xNNNNNNNN
                        where TAG is unique tag assigned to new action

     NFALRTCL /EDITPACT <actsel> /NEWHAND:"handler_id"
                /PARM0:"parm val" /PARM1:"parm val" /ADDPROFTAG:NNNNNNNN
                /DELPROFTAG:NNNNNNNN /ADDPROF:"profname" /DEFPROF:"profname"
                   - Edit alert action (profile) with given tag
     NFALRTCL /EDITACT <actsel> /NEWHAND:"handler_id"
                /PARM0:"parm val" /PARM1:"parm val" /ADDtype:xxxyyy
                /ADDAPP:"app_id" /ADDAtype:NNNN /ADDSEV:sev /ADDSENDER:"sender"
                /DELtype:xxxyyy /DELAPP:"app_id" /DELAtype:NNNN
                /DELSEV:sev /DELSENDER:"sender" - Edit alert action with given tag
                Input:
                   where NEWHAND value is new handler ID for action,
                          PARM# value is new value for parameter # of action (0-3)
                          ADDPROFTAG value is hex tag for profile to add to
                              list of profiles for action (can use more than once)
                          DELPROFTAG value is hex tag for profile to remove from
                              list of profiles for action (can use more than once)
                          ADDPROF value is string name of profile to add to
                              list of profiles for action (can use more than once)
                          DELPROF value is string name of profile to remove from
                              list of profiles for action (can use more than once)
                          ADDTYPE value is standard alert type to add to
                              list of matching alert types (can use more than once)
                          DELTYPE value is standard alert type to remove from
                              list of matching alert types (can use more than once)
                          ADDAPP value is application ID string to add to
                              list of matching app IDs (can use more than once)
                          DELAPP value is application ID string to remove from
                              list of matching app IDs (can use more than once)
                          ADDATYPE value is hex application alert type to add
                              to matching list (can use more than once)
                          DELATYPE value is hex application alert type to remove
                              from matching list (can use more than once)
                          ADDSEV value is alert severity (0-7) to add to
                              matching list (can use more than once)
                          DELSEV value is alert severity (0-7) to remove from
                              matching list (can use more than once)
                          ADDSENDER value is sender path string to add to
                              matching list (can use more than once)
                          DELSENDER value is sender path string to remove from
                              matching list (can use more than once)
                Output: for each selected action
                   TAG=0xNNNNNNNN
                        where TAG is unique tag for edited action

     NFALRTCL /LISTPROF <profsel> - List all selected alert profiles
                Output: for each selected alert profile
                   { TAG=0xNNNNNNNN, NAME="profile name", PROFTAG:{ 0xNNNNNNNN, ... }
                     TYPES={ alert_type, ... }, APPS={ "app_id", ... },
                     ATYPES={ atype, ... }, SEVS={ sev, ... },
                     SENDERS={ "sender", ... }, others }
                    where TAG value is unique identifier for profile,
                          NAME value is string identifier for alert profile
                          PROFTAG value is a bracketted list of hex tag IDs for
                              profiles which define the profile (composite) (if used)
                          TYPES value is a bracketted list of standard alert types,
                              with an empty list indicating ANY.  Only present if
                              profiles not used for action
                          APPS value is a bracketted list of application ID strings,
```

with an empty list indicating ANY.  Only present if
                                        profiles not used for action
                               ATYPES value is a bracketted list of hex numbers for
                                        application alert types, with an empty list
                                        indicating ANY.  Only present if profiles not used
                                        for action
                               SEVS value is a bracketted list of alert severities,
                                        with an empty list indicating ANY.  Only present if
                                        profiles not used for action
                               SENDERS value is a bracketted list of sender path strings,
                                        with an empty list indicating ANY.  Only present if
                                        profiles not used for action

NFALRTCL /DELPROFILE <profsel> - Delete all selected alert profiles
          Output: for each selected profile
              TAG=0xNNNNNNNN
                   where TAG is unique tag for deleted action

NFALRTCL /ADDCPROF:"profname" /ADDPROFTAG:NNNNNNNN /ADDPROF:"profname" -
                   Add profile based on profile(s) (composite profile)
NFALRTCL /ADDPROFILE:"profname" /ADDtype:xxxyyy /ADDAPP:"app_id"
                   /ADDAtype:NNNN /ADDSEV:sev /ADDSENDER:"sender" - Add
                   profile based on matching rules
          Input:
              where ADDPROFILE value is name for profile,
                    ADDCPROF value is name for profile,
                    ADDPROFTAG value is hex tag for profile to add to
                         list of profiles for action (can use more than once)
                    ADDPROF value is string name of profile to add to
                         list of profiles for action (can use more than once)
                    ADDTYPE value is standard alert type to add to
                         list of matching alert types (can use more than once)
                    ADDAPP value is application ID string to add to
                         list of matching app IDs (can use more than once)
                    ADDATYPE value is hex application alert type to add
                         to matching list (can use more than once)
                    ADDSEV value is alert severity (0-7) to add to
                         matching list (can use more than once)
                    ADDSENDER value is sender path string to add to
                         matching list (can use more than once)
          Output:
              TAG=0xNNNNNNNN
                   where TAG is unique tag assigned to new action

NFALRTCL /EDITCPROF <profsel> /NEWNAME:"new profile"
              /ADDPROFTAG:NNNNNNNN /DELPROFTAG:NNNNNNNN /ADDPROF:"profname"
              /DELPROF:"profname" - Edit selected profiles (composite profile)
NFALRTCL /EDITPROF <profsel> /NEWNAME:"new profile"
              /ADDtype:xxxyyy /ADDAPP:"app_id" /ADDAtype:NNNN /ADDSEV:sev
              /ADDSENDER:"sender" /DELtype:xxxyyy /DELAPP:"app_id"
              /DELAtype:NNNN /DELSEV:sev /DELSENDER:"sender" - Edit
                                 selected profiles
          Input:
              where NEWNAME value is new profile name,
                    ADDPROFTAG value is hex tag for profile to add to
                         list of profiles for action (can use more than once)
                    DELPROFTAG value is hex tag for profile to remove from
                         list of profiles for action (can use more than once)
                    ADDPROF value is string name of profile to add to
                         list of profiles for action (can use more than once)
                    DELPROF value is string name of profile to remove from
                         list of profiles for action (can use more than once)
                    ADDTYPE value is standard alert type to add to
                         list of matching alert types (can use more than once)
                    DELTYPE value is standard alert type to remove from
                         list of matching alert types (can use more than once)
                    ADDAPP value is application ID string to add to
                         list of matching app IDs (can use more than once)
                    DELAPP value is application ID string to remove from
                         list of matching app IDs (can use more than once)
                    ADDATYPE value is hex application alert type to add
                         to matching list (can use more than once)
                    DELATYPE value is hex application alert type to remove
                         from matching list (can use more than once)
                    ADDSEV value is alert severity (0-7) to add to

```
                                    matching list (can use more than once)
                            DELSEV value is alert severity (0-7) to remove from
                                    matching list (can use more than once)
                            ADDSENDER value is sender path string to add to
                                    matching list (can use more than once)
                            DELSENDER value is sender path string to remove from
                                    matching list (can use more than once)
                Output: for each selected action
                    TAG=0xNNNNNNNN
                            where TAG is unique tag for edited action

    NFALRTCL supports the /N tag on all commands for selecting the target
                alert manager service to interact with.

        Tool-specific return codes:
            rc=200 - Alert log tag not valid
            rc=201 - Alert action tag not valid
            rc=202 - Alert handler not valid
            rc=203 - Alert action match not found
            rc=204 - Invalid alert type
            rc=205 - Invalid alert severity
            rc=206 - Invalid application alert type
            rc=207 - Invalid application ID
            rc=208 - Profile in use by existing actions
            rc=209 - Profiles not supported
            rc=210 - Profile tag/name not valid




    NFALRTCL /? - Command line help

    <actsel> = either /ALL, /TAG, or /HASHAND (with other /HAS* parms optional)
                where ALL selects all configured actions
                        TAG value is the hex identifier of a specific action
                        HASHAND value is handler ID of actions to select,
                        HASPARM# value is value of parm # (0-3) of actions to
                            select (optional),
                        HASPROFTAG value is hex profile tag possessed by
                            actions to select (optional, multiple allowed, with
                            only actions which contain all provided profile
                            tags being selected)
                        HASPROF value is name of profile used by
                            actions to select (optional, multiple allowed, with
                            only actions which contain all provided profile
                            names being selected)
                        HASTYPE value is standard alert type matched by
                            actions to select (optional, multiple allowed, with
                            only actions which match all provided
                            types being selected (actions which match on "any"
                            type are only selected with /HAStype:ANY))
                        HASAPP value is application IDs matched by actions to
                            select (optional, multiple allowed, with
                            only actions which match all provided IDs
                            being selected (actions which match on "any"
                            type are only selected with /HASAPP:ANY))
                        HASATYPE value is hex application
                            alert types matched by actions to select (optional,
                            multiple allowed, with only actions which match
                            all provided types being selected (actions
                            which match on "any" application alert
                            type are only selected with /HASAtype:ANY))
                        HASSEV value is alert severity matched by actions
                            to select (optional, multiple allowed, with only
                            actions which match all provided severities
                            being selected (actions which match on "any"
                            severity are only selected with /HASSEV:ANY))
                        HASSENDER value is sender path matched by actions
                            to select (optional, multiple allowed, with only
                            actions which match all provided sender paths
                            being selected (actions which match on "any"
                            sender are only selected with /HASSENDER:ANY))

    <profsel> = one of the following
```

```
                   /ALL - Select all profiles
                   /TAG:NNNNNNNN - Select profile with given tag
                   /NAME:"profname" - Select profile(s) with given name


<logsel> =  one of the following
                   /ALL - Select all alerts in log
                   /TAG:NNNNNNNN - Select alert with given tag


<handsel> = one of the following
                   /ALL - Select all handlers
                   /NAME:"handname" - Select handler with given ID string


NFALRTCL /LISTLOG <logsel> - List all selected alerts in alert log
        Output: for each selected alert in log
           { TAG=0xNNNNNNNN, TEXT="alert text", TIME=alert_time,
             DATE=alert_date, SEV=severity, TYPE=alrt_type,
             APP="app_id", ATYPE=0xNNNN, SYSNAME="system name",
             SENDER="sender netpath", P1="parm1", others }
           where TAG value is identifier for log entry,
                  TEXT value is string for alert text,
                  TIME value is time attribute of alert (hh:mm:ss),
                  DATE value is date attribute of alert (dd-mm-yyyy),
                  SEV value is severity of alert (0-7),
                  TYPE value is standard alert type: format is two
                       3-character sequences (as used in GENALERT): xxxyyy
                       where xxx is alert type:
                          UNK  - Unknown
                          SYS  - System
                          DSK  - Disk or DASD
                          NET  - Network
                          OS_  - Operating System
                          APP  - Application
                          DEV  - Device
                          SEC  - Security
                       and yyy is alert class:
                          UNK  - Unknown
                          FLT  - Fault or Failure
                          ERR  - Error
                          WRN  - Warning
                          INF  - Information
                  APP value is string for application ID of alert,
                  ATYPE value is hex number of application alert type,
                  SYSNAME value is string for system name,
                  SENDER value is string for alert sender's path,
                  P# value is string for alert parameter # (#=1-9)


NFALRTCL /DELLOG <logsel> - Delete all selected alerts in alert log


NFALRTCL /LISTHAND <handsel> - List all selected alert handlers
        Output: for each selected alert handler
           { HANDLER="handler_id", NAME="handler name",
             PARMLAB0="parm label", PARMLAB1="parm label", others }
           where HANDLER value is string identifier for alert handler:
                    "WEBACT/0" - Send alert as TCP/IP Web mail
                    "SETERRORCONDITION" - Set error condition for
                                          sending system
                    "CLEARERRORCONDITION" - Clear error condition for
                                          sending system
                    "DMIACT/0" - Semd DMI Event through DMI Service Layer
                    "ALERTLOG" - Add alert to log file
                    "ALERTFORWARD" - Forward alert through network
                    "ALERTPOPUP" - Notify user with pop-up
                    "RUNCOMMAND" - Execute command
                    "RUNMINCOMMAND" - Execute minimized command
                    "PAGERACT/0" - Activate numeric pager
                    "PAGERACT/1" - Send to alphanumeric pager
                    "TCPIPACT/0" - Send SNMP alert
                    "TCPIPACT/1" - Send TCP/IP mail
                    "TCPIPACT/2" - Send mapped SNMP alert
                    others: name is uppercase DLL name, followed by '/',
                        followed by entry point number (AlertHandler#).
                  NAME value is string for alert label
                  PARMLAB# value is string label for alert parameter (0-3)


