

# Contents

<b>1</b>	<b>Concepts.</b>	<b>1</b>
1.1	Managing networks. . . . .	1
1.2	Processes. . . . .	2
1.3	Window components. . . . .	3
<b>2</b>	<b>Installation.</b>	<b>5</b>
2.1	Hardware requirements. . . . .	5
2.2	Loading the software. . . . .	6
2.3	Required environment variables. . . . .	6
2.4	Optional environment variables. . . . .	7
2.5	RDBMS Installation. . . . .	8
2.5.1	Using NetEye with University Ingres. . . . .	8
2.6	Troubleshooting. . . . .	9
2.7	Transcript of sample installation. . . . .	10
<b>3</b>	<b>Using NetEye</b>	<b>14</b>
3.1	Creating Network Maps. . . . .	14
3.1.1	Grouping Objects. . . . .	15
3.1.2	Creating Physical Network Objects. . . . .	16
3.2	Objects Pop Up Menu. . . . .	18
3.3	Exploding a Group. . . . .	18
3.4	Deleting objects. . . . .	20
3.5	Moving objects. . . . .	20
3.6	Locating Objects. . . . .	22
3.7	Adding objects to a Group. . . . .	22
3.8	Icons . . . . .	22
3.9	Backgrounds. . . . .	23
<b>4</b>	<b>Control Panel.</b>	<b>24</b>
4.1	The Menu Bar. . . . .	24
4.2	Status Line. . . . .	26
4.3	Counters. . . . .	26
4.3.1	Resetting Counters. . . . .	27
4.4	Using Interactive Help. . . . .	27

4.5	NetEye version information. . . . .	27
4.6	The Clipboard. . . . .	29
4.6.1	Basic use of the Clipboard. . . . .	29
4.6.2	Advanced selections. . . . .	31
4.6.3	Handling interconnected networks. . . . .	31
4.6.4	The Clipboard as a temporary object store. . . . .	32
4.7	Color codes legend. . . . .	32
<b>5</b>	<b>Navigating SNMP variables</b>	<b>34</b>
5.1	MIB Browser buttons. . . . .	36
5.2	Loading private MIBs. . . . .	36
<b>6</b>	<b>Polling SNMP variables</b>	<b>37</b>
6.1	Plotting SNMP variables. . . . .	37
6.2	Plotted values. . . . .	39
6.3	Multiple plots. . . . .	39
6.4	Resetting and deleting graphs. . . . .	39
6.5	Printing graphs. . . . .	40
6.6	Zooming. . . . .	40
6.7	Saving plotted data. . . . .	40
6.8	Switching representation. . . . .	41
6.9	Setting Thresholds. . . . .	42
6.10	Actions. . . . .	43
<b>7</b>	<b>Get, GetNext and Set of SNMP variables</b>	<b>46</b>
7.1	Get and GetNext operations. . . . .	46
7.2	Set operations. . . . .	48
<b>8</b>	<b>Controlling the state of a device.</b>	<b>51</b>
8.1	Actions on state changes. . . . .	51
<b>9</b>	<b>Handling SNMP Traps.</b>	<b>54</b>
9.1	Filtering traps. . . . .	56
9.2	SNMP traps and actions. . . . .	57
9.3	Interrogating the trap database. . . . .	58
<b>10</b>	<b>Trouble Tickets.</b>	<b>61</b>
10.1	Structure of a Trouble Ticket. . . . .	63
10.2	Creating tickets. . . . .	63
10.3	Searching for tickets. . . . .	65
10.4	Updating tickets. . . . .	68
10.5	Deleting tickets. . . . .	69
10.6	Printing tickets. . . . .	70
10.7	Faxing and emailing tickets. . . . .	71

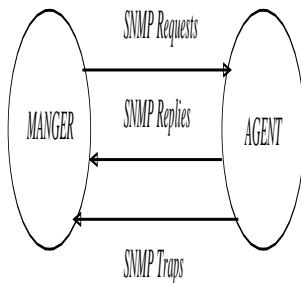
<b>11 User and configuration management.</b>	<b>72</b>
11.1 Address Book. . . . .	72
11.2 Output queues. . . . .	72
11.3 Miscellaneous configurations. . . . .	74
<b>12 Application defaults.</b>	<b>76</b>
<b>13 Appendix.</b>	<b>78</b>
13.1 Directory structure. . . . .	78
13.2 Database structure. . . . .	78
13.3 Internal protocols. . . . .	78
13.4 Structure of help files. . . . .	80