NFALRTCL /LISTACT <actsel> - List all selected alert actions
```

```
Output: for each selected alert action
  { TAG=0xNNNNNNNN, NAME="name", HANDLER="handler_id",
    PARM0="parm value", PARM1="parm value",
    PROFTAG={ 0xNNNNNNNN, ... }, PROF={"profname", ... },
    TYPES={ alert_type, ... }, APPS={ "app_id", ... },
    ATYPES={ atype, ... }, SEVS={ sev, ... },
    SENDERS={ "sender", ... }, others }
  where TAG value is unique identifier for action,
        NAME value is string label (on profile actions),
        HANDLER value is string identifier for alert handler
        PARM# value is string value for parameter # (0-3)
        PROFTAG value is a bracketted list of hex tag IDs for
           profiles used to match with action (if used)
        PROF value is a bracketted list of strings for
           profile names used to match with action (if used)
        TYPES value is a bracketted list of standard alert types,
           with an empty list indicating ANY.  Only present if
           profiles not used for action
        APPS value is a bracketted list of application ID strings,
           with an empty list indicating ANY.  Only present if
           profiles not used for action
        ATYPES value is a bracketted list of hex numbers for
           application alert types, with an empty list
           indicating ANY.  Only present if profiles not used
           for action
        SEVS value is a bracketted list of alert severities,
           with an empty list indicating ANY.  Only present if
           profiles not used for action
        SENDERS value is a bracketted list of sender path strings,
           with an empty list indicating ANY.  Only present if
           profiles not used for action

NFALRTCL /DELACT <actsel> - Delete all selected alert actions
        Output: for each selected action
            TAG=0xNNNNNNNN
                where TAG is unique tag for deleted action

NFALRTCL /ADDPACT:"handler_id" /NEWNAME:"name" /PARM0:"parm val"
            /PARM1:"parm val" /ADDPROFTAG:NNNNNNNN /ADDPROF:"profname" -
                Add alert action based on profile(s)
NFALRTCL /ADDACT:"handler_id" /PARM0:"parm val" /PARM1:"parm val"
            /ADDtype:xxxyyy /ADDAPP:"app_id" /ADDAtype:NNNN /ADDSEV:sev
            /ADDSENDER:"sender" - Add alert action based on matching rules
        Input:
            where ADDACT value is handler ID for action,
                  ADDPROFACT value is handler ID for action,
                  PARM# value is value for parameter # of action (0-3)
                  NEWNAME value is string label for profile-based action
                  ADDPROFTAG value is hex tag for profile to add to
                     list of profiles for action (can use more than once)
                  ADDPROF value is string name of profile to add to
                     list of profiles for action (can use more than once)
                  ADDTYPE value is standard alert type to add to
                     list of matching alert types (can use more than once)
                  ADDAPP value is application ID string to add to
                     list of matching app IDs (can use more than once)
                  ADDATYPE value is hex application alert type to add
                     to matching list (can use more than once)
                  ADDSEV value is alert severity (0-7) to add to
                     matching list (can use more than once)
                  ADDSENDER value is sender path string to add to
                     matching list (can use more than once)
        Output:
            TAG=0xNNNNNNNN
                where TAG is unique tag assigned to new action

NFALRTCL /EDITPACT <actsel> /NEWHAND:"handler_id"
            /PARM0:"parm val" /PARM1:"parm val" /ADDPROFTAG:NNNNNNNN
            /DELPROFTAG:NNNNNNNN /ADDPROF:"profname" /DEFPROF:"profname"
                - Edit alert action (profile) with given tag
NFALRTCL /EDITACT <actsel> /NEWHAND:"handler_id"
            /PARM0:"parm val" /PARM1:"parm val" /ADDtype:xxxyyy
            /ADDAPP:"app_id" /ADDAtype:NNNN /ADDSEV:sev /ADDSENDER:"sender"
            /DELtype:xxxyyy /DELAPP:"app_id" /DELAtype:NNNN
            /DELSEV:sev /DELSENDER:"sender" - Edit alert action with given tag
```

```
Input:
    where NEWHAND value is new handler ID for action,
          PARM# value is new value for parameter # of action (0-3)
          ADDPROFTAG value is hex tag for profile to add to
              list of profiles for action (can use more than once)
          DELPROFTAG value is hex tag for profile to remove from
              list of profiles for action (can use more than once)
          ADDPROF value is string name of profile to add to
              list of profiles for action (can use more than once)
          DELPROF value is string name of profile to remove from
              list of profiles for action (can use more than once)
          ADDTYPE value is standard alert type to add to
              list of matching alert types (can use more than once)
          DELTYPE value is standard alert type to remove from
              list of matching alert types (can use more than once)
          ADDAPP value is application ID string to add to
              list of matching app IDs (can use more than once)
          DELAPP value is application ID string to remove from
              list of matching app IDs (can use more than once)
          ADDATYPE value is hex application alert type to add
              to matching list (can use more than once)
          DELATYPE value is hex application alert type to remove
              from matching list (can use more than once)
          ADDSEV value is alert severity (0-7) to add to
              matching list (can use more than once)
          DELSEV value is alert severity (0-7) to remove from
              matching list (can use more than once)
          ADDSENDER value is sender path string to add to
              matching list (can use more than once)
          DELSENDER value is sender path string to remove from
              matching list (can use more than once)
Output: for each selected action
    TAG=0xNNNNNNNN
        where TAG is unique tag for edited action


NFALRTCL /LISTPROF <profsel> - List all selected alert profiles
        Output: for each selected alert profile
            { TAG=0xNNNNNNNN, NAME="profile name", PROFTAG:{ 0xNNNNNNNN, ... }
              TYPES={ alert_type, ... }, APPS={ "app_id", ... },
              ATYPES={ atype, ... }, SEVS={ sev, ... },
              SENDERS={ "sender", ... }, others }
            where TAG value is unique identifier for profile,
                  NAME value is string identifier for alert profile
                  PROFTAG value is a bracketted list of hex tag IDs for
                      profiles which define the profile (composite) (if used)
                  TYPES value is a bracketted list of standard alert types,
                      with an empty list indicating ANY.  Only present if
                      profiles not used for action
                  APPS value is a bracketted list of application ID strings,
                      with an empty list indicating ANY.  Only present if
                      profiles not used for action
                  ATYPES value is a bracketted list of hex numbers for
                      application alert types, with an empty list
                      indicating ANY.  Only present if profiles not used
                      for action
                  SEVS value is a bracketted list of alert severities,
                      with an empty list indicating ANY.  Only present if
                      profiles not used for action
                  SENDERS value is a bracketted list of sender path strings,
                      with an empty list indicating ANY.  Only present if
                      profiles not used for action

NFALRTCL /DELPROFILE <profsel> - Delete all selected alert profiles
        Output: for each selected profile
            TAG=0xNNNNNNNN
                where TAG is unique tag for deleted action


NFALRTCL /ADDCPROF:"profname" /ADDPROFTAG:NNNNNNNN /ADDPROF:"profname" -
              Add profile based on profile(s) (composite profile)
NFALRTCL /ADDPROFILE:"profname" /ADDtype:xxxyyy /ADDAPP:"app_id"
              /ADDAtype:NNNN /ADDSEV:sev /ADDSENDER:"sender" - Add
              profile based on matching rules
        Input:
            where ADDPROFILE value is name for profile,
                  ADDCPROF value is name for profile,
```

```
                           ADDPROFTAG value is hex tag for profile to add to
                               list of profiles for action (can use more than once)
                           ADDPROF value is string name of profile to add to
                               list of profiles for action (can use more than once)
                           ADDTYPE value is standard alert type to add to
                               list of matching alert types (can use more than once)
                           ADDAPP value is application ID string to add to
                               list of matching app IDs (can use more than once)
                           ADDATYPE value is hex application alert type to add
                               to matching list (can use more than once)
                           ADDSEV value is alert severity (0-7) to add to
                               matching list (can use more than once)
                           ADDSENDER value is sender path string to add to
                               matching list (can use more than once)
                   Output:
                       TAG=0xNNNNNNNN
                           where TAG is unique tag assigned to new action

        NFALRTCL /EDITCPROF <profsel> /NEWNAME:"new profile"
                   /ADDPROFTAG:NNNNNNNN /DELPROFTAG:NNNNNNNN /ADDPROF:"profname"
                   /DELPROF:"profname" - Edit selected profiles (composite profile)
        NFALRTCL /EDITPROF <profsel> /NEWNAME:"new profile"
                   /ADDtype:xxxyyy /ADDAPP:"app_id" /ADDAtype:NNNN /ADDSEV:sev
                   /ADDSENDER:"sender" /DELtype:xxxyyy /DELAPP:"app_id"
                   /DELAtype:NNNN /DELSEV:sev /DELSENDER:"sender" - Edit
                                   selected profiles
                   Input:
                        where NEWNAME value is new profile name,
                           ADDPROFTAG value is hex tag for profile to add to
                               list of profiles for action (can use more than once)
                           DELPROFTAG value is hex tag for profile to remove from
                               list of profiles for action (can use more than once)
                           ADDPROF value is string name of profile to add to
                               list of profiles for action (can use more than once)
                           DELPROF value is string name of profile to remove from
                               list of profiles for action (can use more than once)
                           ADDTYPE value is standard alert type to add to
                               list of matching alert types (can use more than once)
                           DELTYPE value is standard alert type to remove from
                               list of matching alert types (can use more than once)
                           ADDAPP value is application ID string to add to
                               list of matching app IDs (can use more than once)
                           DELAPP value is application ID string to remove from
                               list of matching app IDs (can use more than once)
                           ADDATYPE value is hex application alert type to add
                               to matching list (can use more than once)
                           DELATYPE value is hex application alert type to remove
                               from matching list (can use more than once)
                           ADDSEV value is alert severity (0-7) to add to
                               matching list (can use more than once)
                           DELSEV value is alert severity (0-7) to remove from
                               matching list (can use more than once)
                           ADDSENDER value is sender path string to add to
                               matching list (can use more than once)
                           DELSENDER value is sender path string to remove from
                               matching list (can use more than once)
                    Output: for each selected action
                       TAG=0xNNNNNNNN
                           where TAG is unique tag for edited action

        NFALRTCL supports the /N tag on all commands for selecting the target
                   alert manager service to interact with.

        Tool-specific return codes:
             rc=200 - Alert log tag not valid
             rc=201 - Alert action tag not valid
             rc=202 - Alert handler not valid
             rc=203 - Alert action match not found
             rc=204 - Invalid alert type
             rc=205 - Invalid alert severity
             rc=206 - Invalid application alert type
             rc=207 - Invalid application ID
             rc=208 - Profile in use by existing actions
             rc=209 - Profiles not supported
             rc=210 - Profile tag/name not valid
```

```
NFSMONCL /? - Command line help


<mon_attr_sel> = one of the following
                /ALL - all monitors and attribute groups
                /MONNAME:"name" - monitors/attribute groups with given name
                /MONID:NNNNNNNN - monitors/attribute groups with given
                        monitor ID

<monthr_sel> = one of the following
                /ALL - all monitor thresholds
                /MONNAME:"name" - Thresholds of monitors with given name
                /MONID:NNNNNNNN - Thresholds of monitor with given monitor ID
                 optional additional selector
                /THRNAME:"thrname" = Limits selection to thresholds with
                        given name

NFSMONCL /GETMON <mon_attr_sel> - List monitor and attribute group
        information
         Output: for each selected monitor
            { MONITOR_ID=OxNNNNNNNN, NAME="name", SAMPLE=time_msec,
             VALUE=cur_val, UNITS_LBL="units string", RECORDING=enab,
             MINVAL=min_val, MAXVAL=max_val, others }
            where MONITOR_ID value is hex number for monitor ID,
                   NAME value is string attribute name,
                   SAMPLE value is attribute sample rate, in
                       milliseconds,
                   VALUE value is decimal value of attribute currently,
                   UNITS_LBL value is string label for data type,
                   RECORDING value is keyword for data recording state,
                       either ENABLED or DISABLED
                   MINVAL value is decimal value of minimum for value range
                   MAXVAL value is decimal value of maximum for value range
         Output: for each selected attribute group
            { ATTRIB_ID=OxNNNNNNNN, NAME="name", SAMPLE=time_msec,
             RECORDING=enab, { attrib_info, attrib_info, ... }, others }
            where ATTRIB_ID value is hex number for attribute ID,
                   NAME value is string attribute group name,
                   SAMPLE value is attribute group sample rate, in
                       milliseconds,
                   RECORDING value is keyword for data recording state,
                       either ENABLED or DISABLED
                   attrib_info is bracketted data block formatted as follows:
                       { ATTRIB_SUBID=OxNNNNNNNN, NAME="name",
                         VAL_ENUM={ "val_desc", ... }, VALUE=val }
                       where ATTRIB_SUBID value is hex ID for attribute in
                              group,
                             NAME value is string attribute name,
                             VAL_ENUM value is bracketted list of strings
                              for enumeration values (O-based index),
                             VALUE value is either index in VAL_ENUM list of
                              current value, or a string for the current
                              value (if no VAL_ENUM list)

NFSMONCL /SETMONREC:enab <mon_attr_sel> - Set recording on selected monitors
        Input:
                where SETMONREC parm is recording state (ENABLED or DISABLED)
        Output: for each modified monitor
                MONITOR_ID=OxNNNNNNNN
                 for each modified attribute group
                ATTRIB_ID=OxNNNNNNNN

NFSMONCL /GETMONTHR <monthr_sel> - List information for selected monitor
                                thresholds
        Output: for each selected threshold
                { MONITOR_ID=OxNNNNNNNN, THRESHNAME="name", DUR=secs,
                 RPT=secs, HIERRVAL=val, HIERRSEV=sev, HIERRNTFY=enab,
                 HIWRNVAL=val, HIWRNSEV=sev, HIWRNNTFY=enab,
                 LOWRNVAL=val, LOWRNSEV=sev, LOWRNNTFY=enab,
                 LOERRVAL=val, LOERRSEV=sev, LOERRNTFY=enab,
                 RTNSEV=sev, RTNNTFY=enab, LOCALNTFY=enab, others }
```

```
                    where MONITOR_ID value is hex monitor ID,
                          THRESHNAME value is string for threshold name,
                          DUR value is number of seconds for
                                  duration of threshold values,
                          RPT value is number of seconds for
                                  resend duration of threshold values,
                          HIERRVAL value is decimal value of high-error
                                  threshold (if defined), or NONE,
                          HIERRSEV value is severity of threshold alert (0-7)
                                  for high-error threshold,
                          HIERRNTFY value is notify state of threshold with
                                  respect to the managing system (ENABLED or
                                  DISABLED),
                          HIWRNVAL value is decimal value of high-warning
                                  threshold (if defined), or NONE,
                          HIWRNSEV value is severity of threshold alert (0-7)
                                  for high-warning threshold,
                          HIWRNNTFY value is notify state of threshold with
                                  respect to the managing system (ENABLED or
                                  DISABLED),
                          LOWRNVAL value is decimal value of low-warning
                                  threshold (if defined), or NONE,
                          LOWRNSEV value is severity of threshold alert (0-7)
                                  for low-warning threshold,
                          LOWRNNTFY value is notify state of threshold with
                                  respect to the managing system (ENABLED or
                                  DISABLED),
                          LOERRVAL value is decimal value of low-error
                                  threshold (if defined), or NONE,
                          LOERRSEV value is severity of threshold alert (0-7)
                                  for low-error threshold,
                          LOERRNTFY value is notify state of threshold with
                                  respect to the managing system (ENABLED or
                                  DISABLED),
                          RTNSEV value is severity of return-to-normal alert
                                  (0-7),
                          RTNNTFY value is notify state of return-to-normal
                                  with respect to the managing system (ENABLED
                                  or DISABLED)
                          LOCALNTFY value is notify state with respect to the
                                  local system (ENABLED or DISABLED)

   NFSMONCL /DELMONTHR <monthr_sel> - Delete selected monitor thresholds
          Output: for each selected threshold
                  { MONITOR_ID=0xNNNNNNNN, THRESHNAME="name" }
                  where MONITOR_ID value is monitor ID,
                        THRESHNAME value is string of threshold name


   NFSMONCL /ADDMONTHR:"name" /MONID:NNNNNNNN /DUR:secs /RPT:secs
          /HIERRVAL:val /HIERRSEV:sev /HIERRNTFY:enab /HIWRNVAL:val
          /HIWRNSEV:sev /HIWRNNTFY:enab /LOWRNVAL:val /LOWRNSEV:sev
          /LOWRNNTFY:enab /LOERRVAL:val /LOERRSEV:sev /LOERRNTFY:enab
          /RTNSEV:sev /RTNNTFY:enab /LOCALNTFY:enab - Add a new monitor
                  threshold
           Input: where ADDMONTHR value is string threshold name (required),
                        MONID value is hex attribute ID (required),
                        DUR value is number of seconds for duration of
                                threshold values (default is 0),
                        RPT value is number of seconds for resend duration
                                of threshold values (default is 0),
                        HIERRVAL value is decimal value of high-error
                                threshold (if defined), or NONE (default),
                        HIERRSEV value is severity of threshold alert (0-7)
                                for high-error threshold (default is 2),
                        HIERRNTFY value is notify state of threshold with
                                respect to the managing system (ENABLED or
                                DISABLED) (default is DISABLED),
                        HIWRNVAL value is decimal value of high-warning
                                threshold (if defined), or NONE (default),
                        HIWRNSEV value is severity of threshold alert (0-7)
                                for high-warning threshold (default is 4),
                        HIWRNNTFY value is notify state of threshold with
                                respect to the managing system (ENABLED or
                                DISABLED) (default is DISABLED),
                        LOWRNVAL value is decimal value of low-warning
```

```
                                threshold (if defined), or NONE (default),
                        LOWRNSEV value is severity of threshold alert (0-7)
                                for low-warning threshold (default is 4),
                        LOWRNNTFY value is notify state of threshold with
                                respect to the managing system (ENABLED or
                                DISABLED) (default is DISABLED),
                        LOERRVAL value is decimal value of low-error
                                threshold (if defined), or NONE (default),
                        LOERRSEV value is severity of threshold alert (0-7)
                                for low-error threshold (default is 2),
                        LOERRNTFY value is notify state of threshold with
                                respect to the managing system (ENABLED or
                                DISABLED) (default is DISABLED),
                        RTNSEV value is severity of return-to-normal alert
                                (0-7) (default is 6),
                        RTNNTFY value is notify state of return-to-normal
                                with respect to the managing system (ENABLED
                                or DISABLED) (default is DISABLED)
                        LOCALNTFY value is notify state for local notifies
                                (ENABLED or DISABLED) (default is DISABLED)
        Output: for new threshold
                { MONITOR_ID=0xNNNNNNNNN, THRESHNAME="name" }
                where MONITOR_ID value is monitor ID,
                        THRESHNAME value is string of threshold name


NFSMONCL /EDITMONTHR <monthr_sel> /DUR:secs /RPT:secs
        /HIERRVAL:val /HIERRSEV:sev /HIERRNTFY:enab /HIWRNVAL:val
        /HIWRNSEV:sev /HIWRNNTFY:enab /LOWRNVAL:val /LOWRNSEV:sev
        /LOWRNNTFY:enab /LOERRVAL:val /LOERRSEV:sev /LOERRNTFY:enab
        /RTNSEV:sev /RTNNTFY:enab - Modify selected monitor thresholds
         Input: where DUR value is number of seconds for duration of
                                threshold values (default is 5),
                        RPT value is number of seconds for resend duration
                                of threshold values (default is 0),
                        HIERRVAL value is decimal value of high-error
                                threshold (if defined), or NONE (default),
                        HIERRSEV value is severity of threshold alert (0-7)
                                for high-error threshold (default is 2),
                        HIERRNTFY value is notify state of threshold with
                                respect to the managing system (ENABLED or
                                DISABLED) (default is DISABLED),
                        HIWRNVAL value is decimal value of high-warning
                                threshold (if defined), or NONE (default),
                        HIWRNSEV value is severity of threshold alert (0-7)
                                for high-warning threshold (default is 4),
                        HIWRNNTFY value is notify state of threshold with
                                respect to the managing system (ENABLED or
                                DISABLED) (default is DISABLED),
                        LOWRNVAL value is decimal value of low-warning
                                threshold (if defined), or NONE (default),
                        LOWRNSEV value is severity of threshold alert (0-7)
                                for low-warning threshold (default is 4),
                        LOWRNNTFY value is notify state of threshold with
                                respect to the managing system (ENABLED or
                                DISABLED) (default is DISABLED),
                        LOERRVAL value is decimal value of low-error
                                threshold (if defined), or NONE (default),
                        LOERRSEV value is severity of threshold alert (0-7)
                                for low-error threshold (default is 2),
                        LOERRNTFY value is notify state of threshold with
                                respect to the managing system (ENABLED or
                                DISABLED) (default is DISABLED),
                        RTNSEV value is severity of return-to-normal alert
                                (0-7) (default is 6),
                        RTNNTFY value is notify state of return-to-normal
                                with respect to the managing system (ENABLED
                                or DISABLED) (default is DISABLED)
                        LOCALNTFY value is notify state for local notifies
                                (ENABLED or DISABLED) (default is DISABLED)

    Tool-specific return codes:
        rc=200 - Unknown monitor or attribute group ID
        rc=201 - Unknown monitor or attribute group name
        rc=202 - Unknown monitor or attribute group threshold name
```

```
NFCRTFCL /? - Command line help

NFCRTFCL /LIST - List information about the monitored files.
         Output: for each file being monitored
             { FILE="fully_qualified_path", SEV=x, TYPE=file_type }
              where FILE value is string for fully qualified path to file
                    SEV value is severity of alert that will be generated (0-7)
                    TYPE value is type of file being monitored, either
                         SYSTEM_FILE OR STANDARD_FILE

NFCRTFCL /DELSYS:filename_without_path - Delete a "system file" from the
                 list of monitored files.
         Input:
             where DELSYS value is filename only (path should not be included)

NFCRTFCL /DELSTD:fully_qualified_path - Delete a "standard file" from the
                 list of monitored files.
         Input:
             where DELSTD value is fully qualified path to file

NFCRTFCL /SETSYS:filename_without_path /SEV:x - Add a new "system file"
                 to be monitored or change the alert severity of a "system file"
                 already being monitored.
         Input:
             where SETSYS value is filename only (path should not be included)
                   SEV value is severity of alert that will be generated (0-7)

NFCRTFCL /SETSTD:fully_qualified_path /SEV:x  - Add a new "standard file"
                 to be monitored or change the alert severity of a "standard
                 file" already being monitored.
         Input:
             where SETSTD value is fully qualified path to file
                   SEV value is severity of alert that will be generated (0-7)

NFCRTFCL supports the /N tag on all commands for selecting the target
              critical file monitor service to interact with.

   Tool-specific return codes:
         rc=200 - fully qualified path missing
            201 - fully qualified path invalid
            202 - filename (without path) missing
            203 - filename (without path) invalid
            204 - severity missing
            205 - severity invalid
            206 - severity out of range




NFPROCCL /? - Command line help

<procsel> = one of the following
               /ALL - Select all process alerts
               /PROCNAME:"name" - Select process alerts with name
               /TAG:NNNNNNNN - Select process alerts with given tag

NFPROCCL /GETPROC - List current process information
         Output: for each active process
             { PROCESS_ID=0xNNNNNNNN, EXENAME="name", CMDLINE="cmd line",
               THREADS=numthrds, FILES=numfiles, PARENT_ID=0xNNNNNNNN,
               SESSION_ID=0xNNNNNNNN, USER_ID="user_id",
               PRIORITY=prio, START_TIME=timedate, EXEVER="version",
               EXEDATE=date, EXEDESC="description", other }
              where PROCESS_ID value is hex number for process ID,
                    EXENAME value is string process name,
                    CMDLINE value is string command line (OS-specific),
                    THREADS value is integer thread count (OS-specific),
                    FILES value is integer open file count (OS-specific),
                    PROCESS_ID value is hex number for parent process ID,
                    SESSION_ID value is hex number for session ID
                        (OS-specific),
                    USER_ID value is string user ID (OS-specific),
```

```
                        PRIORITY value is keyword for process priority:
                              IDLE, LOW, BACKGROUND, NORMAL, FOREGROUND, HIGH,
                              SERVER, TIMECRIT, REALTIME (OS-specific),
                        START_TIME value is timestamp for process start time
                              (OS-specific),
                        EXEVER value is executable version string (OS-specific),
                        EXEDATE value is timestamp for executable (OS-specific),
                        EXEDESC value is string executable description
                              (OS-specific)


NFPROCCL /RUNCMD:"command string" - Run a requested command


NFPROCCL /LISTMON <procsel> - List process monitors
           Output: for each selected process monitor
               { TAG=0xNNNNNNNN, PROCNAME="process name", SEV=severity,
                 ONSTART=enable, ONSTOP=enable, ONNORUN=time,
                 NOTIFY={"path", ...}, others }
                where TAG value is hex number for unique identifier,
                      PROCNAME value is string process name,
                      SEV value is integer alert severity (0-7),
                      ONSTART value is ENABLED or DISABLED,
                      ONSTOP value is ENABLED or DISABLED,
                      ONNORUN value is integer time, in minutes, or DISABLED
                      NOTIFY value is bracketted list of strings for
                            notify paths : LOCAL is local system


NFPROCCL /ADDMON:"process name" /SEV:severity /ONSTART:enabled
                 /ONSTOP:enabled /ONNORUN:time /ADDNOTIFY:path - Add new
                 process alert
           Input:
               where ADDMON value is name for process (required),
                     SEV value is integer for severity (required),
                     ONSTART value is ENABLED or DISABLED (optional,
                           default is DISABLED),
                     ONSTOP value is ENABLED or DISABLED (optional,
                           default is DISABLED),
                     ONNORUN value is DISABLED or integer time in minutes
                           (optional, default is DISABLED),
                     ADDNOTIFY value is network path for alert notification
                           (optional, multiple allowed) : LOCAL is local system,
                           HERE is alias for manager's system (local to NFPROCCL)
           Output:
               TAG=0xNNNNNNNN
                   where TAG is unique tag for new process monitor


NFPROCCL /DELMON <procsel> - Delete all selected process monitors
           Output: for each selected process monitor
               TAG=0xNNNNNNNN
                   where TAG is unique tag for deleted process monitor