# Chapter 1

## Concepts.

### 1.1 Managing networks.

A manageable network consists of one or more network management systems and a collection of network elements. Every **management system** executes a set of network management operations that monitor and supervise each network element. The implementation of all the network management operations is termed **manager**. Every network element can be fully managed if it contains an **agent system** that is responsible for performing the network management operations requested by the manager.



The communication between management system and agents is carried out using the rules set out by **Simple Network Management Protocol**, or **SNMP**. SNMP allows a manager to retrieve (**get**) specific information from the agent. This is done by sending requests to the agent, that will reply with the requested data. Using a network management system an operator can alter (**set**) the management information contained in an agent.

An agent can send information to the manager without and explicit request; this operation is called **trap**. Traps alert the manager of changes in the agent, such as a reboot or a link down. Every agent knows to which manager it must send it's traps; this information is usually set during installation.

Not all information in an agent is accessible by every manager. SNMP defines a password that allows the manager to access the data of an agent; this password is called a **community name** and must be specified with each information request.

All the information on an agent is conceptually called the **Management Information Base** or **MIB**. The MIB is not a physically distinct database as a payroll, but rather it is a concept that encompasses all the configuration and status data of a specific agent. All MIB values must

conform to an Internet-standard **Structure Of Management Information (SMI)** defined in **RFC-1155** and compose a virtual data store on the agent.

*NetEye* uses **polling** through ICMP requests and SNMP requests to the agents to discover the network's topology and monitor it's changes. Changes are generally called **events** and can be SNMP traps from agents, status changes, topology changes, exceeded thresholds, configuration changes and network errors.

## 1.2 Processes.

This section describes all the processes that are key to the operation of *NetEye*. Included are brief explanations of the interactions and relationships among these processes.

All the components of *NetEye* are started using the script **neteye**. Most of the processes are daemons that work in background and must always be present for *NetEye* to work.

**controlp** It is the **Control Panel** and provides the graphical user interface. It interfaces with the other processes associated with network management operations.

**snmpplot** It is used to continuously monitor one or more parameters of the agents on the network and to plot the values graphically. This process is also used for threshold monitoring in combination with the **Trouble Ticketing System**.

**snmptrapd** It is a background process that listens for SNMP traps from the agents on the networks.

**netpoll** Is the background process that polls all SNMP agents. It is responsible of discovering your network topology and then detect configuration, status and topology changes.

**snmpquery** Is the background process that interacts with the **Control Panel** to **get** and **set** SNMP variables in an agent.

**applserver** It is used to handle the outgoing fax and email queues, the address book used for sending emails and faxes and several other configurable parameters.

There are three more processes that are generally invoked from the **Control Panel**:

**tt** The graphical user interface to the **Trouble Ticketing System**.

**xevent** The tool used to handle and filter SNMP traps.

The interactions and relationships among these processes are shown in figure 1.2.