NFPROCCL /EDITMON <procsel> /NEWNAME:"name" /SEV:severity
                 /ONSTART:enabled /ONSTOP:enabled /ONNORUN:time
                 /ADDNOTIFY:path /DELNOTIFY:path - Edit
                 selected process monitors
           Input:
               where EDITMON value is name for process (required),
                     NEWNAME value is string for new process name,
                     SEV value is integer for severity,
                     ONSTART value is ENABLED or DISABLED),
                     ONSTOP value is ENABLED or DISABLED,
                     ONNORUN value is DISABLED or integer time in minutes
                     ADDNOTIFY value is network path for alert notification
                           (optional, multiple allowed) : LOCAL is local system,
                           HERE is alias for manager's system (local to NFPROCCL)
                     DELNOTIFY value is network path for alert notification
                           (optional, multiple allowed) : LOCAL is local system,
                           HERE is alias for manager's system (local to NFPROCCL)
           Output: for each selected process monitor
               TAG=0xNNNNNNNN
                   where TAG is unique tag for edited process monitor


    Tool-specific return codes:
         rc=200 - Process monitor tag/name not valid
         rc=201 - Run command failed
```

```
NFSYSICL /? - Command line help

NFSYSICL /HIST:filename - Generate a history file to 'filename'
NFSYSICL /HISTOUT - Generate a history file to a generated filename
        (based on the /S parameter value, or a random value)
        Output:
            HSTFILE="filename"
          where HSTFILE value is name of file containing output

NFSYSICL /RPT:filename - Generate a textual report to 'filename'
NFSYSICL /RPTOUT - Generate a textual report to a generated filename
        (based on the /S parameter value, or a random value)
        Output:
            RPTFILE="filename"
          where RPTFILE value is name of file containing report

NFSYSICL /EXPORT:dbi_name /DBNAME:"db_name" /NOPROF /NOSYSLEV
        /MGR:"mgr_name" /GRP:"grp_name" /S:"sys_name" -
        Export data to database 'db_name' using database
        driver 'dbi_name'
        Input:
            where EXPORT value is the name of the database interface
                module (DBI) used for exporting to the selected
                database (i.e DB2OS2CI, DB2WINFI, etc) (required)
                DBNAME value is string representing name of the
                target database (format depends upon specific
                DBI driver being used) (required)
                NOPROF indicates that system profile data should
                be excluded from export (optional, default is to
                include the system profile data)
                NOSYSLEV indicates that software information from
                OS/2 SYSLEVEL files should be excluded from
                export (optional, default is to include SYSLEVEL data)
                MGR value is string for identifying requesting manager
                (optional, default is local system name)
                GRP value is string for identifying system's group
                (optional, default is none)
                S value is string for system name (required)

    Tool-specific return codes:
        rc=200 - Error writing to history file
        rc=201 - Error writing to report file
        rc=202 - Error during database export
        rc=203 - Unable to execute sysinfo task (SINFG30.EXE)




NFREPLCL /? - Command line help

filename_from_SCF_dir = name of file in SCF directory under the
            NetFinity executable directory (\NETFIN or \WNETFIN)

NFREPLCL /LISTFILES  - Show all configuration files
        Output, for each file:
            { FILE="filename_from_SCF_dir",
              DESC="description of file"
            }

NFREPLCL /LISTREMOTES  - Show available remote systems
        Output, for each remote system known by Remote System Manager:
            { NAME="system name",
              PATH="system netpath, e.g., NETBIOS::JUPITER"
            }

NFREPLCL /LISTSERVICES  - Show services available to save
        Output, for each available service:
            { NAME="service name, e.g., Alert Manager",
              SERVICE="service, e.g., AlertMgr"
            }
```

```
NFREPLCL /SHOW:filename_from_SCF_dir
        - Summarize the contents of a configuration file
      Output:
        { SERVICE="name of service in file",
          DESC="description of file",
          SYSTEM="name of system whose configuration was saved into file",
          DATE="time and date of saving of configuration",
          SUBSETS={
            { NAME="name of subset of configuration records"
              ID="NNN"
              RECORDS={
                { REC="NNN.NNN",
                  DESC="description of record"
                },
                ...
              }
            },
            ...
          }
        }
        where TAG="NNN" is the 0-index number of the subset,
        and REC="NNN.NNN" is the 0-index number of the subset followed
        by a period and the 0-index number of the record in the subset

NFREPLCL /PRUNE:filename_from_SCF_dir
        /REC:NNN.NNN /REC:NNN.NNN ...
        - Prune the given records from the file
        where /REC:NNN.NNN is the 0-index number of the subset followed
        by a period and the 0-index number of the record in the subset
      Output:  None

NFREPLCL /DELETE:filename_from_SCF_dir
        - Delete file
      Output:  None

NFREPLCL /SAVE:filename_to_go_in_SCF_dir
        /SERVICE:service_to_save_configuration_from
        /REMOTE:path_to_remote_system
        /SYSNAME:nondefault_name_of_remote_system
        /DESC:description_to_give_to_file
        - Save the service from the system to the file
        where /SERVICE refers to the NetFinity name of the service,
        e.g., "AlertMgr"
      Output:  None

NFREPLCL /RESTORE:filename_from_SCF_dir
        /REMOTE:path_to_remote_system
        /CLEAN  - Delete old configuration before restoring
        - Restore the service from the file to the system
      Output:  None

   Tool-specific return codes:
        rc=200 - Means of dealing with service configuration not found.
                 Generally, this would be a mis-specified service name
                 or a missing SCF*.DLL.


All NetFinity CLI tools conform to the following specifications:

 1. Tools are text-mode executables which direct their outputs to stdout and stderr, and
    accept all inputs from command-line parameters and (optionally) a textual response file.
    On Win95 and WinNT, this means that the tools are Win32 Console apps.  On OS/2, this
    means that the tools are VIO apps. Win16 apps are not supported.

 2. CLI tools return errorlevels conforming to the following definitions, as appropriate:

    rc=0  (NFCMDLINE_SUCCESS) Operation requested was completed successfully

    rc=1  (NFCMDLINE_SYSTEM_UNAVAILABLE) Target system is unavailable or unreachable

    rc=2  (NFCMDLINE_SERVICE_DOES_NOT_EXIST) Needed service(s) on target system is not
             available or defined

    rc=3  (NFCMDLINE_NOT_AUTHORIZED) Authorization for needed service(s) on target
```

```
                          system is not available

    rc=4  (NFCMDLINE_TIMEOUT) Communication with target system failed due to time-out

    rc=5  (NFCMDLINE_COMMUNICATION_ERROR) Communication with target system failed
              (non-timeout)

    rc=6  (NFCMDLINE_SERVICE_BUSY) Needed service(s) on target system are available, but
              currently not available for use

    rc=7  (NFCMDLINE_SVC_NODE_CREATE_FAILED) Attempt to create ServiceNode failed
              (probably missing DLLs)

    rc=8  (NFCMDLINE_SVC_START_FAILED) Request to start required service(s) on target
              system failed

    rc=9  (NFCMDLINE_SVC_STOP_FAILED) Request to stop required service(s) on target
              system failed

    rc=10 (NFCMDLINE_BAD_COMMAND_LINE) Command-line syntax error or invalid input

    rc=11 (NFCMDLINE_FUNCTION_NOT_SUPPORTED) Requested operation is not supported on
              the service(s) on the target system

    rc=12 (NFCMDLINE_FILE_NOT_FOUND) Requested operation failed due to not finding
              needed local file

    rc=13 (NFCMDLINE_FILE_OUTPUT_ERROR) Requested operation failed due I/O error to
              local file
```

3. Data output for CLI tools is intended for programmatic processing by other tools.  To
   support this, informational outputs from the CLI tools are as follows:

   a. All information outputs (as opposed to error messages, help messages, logo messages,
      copyright messages, and other human-oriented outputs) are directed to 'stdout'.  All
      other outputs are directed to 'stderr'.

   b. Information outputs consist of keywords or keyword/value pairs, i.e. MANAGER,
      ATTRIBNAME=VALUE, TOKEN="string"

   c. Grouped outputs are formatted by listing keyword and keyword/values pairs, separated
      by commas, spaces, or new lines or consecutive lines, with a blank-line marking the
      end of each record.  Each group of data is enclosed in '{' and '}' brackets.  Groups of
      data can be nested, for example

```
              { TAG=1234, NAME='xxx', MANAGER, MAC=4000123677 }
              { TAG=1236, NAME='yyy', MAC=4000123678 }
```

      or

```
              { TAG=1234
                NAME='xxx'
                MANAGER
                MAC=4000123677 }
              { TAG=1236
                NAME='yyy'
                MAC=4000123678 }
```

      or

```
              { TAG=5555, GROUPNAME='aaa'
                SYSTEMS={ { TAG=1234, NAME='xxx', MANAGER, MAC=4000123677 }
                         { TAG=1236, NAME='yyy', MAC=4000123678 } } }
```

   d. Values in keyword/value pairs are formatted as follows:

      keyword=0xNNNN for hexadecimal values (capital letters for A-F)

      keyword=NNNN for decimal values (a leading '-' or '+' is allowed)

      keyword=AAAAAA for keyword-type values (i.e. ON, OFF, ENABLED)

      keyword="string" for string-like values (use two double-quotes for double-quote
      characters within the string. i.e. "")

keyword=NNN.NNN for floating-point decimal values

keyword=dd/mm/yyyy for dates (ISO ordering, not local)

keyword=hh:mm:ss for times (always 24 hour)

keyword=dd/mm/yyyy:hh:mm:ss for full local timestamp

keyword=dd/mm/yyyy:hh:mm:ss+off for full universal timestamp (off = minutes from UT ("timezone"), use - instead of + if negative delta)

keyword={ other_keywords_or_values } - for collections of values or attributes

Only "string" fields contain non-ASCII characters (as well as translated or country-local values).

e. Keywords start with an ASCII letter ('A'-'Z', 'a'-'z', '_'), followed by ASCII alphanumeric characters ('0'-'9', 'A'-'Z', 'a'-'z', '_').

f. Commas, spaces, new-lines (CR, LF), and tab characters, when not enclosed in "string" fields, are treated as interchangeable separators.

g. "String" fields are expressed in the local codepage of the system running the CLI tool. The caller of the tool is expected to provide any required codepage mapping of this data.

4. Command-line parameters are structured as follows:

   a. Each command-line parameter must have a leading '/' or '-' character, followed by a case-insensitive ASCII keyword. If a value is to be provided with the keyword, the keyword will be followed by a colon and the value (which may be enclosed in double-quotes, as needed).

   For example:

   /ADDSYS, /DELSYS:name, -REMSYS:"name with spaces"

   b. Textual values on the command line must be provided in the local codepage of the command-line tool. The caller of the tools is expected to provide any required codepage mapping of these inputs.

   c. All CLI tools support the following command-line values, with the following default behaviors if the parameter is not provided:

      1. /N:netpath - Provides NetFinity-style network path to target system. For example:

         /N:NETBIOS::PRIMM,

         or

         /N:TCPIP::primm.raleigh.ibm.com::NETBIOS::NetFin

         If not provided, local system is target system.

      2. /S:"sysname" - Provides optional system name for target system. For exapmle:

         /S:"Mike's system"

         No optional system name, if not provided.

      3. /? - Print command-line help for tool (if any)

      4. Standard parameters are not order sensitive, and tool-specific parameters are not order-specific.

      5. The presence of any unsupported or inappropriate parameters results in no action, with a return code of NFCMDLINE_BAD_COMMAND_LINE.

      6. Response file support may optionally be provided by a tool. The response file must be provided with a leading '@'. The contents of the response file are processed as if they were substituted for the response file option on the command line (with new-line and tab characters replaced with spaces).

7. Command-line help, logos, copyright statements, error messages, and other human-oriented outputs are directed to 'stderr', not ' stdout' (to prevent any difficulties in parsing the informational output directed to 'stdout', particularly due to NLS issues and the like).

## B.3 Return Codes

A table showing the basic outline of return codes for the command line interface follows:

*Table 5. Return Codes - Overview*

| r c = | name NFCMDLINE_* | Description |
|---|---|---|
| 0 | SUCCESS | Operation requested was completed successfully. |
| 1 | SYSTEM_UNAVAILABLE | Target system is unavailable or unreachable. |
| 2 | SERVICE_DOES_NOT_EXIST | Needed service(s) on target system is not available or defined. |
| 3 | NOT_AUTHORIZED | Authorization for needed service(s) on target system is not available. |
| 4 | TIMEOUT | Communication with target system failed due to timeout. |
| 5 | COMMUNICATION_ERROR | Communication with target system failed (non-timeout). |
| 6 | SERVICE_BUSY | Needed service(s) on target system are available, but currently not available for use. |
| 7 | SVC_NODE_CREATE_FAILED | Attempt to create ServiceNode failed (probably missing DLLs). |
| 8 | SVC_START_FAILED | Request to start required service(s) on target system failed. |
| 9 | SVC_STOP_FAILED | Request to stop required service(s) on target system failed. |
| 10 | BAD_COMMAND_LINE | Command line syntax error or invalid input. |
| 11 | FUNCTION_NOT_SUPPORTED | Requested operation is not supported on the service(s) on the target machine. |
| 12 | FILE_NOT_FOUND | Requested operation failed due to not finding needed local file. |
| 13 | FILE_OUTPUT_ERROR | Requested operation failed due I/O error to local file. |
| 14 - 199 | | Reserved for future standard error levels. |
| 200 - 399 | | Tool-specific error codes (for non-retryable errors). |
| 400 - 599 | | Tool-specific error codes (for retryable errors). |
| 600 - 799 | | Tool-specific information codes (for non-errors) |

## B.3.1 Tool-Specific Return Codes

The following table breaks out the return codes for the specific services in NetFinity that use the command-line interface:

| Table 6 (Page 1 of 2). Tool-Specific Return Codes | | |
|---|---|---|
| **Tool** | **r c =** | **description** |
| Remote System Manager | 200 | Group name not defined |
| | 201 | System name not defined |
| | 202 | Group tag not defined |
| | 203 | System tag not defined |
| | 204 | Bad OS type |
| | 205 | Bad protocol type |
| | 206 | Bad ping interval |
| | 207 | Bad notification severity |
| | 208 | Bad autodiscovery interval |
| Security Manager | 200 | Inbound user ID not defined |
| | 201 | Outbound node not defined |
| | 202 | Bad password |
| | 203 | Bad user ID |
| | 204 | Bad service |
| System Profile | 200 | Attribute missing |
| | 201 | Attribute invalid |
| | 202 | Value missing |
| | 203 | Value invalid |
| | 204 | Input file missing |
| | 205 | Date format error |
| | 206 | Date invalid |
| | 207 | Date out-of-range |
| | 209 | Time format error |
| | 210 | Time invalid |
| | 211 | Time out-of-range |
| Alert Manager | 200 | Alert log tag not valid |
| | 201 | Alert action tag not valid |
| | 202 | Alert handler not valid |
| | 203 | Alert action match not found |
| | 204 | Invalid alert type |
| | 205 | Invalid alert severity |
| | 206 | Invalid application alert type |
| | 207 | Invalid application ID |
| | 208 | Profile in use by existing actions |
| | 209 | Profiles not supported |
| | 210 | Profile tag/name not valid |

| Table 6 (Page 2 of 2). Tool-Specific Return Codes | | |
|---|---|---|
| **Tool** | **r c =** | **description** |
| System Monitor | 200 | Unknown monitor or attribute group ID |
| | 201 | Unknown monitor or attribute group name |
| | 202 | Unknown monitor or attribute group threshold name |
| Critical File Monitor | 200 | Fully qualified path missing |
| | 201 | Fully qualified path invalid |
| | 202 | Filename (without path) is missing |
| | 203 | Filename (without path) is invalid |
| | 204 | Severity missing |
| | 205 | Severity invalid |
| | 206 | Severity out of range |
| Process Manager | 200 | Process monitor tag/name not valid |
| | 201 | Run command failed |
| System Information | 200 | Error writing to history file |
| | 201 | Error writing to report file |
| | 202 | Error during database export |
| | 203 | Unable to execute sysinfo task (SINFG30.EXE) |
| System Configuration Manager | 200 | Means of dealing with service configuration not found.  Generally a service name specified incorrectly or a missing SCF*.DLL |

# Appendix C.  Miscellaneous Information for the LMU Chapter

This appendix includes the LMU control file as well as the startup procedure that was used during the LMU and NetFinity 5.0 comparisons.

## C.1  LMU.CTL

```
################################################################
#    IDENTIFIES THE PATH AND FILENAME OF THE LMU PROFILE.      #
#    THE LMU PROFILE MUST RESIDE IN THE ROOT DIRECTORY OF      #
#    THE OS/2 BOOT DRIVE AND HAVE THE FILENAME LMU.INI         #
################################################################


DEFINE_PROFILE INI_FILE(C:\LMU.INI)


################################################################
#                                                              #
#    THE FOLLOWING PARAMETERS DO NOT HAVE A DEFAULT VALUE      #
#    AND MUST BE MODIFIED BEFORE USING THE COMPONENT WHICH     #
#    REFERENCES THEM:                                          #
#                                                              #
#      MANAGING_SYSTEM                                         #
#      MANAGING_SYSTEM_WITH_DATABASE                           #
#      FAULT_MANAGER                                           #
#      GRAPHICAL_USER_INTERFACE                                #
#      SNMP_PROXY_AGENT                                        #
#      GUI_QUERY_MANAGING_SYSTEMS                              #
#      PROXY_QUERY_MANAGING_SYSTEMS                            #
#      SNMP_PROXY_INFORMATION                                  #
#                                                              #
################################################################


################################################################
#    THE FOLLOWING PARAMETERS APPLY TO ALL WORKSTATIONS.       #
################################################################


    # The computername or internetwork address specified
    # identifies this workstation's managing system.
    # Ex. LMUMANG (IBM requester)
    #          or
    # Ex. 000000A1:100012345678 (NetWare requester)

    APP(LMU_UTILITY),
        KEY(MANAGING_SYSTEM),
        ASCIIZ(WTRO5144);


    # The computername or internetwork address specified
    # identifies this workstation's managing system with database,
    # which is the system maintaining the LMU database.
    # Ex. LMUMANG (IBM requester)
    #          or
    # Ex. 000000A1:100012345678 (NetWare requester)

    APP(LMU_UTILITY),
        KEY(MANAGING_SYSTEM_WITH_DATABASE),
```

**267**

```
                    ASCIIZ(WTRO5144);


          # The computername or internetwork address specified identifies this
          # workstation's fault manager, which is the system to receive alerts
          # generated by the LMU applications.
          # Ex. FAULTMAN (IBM requester)
          #          or
          # Ex. 000000A1:100012345678 (NetWare requester)

          APP(LMU_UTILITY),
              KEY(FAULT_MANAGER),
              ASCIIZ(WTRO5144);


          # Identifies the location of the file to contain
          # the messages issued by LMU.

          APP(LMU_UTILITY),
              KEY(MESSAGE_LOG),
              ASCIIZ(C:\LMU2\LMU.LOG);


          # Symbolic name for workstation
          # Ex. WKSTA001

      #   APP(LMU_UTILITY),
      #       KEY(SYMBOLIC_NAME),
      #       ASCIIZ(symbolic name);


          # Identifies the LAN adapter used in NetBIOS communications.
          # Value 00 indicates the primary adapter and value 01 indicates
          # the secondary adapter.  This key is optional and if not
          # specified the primary adapter (00) will be used.
          #
          # NOTE: The hexnum value for LAN_ADAPTER must be specified
          # as 2 hexadecimal digits (for example, 01).

          APP(LMU_UTILITY),
              KEY(LAN_ADAPTER),
              HNUM(00);


          # Identifies the location in which the *.BND files were
          # installed.
          # Managing System and SNMP Proxy Agent workstations using
          # database ONLY.

          APP(LMU_UTILITY),
              KEY(BIND),
              ASCIIZ(C:\LMU2);

   ###############################################################
   #    THE FOLLOWING PARAMETERS APPLY TO "MANAGED SYSTEMS".         #
   ###############################################################

          # Identifies the frequency in minutes that the heartbeat
          # function will send a message to the managing system.
```

```
    #
    # NOTE: The hexnum value must be specified as 4 hexadecimal
    #       digits, e.g. (000A) to indicate 10 minutes.
    #
    # If '0000' is specified only the initial and terminal
    # heartbeats are sent.

    APP(LMU_UTILITY),
        KEY(PULSE_RATE),
        HNUM(0005);


    # Identifies the type of command authorization being used.
    # Valid values are REJECT, ADMIN, USER, or NONE.

    APP(LMU_UTILITY),
        KEY(SECURITY),
        ASCIIZ(NONE);


    # Identifies the fully-qualified file that contains
    # a list of workstations authorized to send commands
    # to this workstation.  Specify an "*" when no table.

    APP(LMU_UTILITY),
        KEY(AUTHORIZATION_TABLE),
        ASCIIZ(*);

###################################################################
#    THE FOLLOWING PARAMETERS APPLY TO "MANAGING SYSTEMS".        #
###################################################################

    # Identifies the file to contain
    # the node description change log.

    APP(LMU_UTILITY),
        KEY(CHANGE_LOG),
        ASCIIZ(C:\LMU2\CHANGE.LOG);


    # Identifies the location to which transferred
    # files are to written.

    APP(LMU_UTILITY),
        KEY(FILE_PATH),
        ASCIIZ(C:\LMU2);

###################################################################
#    THE FOLLOWING PARAMETERS APPLY TO "ADMINISTRATOR"            #
#    WORKSTATIONS THAT ISSUE REMOTE COMMANDS.                     #
###################################################################

    # Indicates the user exit that checks command authorization.
    # Specify the .DLL (without the .DLL extension) that
    # checks command authorization.

 #   APP(LMU_UTILITY),
 #       KEY(COMMAND_AUTHORIZATION_EXIT),
 #       ASCIIZ(LMUAXNOS);
```

```
##############################################################
#    THE FOLLOWING PARAMETERS APPLY TO "FAULT MANAGER"          #
#    WORKSTATIONS.                                             #
##############################################################

    # Identifies the Fault Manager's input user table.
    # For example C:\LMU2\AUEUSER.TAB

    APP(LMU_UTILITY),
        KEY(FAULT_TABLE),
        ASCIIZ(C:\LMU2\AUEUSER.TAB);


    # Alerts can be forwarded to a specific adapter address if desired.
    # This key is optional and if not specified the default LAN management
    # functional address of 'C00000002000' is used.

    APP(LMU_UTILITY),
        KEY(FM_FORWARDING_ADDR),
        ASCIIZ(C00000002000);


    # Identifies the computer names and/or internetwork addresses
    # of the GUI workstations to the FAULT MANAGER.
    #
    # Note: The character values for this field must be specified as
    # character fields separated by a comma.
    # Ex. (LMUGUI,000000A1:100012345678)

    APP(LMU_UTILITY),
        KEY(GRAPHICAL_USER_INTERFACE),
        ASCIIZ(WTR05144);


    # Identifies the computer names and/or internetwork addresses
    # of the SNMP Proxy Agent workstations to the FAULT MANAGER.
    #
    # Note: The character values for this field must be specified as
    # character fields separated by a comma.
    # Ex. (LMUSNMPD,000000A1:100012345678)

    APP(LMU_UTILITY),
        KEY(SNMP_PROXY_AGENT),
        ASCIIZ(WTR05144);

##############################################################
#    THE FOLLOWING PARAMETERS APPLY TO "SCHEDULER"             #
#    WORKSTATIONS.                                             #
##############################################################

    # Identifies the path and file name of the Schedule log file
    # For example C:\LMU2\SCHEDULE.LOG

    APP(LMU_UTILITY),
        KEY(SCHEDULE_LOG),
        ASCIIZ(C:\LMU2\SCHEDULE.LOG);
```

```
# Identifies the path and file name of the Schedule file
# For example C:\LMU2\SCHEDULE.TIM

APP(LMU_UTILITY),
    KEY(SCHEDULE_FILE),
    ASCIIZ(C:\LMU2\SCHEDULE.TIM);


# Identifies the path and file name of the Schedule group file
# For example C:\LMU2\SCHEDULE.GRP

APP(LMU_UTILITY),
    KEY(SCHEDULE_GROUP_FILE),
    ASCIIZ(C:\LMU2\SCHEDULE.GRP);


# Identifies the maximum number of tasks that the Scheduler
# will start to execute concurrent commands (Range 1 - 30).
# For example 5

APP(LMU_UTILITY),
    KEY(SCHEDULE_MAX_TASKS),
    ASCIIZ(30);


# Identifies the frequency in minutes that the schedule
# file will be checked for changes.
# For example 60

APP(LMU_UTILITY),
    KEY(SCHEDULE_READ),
    ASCIIZ(60);

################################################################
#   THE FOLLOWING PARAMETERS APPLY TO "ADMINISTRATOR"          #
#   WORKSTATIONS RUNNING THE GRAPHICAL USER INTERFACE (GUI).   #
################################################################

# Indicates managing systems to be queried at GUI startup.
#
# Note: The character values for this field must be specified as
# character fields separated by a comma.
# For example:  (LmuMgr,HelpDesk,Ops1).

APP(LMU_UTILITY),
    KEY(GUI_QUERY_MANAGING_SYSTEMS),
    ASCIIZ(WTR05144);


# Indicates the resource layout limit to be used when displaying
# the GUI views.
#
# Note: The character value for this field must be specified as
# a 3-digit number.  For example: (020) to indicate that up to
# 20 resources will be displayed as an ellipse, while 21 or more
# will be displayed in rows and columns.

APP(LMU_UTILITY),
    KEY(GUI_RESOURCE_LAYOUT_LIMIT),
```

```
                ASCIIZ(O2O);

        # Indicates the type of view to be displayed at GUI startup.
        #
        # The default is "A" for all managing and managed systems.

        APP(LMU_UTILITY),
            KEY(GUI_INITIAL_DISPLAY),
            ASCIIZ(A);


        # Indicates if the view is to be refreshed when a resource is
        # added due to the arrival of an event for an unknown workstation
        # that matches the type of resources being displayed.
        # Coding N or n indicates that an automatic refresh is not done.
        #
        # The default is "Y".

        APP(LMU_UTILITY),
            KEY(GUI_AUTO_REFRESH),
            ASCIIZ(Y);


        # Indicates if the user wishes to be notified of the arrival of an alert
        # by several beeps and the flashing of the title bar or minimized icon.
        #
        # The default is "Y".

        APP(LMU_UTILITY),
            KEY(GUI_BEEP_AND_FLASH),
            ASCIIZ(Y);


        # Indicates if events received for unknown managed systems should be ignored.
        # When disabled, a managed system will be created when alerts are received for
        # previously unknown managed systems.
        #
        # The default is "N".