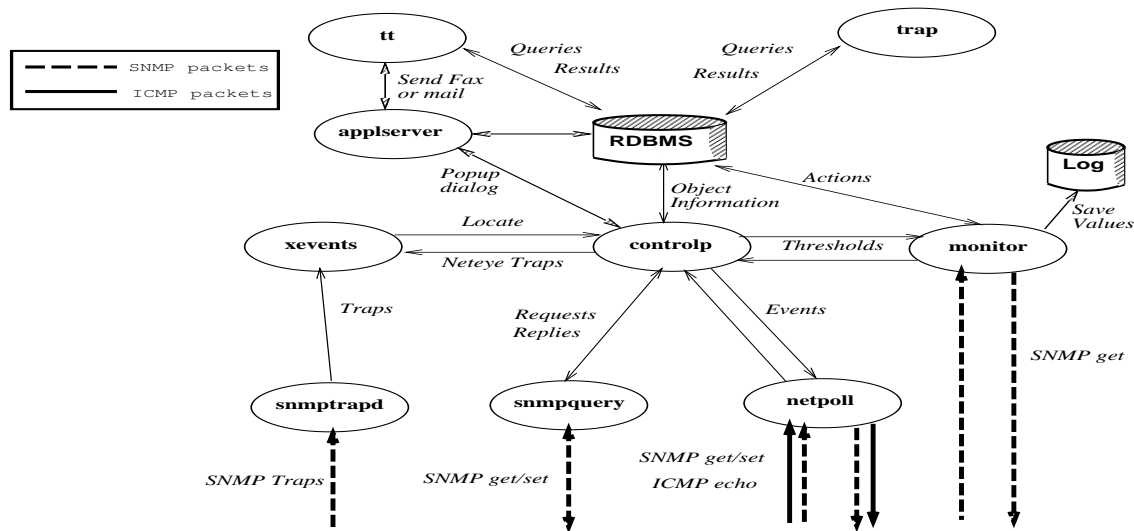


Figure 1.2: Manager processes

### 1.3 Window components.

Throughout this manual a specific terminology will be used to indicate window components; they are shown in figure 1.3 and are

**Menu bar** an area, placed at the top of a window, containing a set of buttons that invoke menus.

**Pull Down Menu** a menu activated by a button in a menu bar.

**Cascade Menu** a menu activated by another menu.

*NetEye* assumes a three button mouse and gives the following names to mouse buttons.

Button	Name	Description
1	Select	Used for selection
2	Custom	Context specific
3	Menu	Used to display pop up menus

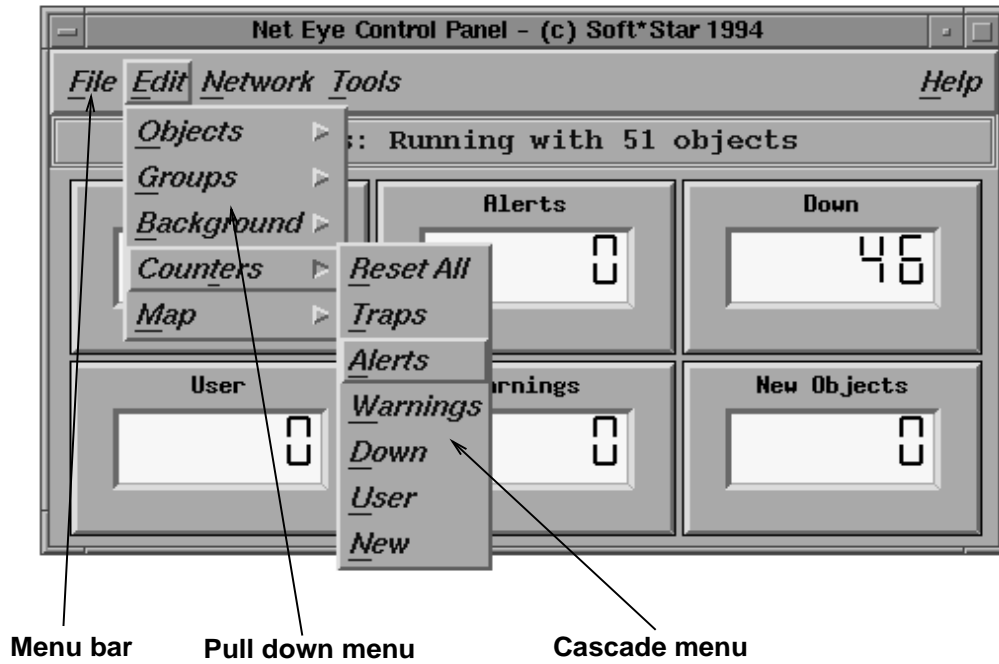
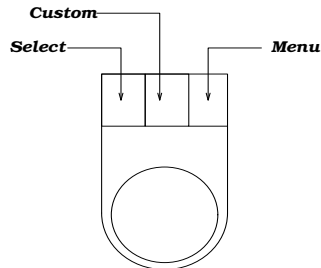


Figure 1.3: Window components



Activation of window components is done with the mouse, by pressing the *Select* button, or with the keyboard. In this case you must use the keyboard accelerators by pressing the *Alt* key and the underlined mnemonic letter associated with each button.