        APP(LMU_UTILITY),
            KEY(GUI_IGNORE_UNKNOWNS),
            ASCIIZ(N);


        # Indicates if the computername or internetwork address of each
        # managed or managing system (ie, the first text field) is to be
        # displayed.
        #
        # The default is "Y".

        APP(LMU_UTILITY),
            KEY(GUI_DISPLAY_NAME),
            ASCIIZ(Y);

        # Indicates if the node address of each managed or managing system
        # (ie, the second text field) is to be displayed.
        #
        # The default is "Y".
```

```
APP(LMU_UTILITY),
    KEY(GUI_DISPLAY_NODE_ADDRESS),
    ASCIIZ(Y);
```

```
# Indicates if the symbolic name for each managed or managing system
# (ie, the third text field) is to be displayed.
#
# The default is "Y".
```

```
APP(LMU_UTILITY),
    KEY(GUI_DISPLAY_SYMBOLIC_NAME),
    ASCIIZ(Y);
```

```
# Indicates if the userid for each managed or managing system
# (ie, the fourth text field) is to be displayed.
#
# The default is "Y".
```

```
APP(LMU_UTILITY),
    KEY(GUI_DISPLAY_USERID),
    ASCIIZ(Y);
```

```
# Indicates if the icon of each managed or managing system is to be
# displayed.
#
# The default is "Y".
```

```
APP(LMU_UTILITY),
    KEY(GUI_DISPLAY_ICON),
    ASCIIZ(Y);
```

```
# Indicates if the links between the various resources should
# be displayed.
#
# The default is "N".
```

```
APP(LMU_UTILITY),
    KEY(GUI_DISPLAY_LINKS),
    ASCIIZ(N);
```

```
# Indicates the background color of the view window and of the
# legend.
#
# The default is "White".
```

```
APP(LMU_UTILITY),
    KEY(GUI_VIEW_WINDOW_COLOR),
    ASCIIZ(White);
```

```
# Indicates the color used to outline a selected resource.  This
# color is also used as the section title color in the legend
#
# The default is "Blue".
```

```
              APP(LMU_UTILITY),
                  KEY(GUI_VIEW_HIGHLIGHT_COLOR),
                  ASCIIZ(Blue);


              # Indicates the color used for the information text at the bottom
              # of the view window, the color of the three text fields associated
              # with each managed or managing system, and the color of the legend
              # description.
              #
              # The default is "Black".

              APP(LMU_UTILITY),
                  KEY(GUI_VIEW_TEXT_COLOR),
                  ASCIIZ(Black);

              # Indicates the background color used in any windows that are
              # opened as a result of submitting commands via the GUI.
              #
              # The default is "White".


              APP(LMU_UTILITY),
                  KEY(GUI_COMMAND_BACKGROUND_COLOR),
                  ASCIIZ(White);


              # Indicates the text color used in any windows that are opened
              # as a result of submitting commands via the GUI.
              #
              # The default is "Blue".

              APP(LMU_UTILITY),
                  KEY(GUI_COMMAND_TEXT_COLOR),
                  ASCIIZ(Blue);


              # Indicates which symbol is associated with the thirteen types of
              # view objects.
              #
              # NOTE: The hexnum value for this field must be specified
              # as 26 hexadecimal digits, e.g. (04030A0806020709050B01120E).

              APP(LMU_UTILITY),
                  KEY(GUI_NODE_SYMBOLS),
                  HNUM(04030A0806020709050B01120E);


              # Indicates the name and the nominal point size, in tenths, of the
              # image font used to display all text in the GUI window.
              #
              # Note: The character values for this field must be specified as a
              # character field and a 3-digit number separated by a comma,
              # e.g. (Helv,100) to indicate Helvetica 10 point.

          #   APP(LMU_UTILITY),
          #       KEY(GUI_FONTS),
          #       ASCIIZ(System Proportional,120);
```

```
    # Identifies the location of the file to contain
    # command sequences store by the GUI.

    APP(LMU_UTILITY),
        KEY(GUI_COMMANDS_TABLE),
        ASCIIZ(C:\LMU2\LMUGUI.TAB);


    # Specifies which pattern is used to indicate that a managed
    # system has received alerts or that a managing system has managed
    # systems that have received alerts.
    #
    # NOTE: The hexnum value for this field must be specified
    # as 2 hexadecimal digits, e.g. (OC) to indicate medium
    # density, diagonal hash marks.

    APP(LMU_UTILITY),
        KEY(GUI_PATTERN),
        HNUM(OC);


    # Indicates the height and width ratios that will be used as the
    # aspect ratio for the ellipse used to display the workstations.
    #
    # Note: The character values for this field must be specified as
    # two 3-digit numbers separated by a comma, e.g. (480,640) to
    # indicate an ellipse that approximates the height to width ratio
    # of a standard file monitor in 640 X 480 mode.
    # Note: specifying (001,001) will result in a circle.

    APP(LMU_UTILITY),
        KEY(GUI_COORDINATES),
        ASCIIZ(001,001);

###############################################################
#   THE FOLLOWING PARAMETERS APPLY TO "SNMP PROXY AGENT"      #
#   WORKSTATIONS.                                             #
###############################################################

    # Indicates managing systems to be queried at
    # SNMP Proxy Agent startup.
    #
    # Note: The character values for this field must be specified as
    # character fields separated by a comma.
    # For example:  (LmuMgr,HelpDesk,Ops1).

    APP(LMU_UTILITY),
        KEY(PROXY_QUERY_MANAGING_SYSTEMS),
        ASCIIZ(WTRO5144);

###############################################################
#   THE FOLLOWING PARAMETERS APPLY TO "LAN NETVIEW" (LMULNV)  #
#   WORKSTATIONS.                                             #
###############################################################

    # Identifies the LMU SNMP Proxy Agents to be queried at LMULNV
    # startup.  Specify the IP address (or host name) along with
```

```
                    # a community name for each proxy agent. The community name
                    # defaults to "public" if not specified.
                    #
                    # For example:  (·LmuProxy,public",·9.179.7.66,rtp",·9.179.7.50,").

               APP(LMU_UTILITY),
                    KEY(SNMP_PROXY_INFORMATION),
                    ASCIIZ(·WTRO5144,public");


          ################################################################
          #    THE FOLLOWING PARAMETERS APPLY TO THE APPWATCH UTILITY.    #
          ################################################################

               # Identifies the path and file name of the application watch table.
               # For example C:\LMU2\APPWATCH.SMP

               APP(LMU_UTILITY),
                    KEY(APPWATCH_TABLE),
                    ASCIIZ(C:\LMU2\APPWATCH.TAB);
```

## C.2  LMUSTART.CMD

```
@ECHO OFF
REM C:\LMU2\LMUCUST.EXE    Maintenance Level: LM00240
REM LMUCUST parameters: MANAGING MANAGED ADMINISTRATOR FAULT_MANAGER SCHEDULER PROXY_DB DOC
START "LMU Fault Manager" /C /MIN C:\LMU2\AUECATCH.EXE
START "LMU Fault Automation Manager" /C /MIN C:\LMU2\AUERECVR.EXE
DETACH C:\LMU2\AUEPITCH.EXE
START "LMU Managing System" /C /MIN C:\LMU2\LMUSRV.EXE
C:\LMU2\LMUSLEEP.EXE 45
START C:\LMU2\LMUCLI.EXE
START "LMU Scheduler" /C /MIN C:\LMU2\LMUSTEP.EXE
START "LMU SNMP Proxy Agent" /C /MIN C:\LMU2\LMUSNMPO.EXE /D 9.24.104.100 public
```

# Appendix D.  NetFinity 5.0 Text-Alert MIB

```
IBM-NetFinity-Text-Alert-MIB DEFINITIONS ::= BEGIN

        IMPORTS
                enterprises
                        FROM RFC1155-SMI
                TRAP-TYPE
                        FROM RFC1215
                OBJECT-TYPE
                        FROM RFC1212;

ibm                OBJECT IDENTIFIER ::= { enterprises 2 }

ibmProd            OBJECT IDENTIFIER ::= { ibm 6 }

netFinity          OBJECT IDENTIFIER ::= { ibmProd 71 }


--
--
-- the trap description objects
--
--

trapDesc1          OBJECT-TYPE
                   SYNTAX OCTET STRING
                   ACCESS read-only
                   STATUS mandatory
                   DESCRIPTION
                       "The text of the Netfinity generated alert"
                   ::= { netFinity 1 }

trapSystemName     OBJECT-TYPE
                   SYNTAX OCTET STRING
                   ACCESS read-only
                   STATUS mandatory
                   DESCRIPTION
                       "The NetFinity System Name where the trap occurred"
                   ::= { netFinity 2 }

trapTime           OBJECT-TYPE
                   SYNTAX OCTET STRING
                   ACCESS read-only
                   STATUS mandatory
                   DESCRIPTION
                       "The time the alert was generated"
                   ::= { netFinity 3 }

trapDate           OBJECT-TYPE
                   SYNTAX OCTET STRING
                   ACCESS read-only
                   STATUS mandatory
                   DESCRIPTION
                       "The date the alert was generated"
                   ::= { netFinity 4 }

trapSeverity       OBJECT-TYPE
                   SYNTAX INTEGER
                   ACCESS read-only
                   STATUS mandatory
                   DESCRIPTION
                       "The NetFinity severity"
                   ::= { netFinity 5 }

trapType           OBJECT-TYPE
                   SYNTAX OCTET STRING
                   ACCESS read-only
                   STATUS mandatory
                   DESCRIPTION
                       "The NetFinity Alert Type"
                   ::= { netFinity 6 }

trapApplicationID  OBJECT-TYPE
                   SYNTAX OCTET STRING
                   ACCESS read-only
                   STATUS mandatory
                   DESCRIPTION
                       "The NetFinity Application ID"
                   ::= { netFinity 7 }
```

```
trapAppType            OBJECT-TYPE
                       SYNTAX INTEGER
                       ACCESS read-only
                       STATUS mandatory
                       DESCRIPTION
                           "The NetFinity Application Alert Type"
                       ::= { netFinity 8 }

trapRecFrom            OBJECT-TYPE
                       SYNTAX OCTET STRING
                       ACCESS read-only
                       STATUS mandatory
                       DESCRIPTION
                           "The NetFinity network name of the machine that generated the alert"
                       ::= { netFinity 9 }

--
--  NetFinity SNMP trap
--

trapText1              TRAP-TYPE
                       ENTERPRISE     netFinity
                       VARIABLES      {
                                      trapDesc1,
                                      trapSystemName,
                                      trapTime,
                                      trapDate,
                                      trapSeverity,
                                      trapType,
                                      trapApplicationID,
                                      trapAppType,
                                      trapRecFrom
                                      }
                       DESCRIPTION
                               "Converted Netfinity Alert"
                        --#TYPE "NetFinity Alert Trap Information"
                        --#SUMMARY "%s "
                        --#SUMMARY "System %s "
                        --#SUMMARY "%s %s "
                        --#SUMMARY "Severity %d "
                        --#SUMMARY "Type of Alert %s "
                        --#SUMMARY "Application ID %s "
                        --#SUMMARY "Application Alert Type %d"
                        --#SUMMARY "Received From %s"
                        --#ARGUMENTS {0,1,2,3,4,5,6,7,8}
                        --#SEVERITY INFORMATIONAL
                        --#TIMEINDEX 1
                        --#STATE   OPERATIONAL
                       ::= 1

         END
```

# Appendix E.  NetFinity 5.0 Services MIB

```
TME10NETFINITYSERVICES-MIB DEFINITIONS ::= BEGIN



IMPORTS
    OBJECT-TYPE
    FROM RFC-1212
    enterprises
    FROM RFC1155-SMI
    DisplayString
    FROM RFC1213-MIB;

DmiInteger                ::= INTEGER
DmiDisplaystring    ::= DISPLAYSTRING
DmiDate                          ::= OCTET STRING (SIZE (28))
DmiComponentIndex    ::= INTEGER

NetFinityServicesMIB        OBJECT IDENTIFIER ::=  { enterprises ibm(2) ibmProd(6) netFinity(71) DmiMibs(200) 2}
dmtfGroups          OBJECT IDENTIFIER ::= {NetFinityServicesMIB  1}