## Chapter 2

# Installation.

Below is a summary of the tasks required to install *NetEye* and the following sections will tell how to perform these tasks:

- choose a directory where you will load the software and extract it from the distribution media.
- setup your RDBMS system so that *NetEye* can access it.
- set some environment variables for all those users that will use *NetEye*

If you want to customize the graphical user interface refer to chapter *Application defaults*. All installation steps **must** be performed as **root**.

### 2.1 Hardware requirements.

In order to successfully run *NetEye* you need this minimum hardware:

- 80386 CPU running to least at 40 Mhz.
- 8M bytes RAM.
- Super VGA video board with matching X server.
- A three button mouse or a mouse capable of emulating three buttons.
- Ethernet card, PPP or slip interface to the outside world.

If you have a two button mouse you must configure your X server so that it will handle the three button emulation. This is usually done by adding the keyword **Emulate3Buttons** in **Xconfig** file.

**If your video board is not capable of handling at least 256 colors you will not be able to run NetEye.** For more information refer to the *Troubleshooting* section.

## 2.2 Loading the software.

If you purchased *NetEye* for *Linux* you will receive two sets of floppies; the one labeled **NetEye** contains the network management software, while the one labeled **University Ingres** contains the relational databases used by *NetEye*. The *NetEye* distribution for all other operating systems only contains *NetEye* and expects to find the RDBMS already installed and configured on your machine.

Before starting the installation be sure to create the network and database administrators' accounts. We recommend you to create a `neteye` user for network administration and a `ingres` user for database administration.

To start the installation, log in as `root` then position yourself in the directory where you want to install *NetEye* and extract the software from the distribution media (disks labeled *NetEye*) using the command `tar xvMf /dev/fd0`. The capital **M** is very important because the distribution is on a multivolume archive; you will be prompted when a new floppy disk must be inserted. When you have terminated the extraction of the files from the floppies change to directory `bin` and run `install.neteye`. This script will ask you some questions and configure the environment of the network administrator (`.profile`) and some global files (`/etc/services`). The script will also test to see if you already have *University Ingres* installed; if you don't, you will have the opportunity to do it now without terminating the *NetEye* installation script. If all goes well, when the script terminates *NetEye* will be ready to start supervising your network.

## 2.3 Required environment variables.

In order to run, *NetEye* requires you to set these environment variable; depending on the shell you use you will have to edit your `.profile` or `.login`.

**NETEYEDIR** the top directory where you loaded *NetEye* and HOME directory of user `neteye`.

**PATH** include in your PATH `$NETEYEDIR/bin`, `$NETEYEDIR/install` and all the other paths required to access your RDBMS. If you are using *University Ingres* include `$INGRES/bin`.

**MIBDIR** is the directory containing the MIB files and it must be set to `$NETEYEDIR/mib` or whatever directory you use to store your MIBs.

**MIBFILES** tells *NetEye* which MIBs you want to load automatically at startup time. This variable is a set of file names separated by a column; you must include `mib.txt` file that contains the basic MIB-2 definition. As an example consider `MIBFILES=mib.txt:cisco:` that loads the basic **MIB-2** and *Cisco* enterprise.private MIB.

**XAPPLRESDIR** specifies where application defaults are located. If you didn't copy them in `/usr/lib/X11/app-defaults` this variable should be set to `$NETEYEDIR/app-defaults`.

Failing to set the previously listed environment variables will not permit *NetEye* to startup correctly.

Depending on the type of RDBMS you are using you must set different environment variable; please refer to the RDBMS documentation. As an example here are the settings for some common RDBMS currently supported by *NetEye*:

### Ingres

```
II_SYSTEM = /usr
TERM_INGRES = $TERM
PATH = $II_SYSTEM/ingres/bin:$II_SYSTEM/ingres/utility:$PATH
II_DATE_FORMAT = MULTINATIONAL
```

### Informix On-line

```
INFORMIXDIR = /usr/informix
SQLEXEC = $INFORMIXDIR/lib/sqlturbo
DBDATE = DMY4/
DBMONEY = '$,'
DBEDIT = vi
PATH = $INFORMIXDIR/bin:$PATH
```

### Informix SE

```
INFORMIXDIR = /usr/informix
SQLEXEC = $INFORMIXDIR/lib/sqlexec
DBDATE = DMY4/
DBMONEY = '$,'
DBEDIT = vi
PATH = $INFORMIXDIR/bin:$PATH
```

### Postgres

PGDATA = the data directory

## 2.4 Optional environment variables.

*NetEye* may be tuned with the following set of optional environment variable:

**REPORTCMD** The command to use for printing trouble tickets. The default value is set to `lp`.

**MAILCMD** The command to use when sending notification by email. If not set, the default value is `mail %u < %f`.

**FAXCMD** The command to use when sending notification by fax. The default value for this variable is `fax send -l %n %h %f`.

**FAXHDR** The name of a file to be used as a header page when sending notifications by fax.

**AFMPATH** is the directory where the PostScript fonts used for printing are stored.

If this variable is not set *NetEye* will search in `$NETEYEDIR/libs/afm`.

**POLLTIME** used by the *Autodiscovery* and *Polling* facility; it controls the the number of seconds between two consecutive polls of a network object. The default value is set to 240 seconds.

**NETTIME** a numeric value that represents the mean time of network exploration. Higher the value assigned to **NETTIME** slower will work the *Autodiscovery* facility. The default value is set to 2; change the value if you want *NetEye* to use up less network bandwidth.

**DELAYPOLL** the delay, in seconds, used by the **Control Panel** before sending requests to the *Autodiscovery* facility. The default value is one second. The parameter avoids a surge of request when *NetEye* is starting up.

**MAXSNMPSOCKET** the maximum number of concurrently open sockets; the default value is 10. This parameter is used to adapt *NetEye* to the current configuration of your kernel (maximum number of open descriptors); lowering the value slows *NetEye*'s operation.

As you have seen, both **FAXCMD** and **MAILCMD** can be formatted using a syntax similar to the one used for `printf`. Each conversion specification is introduced by a `%` followed by a single character with these associated meanings:

**n** a telephone number that will be extracted from the internal address book.

**h** the name of a file to use as a header in faxes. It is generally extracted from **FAXHDR** environment variable.

**f** the body of the message.

**u** the email address of the destination.

## 2.5 RDBMS Installation.

If you are using *NetEye* with *Informix* or *Ingres* please refer to the product's installation manual. Some products like *Ingres* require you to enable a user to create a database (`accessdb` command). To create the database run the script called `installdb.your_DB_name`. This is done by default when installing *NetEye*.

### 2.5.1 Using NetEye with University Ingres.

If you are using the *Linux* or *NetBSD* version of *NetEye* you will have to install *University Ingres*. This requires you to complete the following steps:

1. create user `ingres` and it's home directory. This is typically done with `useradd`.
2. load the software in the directory you just created.

3. enable all *NetEye* users to access the RDBMS. It is usefull to enable `root` as a user. Edit `$INGRES/files/users` by adding one line for each user. This file contains the user information in fields seperated by colons. The fields are as follows:
  - User name, taken directly from `/etc/passwd` file.
  - User code, assigned by the INGRES super-user. It must be a unique two character code.
  - UNIX user id. This **MUST** match the entry in the `/etc/passwd` file.
  - UNIX group id. Same comment applies.
  - Status word in octal. Bit values are:
    - 0000001** creatdb permission
    - 0000002** permits batch update override
    - 0000004** permits update of system catalogs
    - 0000020** can use trace flags
    - 0000040** can turn off qrymod
    - 0000100** can use arbitrary proctabs
    - 0000200** can use the =proctab form
    - 0100000** ingres superuser
  - A list of flags automatically set for this user.
  - The process table to use for this user.
  - An initialization file to be read by the monitor at login time.
  - Unassigned.
  - Comma seperated list of databases. If this list is null, the user may enter any database. If it begins with a '-', the user may enter any database except the named databases. Otherwise, the user may only enter the named databases.
4. initialize database `neteye` by running `installdb` script. This can be done as `root` or any other user you defined in `/usr/ingres/files/users`.