SComponentid ::= SEQUENCE {
        a1Manufacturer                          DmiDisplaystring,
        a1Product                               DmiDisplaystring,
        a1Version                               DmiDisplaystring,
        a1SerialNumber                          DmiDisplaystring,
        a1Installation                          DmiDate
}

tComponentid OBJECT-TYPE
        SYNTAX        SEQUENCE OF SComponentid
        ACCESS        not-accessible
        STATUS        mandatory
        DESCRIPTION   "This group defines attributes common to all components.This
group is required."
        ::= {dmtfGroups 1}

eComponentid OBJECT-TYPE
        SYNTAX        SComponentid
        ACCESS        not-accessible
        STATUS        mandatory
        DESCRIPTION   ""
        INDEX         {DmiComponentIndex}
        ::= {tComponentid 1}

a1Manufacturer OBJECT-TYPE
        SYNTAX        DmiDisplaystring
        ACCESS        read-only
        STATUS        mandatory
        DESCRIPTION   "The name of the manufacturer that produces this component."
        ::= {eComponentid 1}

a1Product OBJECT-TYPE
        SYNTAX        DmiDisplaystring
        ACCESS        read-only
        STATUS        mandatory
        DESCRIPTION   "The name of the component."
        ::= {eComponentid 2}

a1Version OBJECT-TYPE
        SYNTAX        DmiDisplaystring
        ACCESS        read-only
        STATUS        mandatory
        DESCRIPTION   "The version for the component."
        ::= {eComponentid 3}

a1SerialNumber OBJECT-TYPE
        SYNTAX        DmiDisplaystring
        ACCESS        read-only
        STATUS        mandatory
        DESCRIPTION   "The serial number for this instance of this component."
        ::= {eComponentid 4}

a1Installation OBJECT-TYPE
        SYNTAX        DmiDate
        ACCESS        read-only
        STATUS        mandatory
        DESCRIPTION   "The time and date of the last install of this component."
        ::= {eComponentid 5}
```

```
STme10NetfinityMonitorAttributes ::= SEQUENCE {
        a3AttributeId                           DmiInteger,
        a3AttributeName                         DmiDisplaystring,
        a3CurrentValueInteger                   DmiInteger,
        a3CurrentValueThousandths               DmiInteger,
        a3CurrentValueString                    DmiDisplaystring,
        a3ValueUnits                            DmiDisplaystring,
        a3RecordingEnabled                      DmiInteger
}

tTme10NetfinityMonitorAttributes OBJECT-TYPE
        SYNTAX          SEQUENCE OF STme10NetfinityMonitorAttributes
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "Current values of TME 10 NetFinity Monitor Attributes."
        ::= {dmtfGroups 3}

eTme10NetfinityMonitorAttributes OBJECT-TYPE
        SYNTAX          STme10NetfinityMonitorAttributes
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     ""
        INDEX           {DmiComponentIndex, a3AttributeId}
        ::= {tTme10NetfinityMonitorAttributes 1}

a3AttributeId OBJECT-TYPE
        SYNTAX          DmiInteger
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Unique ID for monitored attribute."
        ::= {eTme10NetfinityMonitorAttributes 1}

a3AttributeName OBJECT-TYPE
        SYNTAX          DmiDisplaystring
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Textual name of the monitored attribute."
        ::= {eTme10NetfinityMonitorAttributes 2}

a3CurrentValueInteger OBJECT-TYPE
        SYNTAX          DmiInteger
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Integral portion of current value of attribute."
        ::= {eTme10NetfinityMonitorAttributes 3}

a3CurrentValueThousandths OBJECT-TYPE
        SYNTAX          DmiInteger
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Fraction portion of current value of attribute, in units
of 1/1000th"
        ::= {eTme10NetfinityMonitorAttributes 4}

a3CurrentValueString OBJECT-TYPE
        SYNTAX          DmiDisplaystring
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Current value of attribute, as display string."
        ::= {eTme10NetfinityMonitorAttributes 5}

a3ValueUnits OBJECT-TYPE
        SYNTAX          DmiDisplaystring
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Textual label for value units."
        ::= {eTme10NetfinityMonitorAttributes 6}

a3RecordingEnabled OBJECT-TYPE
        SYNTAX          INTEGER {
            vFalse                                      (0),
            vTrue                                       (1)
        }
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Flag indicates if data recording is active."
        ::= {eTme10NetfinityMonitorAttributes 7}

STme10NetfinityAttributeGroups ::= SEQUENCE {
        a4AttributeGroupId                      DmiInteger,
        a4AttributeId                           DmiInteger,
        a4AttributeGroupName                    DmiDisplaystring,
```

```
        a4AttributeName                         DmiDisplaystring,
        a4CurrentValueTextual                   DmiDisplaystring,
        a4CurrentValueIndex                     DmiInteger
}

tTme10NetfinityAttributeGroups OBJECT-TYPE
        SYNTAX          SEQUENCE OF STme10NetfinityAttributeGroups
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "Current values of TME 10 NetFinity Attribute Groups."
        ::= {dmtfGroups 4}

eTme10NetfinityAttributeGroups OBJECT-TYPE
        SYNTAX          STme10NetfinityAttributeGroups
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     ""
        INDEX           {DmiComponentIndex, a4AttributeGroupId, a4AttributeId}
        ::= {tTme10NetfinityAttributeGroups 1}

a4AttributeGroupId OBJECT-TYPE
        SYNTAX          DmiInteger
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Unique ID for attribute group."
        ::= {eTme10NetfinityAttributeGroups 1}

a4AttributeId OBJECT-TYPE
        SYNTAX          DmiInteger
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Unique ID for attribute within attribute group."
        ::= {eTme10NetfinityAttributeGroups 2}

a4AttributeGroupName OBJECT-TYPE
        SYNTAX          DmiDisplaystring
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Textual name of the attribute group."
        ::= {eTme10NetfinityAttributeGroups 3}

a4AttributeName OBJECT-TYPE
        SYNTAX          DmiDisplaystring
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Textual name of the attribute within the attribute group."
        ::= {eTme10NetfinityAttributeGroups 4}

a4CurrentValueTextual OBJECT-TYPE
        SYNTAX          DmiDisplaystring
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Current value of attribute, as text."
        ::= {eTme10NetfinityAttributeGroups 5}

a4CurrentValueIndex OBJECT-TYPE
        SYNTAX          DmiInteger
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Index of current value of attribute. Index = -1 if attribute's
value is not indexed."
        ::= {eTme10NetfinityAttributeGroups 6}

END
```

# Appendix F. NetFinity 5.0 Manager MIB

```
TME10NETFINITYMANAGER-MIB DEFINITIONS ::= BEGIN


 -- Manager package for TME 10 NetFinity

IMPORTS
    OBJECT-TYPE
    FROM RFC-1212
    enterprises
    FROM RFC1155-SMI
    DisplayString
    FROM RFC1213-MIB;

DmiInteger                   ::= INTEGER
DmiOctetstring               ::= OCTET STRING
DmiDisplaystring    ::= DISPLAYSTRING
DmiDate                         ::= OCTET STRING (SIZE (28))
DmiComponentIndex   ::= INTEGER

NetFinityManagerMIB   OBJECT IDENTIFIER ::=  { enterprises ibm(2) ibmProd(6) netFinity(71) DmiMibs(200) 3}
dmtfGroups           OBJECT IDENTIFIER ::= {NetFinityManagerMIB  1}

SComponentid ::= SEQUENCE {
      a1Manufacturer                          DmiDisplaystring,
      a1Product                               DmiDisplaystring,
      a1Version                               DmiDisplaystring,
      a1SerialNumber                          DmiDisplaystring,
      a1Installation                          DmiDate
}

tComponentid OBJECT-TYPE
      SYNTAX        SEQUENCE OF SComponentid
      ACCESS        not-accessible
      STATUS        mandatory
      DESCRIPTION   "This group defines attributes common to all components.This
group is required."
      ::= {dmtfGroups 1}

eComponentid OBJECT-TYPE
      SYNTAX        SComponentid
      ACCESS        not-accessible
      STATUS        mandatory
      DESCRIPTION   ""
      INDEX         {DmiComponentIndex}
      ::= {tComponentid 1}

a1Manufacturer OBJECT-TYPE
      SYNTAX        DmiDisplaystring
      ACCESS        read-only
      STATUS        mandatory
      DESCRIPTION   "The name of the manufacturer that produces this component."
      ::= {eComponentid 1}

a1Product OBJECT-TYPE
      SYNTAX        DmiDisplaystring
      ACCESS        read-only
      STATUS        mandatory
      DESCRIPTION   "The name of the component."
      ::= {eComponentid 2}

a1Version OBJECT-TYPE
      SYNTAX        DmiDisplaystring
      ACCESS        read-only
      STATUS        mandatory
      DESCRIPTION   "The version for the component."
      ::= {eComponentid 3}

a1SerialNumber OBJECT-TYPE
      SYNTAX        DmiDisplaystring
      ACCESS        read-only
      STATUS        mandatory
      DESCRIPTION   "The serial number for this instance of this component."
      ::= {eComponentid 4}

a1Installation OBJECT-TYPE
      SYNTAX        DmiDate
      ACCESS        read-only
      STATUS        mandatory
```

```
                     DESCRIPTION   "The time and date of the last install of this component."
                     ::= {eComponentid 5}

          SRemoteSystems ::= SEQUENCE {
                 a11SystemTag                        DmiInteger,
                 a11SystemName                       DmiDisplaystring,
                 a11ProtocolName                     DmiDisplaystring,
                 a11NetworkAddress                   DmiDisplaystring,
                 a11SystemState                      DmiInteger,
                 a11Server                           DmiInteger,
                 a11Manager                          DmiInteger,
                 a11OperatingSystemType              DmiInteger,
                 a11OsMajorVersion                   DmiInteger,
                 a11OsMinorVersion                   DmiInteger,
                 a11SystemModelId                    DmiOctetstring,
                 a11SystemModelName                  DmiDisplaystring,
                 a11SystemOn-lineNotify              DmiInteger,
                 a11SystemOff-lineNotify             DmiInteger,
                 a11PresenceCheckInterval            DmiInteger,
                 a11MacAddress                       DmiOctetstring
          }

          tRemoteSystems OBJECT-TYPE
                 SYNTAX        SEQUENCE OF SRemoteSystems
                 ACCESS        not-accessible
                 STATUS        mandatory
                 DESCRIPTION   "Table of remote systems defined for managing under the Remote
          System Manager service."
                 ::= {dmtfGroups 11}

          eRemoteSystems OBJECT-TYPE
                 SYNTAX        SRemoteSystems
                 ACCESS        not-accessible
                 STATUS        mandatory
                 DESCRIPTION   ""
                 INDEX         {DmiComponentIndex, a11SystemTag}
                 ::= {tRemoteSystems 1}

          a11SystemTag OBJECT-TYPE
                 SYNTAX        DmiInteger
                 ACCESS        read-only
                 STATUS        mandatory
                 DESCRIPTION   "Unique ID number of this remote system in Remote System Manager."
                 ::= {eRemoteSystems 1}

          a11SystemName OBJECT-TYPE
                 SYNTAX        DmiDisplaystring
                 ACCESS        read-only
                 STATUS        mandatory
                 DESCRIPTION   "Displayable name of the given system."
                 ::= {eRemoteSystems 2}

          a11ProtocolName OBJECT-TYPE
                 SYNTAX        DmiDisplaystring
                 ACCESS        read-only
                 STATUS        mandatory
                 DESCRIPTION   "This is the type of network protocol used for communication
          with the given system."
                 ::= {eRemoteSystems 3}

          a11NetworkAddress OBJECT-TYPE
                 SYNTAX        DmiDisplaystring
                 ACCESS        read-only
                 STATUS        mandatory
                 DESCRIPTION   "This is the network address of the given system."
                 ::= {eRemoteSystems 4}

          a11SystemState OBJECT-TYPE
                 SYNTAX            INTEGER {
                       vOff-line                            (0),
                       vOn-line                             (1),
                       vOff-lineWithErrorCondition          (2),
                       vOn-lineWithErrorCondition           (3)
                 }
                 ACCESS        read-only
                 STATUS        mandatory
                 DESCRIPTION   "This reflects the on-line/off-line state of the given system."
                 ::= {eRemoteSystems 5}

          a11Server OBJECT-TYPE
                 SYNTAX            INTEGER {
                       vFalse                               (0),
                       vTrue                                (1)
```

```
        }
        ACCESS      read-only
        STATUS      mandatory
        DESCRIPTION  "This indicates if the system is some sort of server system."
        ::= {eRemoteSystems 6}


a11Manager OBJECT-TYPE
        SYNTAX      INTEGER {
                vFalse                                  (0),
                vTrue                                   (1)
        }
        ACCESS      read-only
        STATUS      mandatory
        DESCRIPTION  "This indicates if the system is a TME 10 NetFinity Manager
system."
        ::= {eRemoteSystems 7}


a11OperatingSystemType OBJECT-TYPE
        SYNTAX      INTEGER {
                vUnknown                                (0),
                vIbmOs2                                 (1),
                vMicrosoftWindows                       (2),
                vNovellNetware                          (3),
                vMicrosoftWindowsNt                     (4),
                vIbmAix                                 (5),
                vBanyanVines                            (6),
                vIbmPc-dos                              (7),
                vScoXenix                               (8),
                vUnixSystemV                            (9),
                vMicrosoftWindows95                     (11)
        }
        ACCESS      read-only
        STATUS      mandatory
        DESCRIPTION  "The primary operating system running on the given system."
        ::= {eRemoteSystems 8}


a11OsMajorVersion OBJECT-TYPE
        SYNTAX      DmiInteger
        ACCESS      read-only
        STATUS      mandatory
        DESCRIPTION  "The major version of primary operating system running on
the given system."
        ::= {eRemoteSystems 9}


a11OsMinorVersion OBJECT-TYPE
        SYNTAX      DmiInteger
        ACCESS      read-only
        STATUS      mandatory
        DESCRIPTION  "The minor version of primary operating system running on
the given system."
        ::= {eRemoteSystems 10}


a11SystemModelId OBJECT-TYPE
        SYNTAX      DmiOctetstring
        ACCESS      read-only
        STATUS      mandatory
        DESCRIPTION  "These are the model identifier values for the given system.
 Systems which are the same non-zero model identifier are the same type of
system."
        ::= {eRemoteSystems 11}


a11SystemModelName OBJECT-TYPE
        SYNTAX      DmiDisplaystring
        ACCESS      read-only
        STATUS      mandatory
        DESCRIPTION  "This is the model name for the given system, if known."
        ::= {eRemoteSystems 12}


a11SystemOn-lineNotify OBJECT-TYPE
        SYNTAX      INTEGER {
                vSev0                                   (0),
                vSev1                                   (1),
                vSev2                                   (2),
                vSev3                                   (3),
                vSev4                                   (4),
                vSev5                                   (5),
                vSev6                                   (6),
                vSev7                                   (7),
                vNoDefault                              (254),
                vDisabled                               (255)
        }
        ACCESS      read-only
        STATUS      mandatory
```

```
              DESCRIPTION   "Severity of alert generated when system becomes on-line,
if enabled."
       ::= {eRemoteSystems 13}


a11SystemOff-lineNotify OBJECT-TYPE
       SYNTAX      INTEGER {
              vSev0                                (0),
              vSev1                                (1),
              vSev2                                (2),
              vSev3                                (3),
              vSev4                                (4),
              vSev5                                (5),
              vSev6                                (6),
              vSev7                                (7),
              vNoDefault                           (254),
              vDisabled                            (255)
       }
       ACCESS       read-only
       STATUS       mandatory
       DESCRIPTION   "Severity of alert generated when system becomes off-line,
if enabled."
       ::= {eRemoteSystems 14}


a11PresenceCheckInterval OBJECT-TYPE
       SYNTAX       DmiInteger
       ACCESS       read-only
       STATUS       mandatory
       DESCRIPTION   "This is the time between attempts by Remote System Manager
to verify the presence of the remote system, in seconds."
       ::= {eRemoteSystems 15}


a11MacAddress OBJECT-TYPE
       SYNTAX       DmiOctetstring
       ACCESS       read-only
       STATUS       mandatory
       DESCRIPTION   "Media Access Control (MAC) address of remote system, if known."
       ::= {eRemoteSystems 16}


SRemoteSystemGroups ::= SEQUENCE {
       a12GroupTag                         DmiInteger,
       a12GroupName                        DmiDisplaystring,
       a12RequiredKeywordsCombination      DmiInteger,
       a12Keywords                         DmiDisplaystring,
       a12SystemOn-lineNotifyDefault       DmiInteger,
       a12SystemOff-lineNotifyDefault      DmiInteger,
       a12DefaultPresenceCheckInterval     DmiInteger,
       a12DiscoveryStartFlag               DmiInteger
}


tRemoteSystemGroups OBJECT-TYPE
       SYNTAX       SEQUENCE OF SRemoteSystemGroups
       ACCESS       not-accessible
       STATUS       mandatory
       DESCRIPTION   "Table of remote system groups defined for managing under
the Remote System Manager service."
       ::= {dmtfGroups 12}


eRemoteSystemGroups OBJECT-TYPE
       SYNTAX       SRemoteSystemGroups
       ACCESS       not-accessible
       STATUS       mandatory
       DESCRIPTION   ""
       INDEX        {DmiComponentIndex, a12GroupTag}
       ::= {tRemoteSystemGroups 1}


a12GroupTag OBJECT-TYPE
       SYNTAX       DmiInteger
       ACCESS       read-only
       STATUS       mandatory
       DESCRIPTION   "Unique ID number of this system group in Remote System Manager."
       ::= {eRemoteSystemGroups 1}


a12GroupName OBJECT-TYPE
       SYNTAX       DmiDisplaystring
       ACCESS       read-only
       STATUS       mandatory
       DESCRIPTION   "Displayable name of the given group."
       ::= {eRemoteSystemGroups 2}


a12RequiredKeywordsCombination OBJECT-TYPE
       SYNTAX       INTEGER {
              vAllKeywordsMustMatch                   (0),
              vAnyOfTheKeywordsMayMatch               (1),
```

```
                    vExactlyOneOfTheKeywordsMustMatch          (2)
            }
        ACCESS      read-only
        STATUS      mandatory
        DESCRIPTION  "This reflects which whether all, any, or one of the keywords
for the group must be matched during a discovery request in order for a remote
system to be discovered."
        ::= {eRemoteSystemGroups 3}


a12Keywords OBJECT-TYPE
        SYNTAX      DmiDisplaystring
        ACCESS      read-only
        STATUS      mandatory
        DESCRIPTION  "List of keywords which are used for finding remote systems
during a discovery request."
        ::= {eRemoteSystemGroups 4}


a12SystemOn-lineNotifyDefault OBJECT-TYPE
        SYNTAX      INTEGER {
            vSev0                                 (0),
            vSev1                                 (1),
            vSev2                                 (2),
            vSev3                                 (3),
            vSev4                                 (4),
            vSev5                                 (5),
            vSev6                                 (6),
            vSev7                                 (7),
            vNoDefault                            (254),
            vDisabled                             (255)
        }
        ACCESS      read-only
        STATUS      mandatory
        DESCRIPTION  "On-line notification default for systems in this group, if
enabled."
        ::= {eRemoteSystemGroups 5}


a12SystemOff-lineNotifyDefault OBJECT-TYPE
        SYNTAX      INTEGER {
            vSev0                                 (0),
            vSev1                                 (1),
            vSev2                                 (2),
            vSev3                                 (3),
            vSev4                                 (4),
            vSev5                                 (5),
            vSev6                                 (6),
            vSev7                                 (7),
            vNoDefault                            (254),
            vDisabled                             (255)
        }
        ACCESS      read-only
        STATUS      mandatory
        DESCRIPTION  "Off-line notification default for systems in this group,
if enabled."
        ::= {eRemoteSystemGroups 6}


a12DefaultPresenceCheckInterval OBJECT-TYPE
        SYNTAX      DmiInteger
        ACCESS      read-only
        STATUS      mandatory
        DESCRIPTION  "This is the default time between attempts by Remote System
Manager to verify the presence of the system in this group, in seconds.
0 = No group-specific default."
        ::= {eRemoteSystemGroups 7}


a12DiscoveryStartFlag OBJECT-TYPE
        SYNTAX      DmiInteger
        ACCESS      read-write
        STATUS      mandatory
        DESCRIPTION  "The value of this flag is not significant.  To start a discovery
request on the given System Group, write any value to this field."
        ::= {eRemoteSystemGroups 8}


SRemoteSystemGroupMap ::= SEQUENCE {
        a13SystemTag                            DmiInteger,
        a13GroupTag                             DmiInteger
}


tRemoteSystemGroupMap OBJECT-TYPE
        SYNTAX      SEQUENCE OF SRemoteSystemGroupMap
        ACCESS      not-accessible
        STATUS      mandatory
        DESCRIPTION  "Table mapping the membership of remote system in groups in
the Remote System Manager service."
```

```
                      ::= {dmtfGroups 13}

        eRemoteSystemGroupMap OBJECT-TYPE
                SYNTAX          SRemoteSystemGroupMap
                ACCESS          not-accessible
                STATUS          mandatory
                DESCRIPTION     ""
                INDEX           {DmiComponentIndex, a13SystemTag, a13GroupTag}
                ::= {tRemoteSystemGroupMap 1}

        a13SystemTag OBJECT-TYPE
                SYNTAX          DmiInteger
                ACCESS          read-only
                STATUS          mandatory
                DESCRIPTION     "Unique ID number of remote system."
                ::= {eRemoteSystemGroupMap 1}

        a13GroupTag OBJECT-TYPE
                SYNTAX          DmiInteger
                ACCESS          read-only
                STATUS          mandatory
                DESCRIPTION     "Unique ID number of group containing the remote system."
                ::= {eRemoteSystemGroupMap 2}

        END
```

# Appendix G.  Special Notices

This publication is intended to help technical support personnel implement NetFinity 5.0 and install and use the database products that work with NetFinity 5.0.  The information in this publication is not intended as the specification of any programming interfaces that are provided by NetFinity.  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 information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness.  The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment.  While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere.  Customers attempting to adapt these techniques to their own environments do so at their own risk.

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

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

| | |
|---|---|
| AIX | AnyNet |
| AS/400 | Client Access |
| DB2 | DB2/2 |
| FFST | FFST/2 |
| IBM | Multimedia Presentation Manager/2 |
| NetFinity | NetView |
| OS/2 | Predictive Failure Analysis |
| PS/2 | RISC System/6000 |
| RMONitor | SOMobjects |
| Streamer | SystemView |
| Trouble Ticket | Ultimedia |
| 400 | |

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.

Java and HotJava are trademarks of Sun Microsystems, Inc.

| | |
|---|---|
| C + + | American Telephone and Telegraph Company, Incorporated |
| EtherLink/MC | 3Com Corporation |
| IPX | Novell, Incorporated |
| Java | Sun Microsystems, Incorporated |
| Netscape | Netscape Communications Corporation |
| NetWare | Novell, Incorporated |
| NFS | Sun Microsystems Incorporated |
| Novell | Novell, Incorporated |
| OpenView | Hewlett-Packard Company |
| Pentium | Intel Corporation |
| SMC | Standard Microsystems Corporation |
| Tivoli | Tivoli Systems Inc., an IBM Company |
| TME 10 | Tivoli Systems Inc., an IBM Company |
| VINES | Banyan Systems, Incorporated |
| Windows NT | Microsoft Corporation |
| Win32 | Microsoft Corporation |
| 3Com | 3Com Corporation |

Other trademarks are trademarks of their respective companies.

# Appendix H.  Related Publications

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

## H.1  International Technical Support Organization Publications

For information on ordering these ITSO publications see "How to Get ITSO Redbooks" on page 293.

- *LAN Management Process (Alerts/Monitoring) Using NetFinity*, SG24-4517

- *Systems Management from an NT Server Point of View*, SG24-4723

- *TME 3.0 NT - Automated Process*, SG24-4793

- *Setting Up a TME 3.0 NT Environment*, SG24-4819

- *NetFinity V5.0 Database Support*, SG24-4808

## H.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 |
| RS/6000 Redbooks Collection (HTML, BkMgr) | SBOF-7230 | SK2T-8040 |
| RS/6000 Redbooks Collection (PostScript) | SBOF-7205 | SK2T-8041 |
| Application Development Redbooks Collection | SBOF-7290 | SK2T-8037 |
| Personal Systems Redbooks Collection | SBOF-7250 | SK2T-8042 |

## H.3  Other Publications

These publications are also relevant as further information sources:

- *Windows NT Server Survival Guide*, SR23-7364

- *Windows NT Server Professional Reference*, SR23-7365

# 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 BookManager BOOKs of redbooks, type the following command:

        TOOLCAT REDBOOKS

    To get lists of redbooks:

        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.

- **REDBOOKS 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: | 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 |
| (+45) 48 14 2207 (long distance charge) | Outside North America |

- **1-800-IBM-4FAX (United States)** or **(+1)001-408-256-5422 (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 |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

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

Process Manager   40, 79, 88, 127, 148, 149, 214
Process Manager Guru   75
PUBLIC   103

## Q
QUERYVPD   179

## R
Refresh Rate   112
Remote Access Restrictions   44
remote command   120
remote session   65
Remote System Manager   11, 82, 93, 150
reports   200, 204
REXX   127, 131, 137, 216, 223, 227

## S
security   6
Security Manager   20, 134
serial connections   226
Service Configuration Event   66
Service Configuration Guru   75
Service Configuration Manager   66, 81, 103, 119
SINFG30.EXE   41
SNMP   171, 173, 190, 191, 192, 216, 226
SNMP Proxy Agent   221
snmpgets   195
snmpwalk   195
software inventory   67
SPM/2   196, 197
SPM/2 collection facility   196
SQL   187, 189
SYSTAG   14
System Information tool   41
System Monitor   33, 127, 159
System Partition   69
System Profile   21
SYSTEM.MIB   190
Systems Monitor Guru   75
SystemView Agent for OS/2   171, 193

## T
TCP/IP   14, 32, 173
thresholds   34, 119, 155, 161
topology   195
transport protocols   6, 14
trap destination   172, 175

## V
VIRALERT   222, 223
Visualizer   203, 207
Visualizer Flight   179, 189
vital product data (VPD)   21

## W
wake-on-LAN   17
Web Enhancement   3
Web Guru   46, 74, 97
Web help facility   84
Web Manager   44
Web Server Socket   44
Webability   43, 117
Windows NT   57

# ITSO Redbook Evaluation

NetFinity V5.0 Command Line and LMU Support
SG24-4925-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redeval@vnet.ibm.com

**Please rate your overall satisfaction** with this book using the scale:
**(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)**

**Overall Satisfaction**                                                  _____

**Please answer the following questions:**

Was this redbook published in time for your needs?                 Yes____  No____

If no, please explain:

_____

_____

_____

_____


What other redbooks would you like to see published?

_____

_____

_____


**Comments/Suggestions:      ( THANK YOU FOR YOUR FEEDBACK! )**

_____

_____

_____

_____

_____

**299**

**IBM** ®

Printed in U.S.A.