## 2.6 Troubleshooting.

*Problem:* *NetEye* does not start and the message `Cannot open socket in raw mode` is printed.

*Solution:* verify that `netpoll` is `suid` to `root`; this is needed because `netpoll` uses `SOCK_RAW` mode to open sockets.

*Problem:* *NetEye* does not start and the message `Cannot bind trap socket` is printed.

*Solution:* verify that `snmptrapd` is `suid` to `root`; this is needed because `snmptrapd` binds a priviledged port.

*Problem:* *NetEye* does not start and the message `SYSERR: cannot exec INGRES in IIngres` is printed.

*Solution:* your PATH doesn't include the directory that contains *University Ingres* executables. Correct your .profile.

*Problem:* *NetEye* terminates and message `You are not a valid INGRES user.` is printed.

*Solution:* The uid/gid defined in `/etc/passwd` don't match those specified in `$INGRES/files/users`. Refer to section *Using NetEye with University Ingres*.

*Problem:* *NetEye* does not start and the message `Cannot read main Mib file` is printed.

*Solution:* verify that environment variables MIBDIR and MIBFILES are correctly set.

*Problem:* The graphical user interface appears garbled.

*Solution:* verify that environment variable XAPPLRESDIR is set to the directory containing application defaults files. It is usually set to `$NETEYEDIR/app-defaults`.

*Problem:* During startup you get a sequence of warnings such as `Cannot allocate colormap entry for "MediumSeaGreen"` or `XpmCreatePixmapFromData failed` and the user interface has strange colors.

*Solution:* Your video board is not capable of handling more than 16 colors. The best way to handle this problem is to use a SVGA video board, you will also get a much better resolution. If you still want to use the VGA16 Xserver try changing the colors defined in the application defaults files. It is usually set to `$NETEYEDIR/app-defaults`.

*Problem:* *NetEye* terminates and message `License has expired` is printed.

*Solution:* You probably have a demo version of *NetEye* and the trial period has finished. Please contact us if you still want to use *NetEye*. If you have a regular license probably the license file has been corrupted or edited. If you saved a backup copy of the file install it in place of the damaged one. If you don't have a backup extract it from the distribution media with command `tar xvMf /dev/fd0 etc/neteye.lic`.

## 2.7 Transcript of sample installation.

Here is the transcript of a sample installation using *Linux-1.1.20*.

```
felix:~# cd ~neteye
felix:/home/neteye# tar Mxf /dev/fd0
Prepare volume #2 for /dev/fd0 and hit return:
felix:/home/neteye# cd bin
felix:/home/neteye/bin# install.neteye
                        Installation
```

This installation shell script will examine your system and ask you questions to determine how Net Eye should be installed. Many of the questions will have default answers in square brackets--typing carriage return will give you the default.

[ Type carriage return to continue ]

Checking Data Base installation

You must load the ingres software

Do you want load ingres software now ? (y/n) y

Do you want to use drive A,B or Other? (A/B/O) [A] a

Insert floppy "Linux Ingres" in drive.

[ Type carriage return to continue ]

Wait .....

Create Ingres files in directory /home/ingres ?

Please type y to continue or n to abort this operation. (y/n) y

Continue load operation. Wait ...

Wait ... Uncompress Ingres files...

done

Ingres software is loaded

-> NetEye user name is [neteye] ?neteye

-> The home directory of neteye is [/home/neteye] ? (y/n) [Y]y

Uncompress files in /home/neteye/bin

Wait ... done

-> The Postscript font metrics files is in /home/neteye/libs/afm? (y/n) [Y]y

Install Postscript font metrics files. Wait ..

-> Enter the desired print command : [lp] lpr -Plaser

-> Enter the desired fax command

(in the fax command you can specify with %n the position of the fax number, with %h the file to use as a header of the fax and with %f the body of the message): [fax send -l %n %h %f]

-> Enter the desired mail command

(in the mail command you can specify with %u the position of the mail address and with %f the body of the message): [mail %u < %f]

INGRES=/home/ingres

PATH=\$INGRES/bin:\$PATH

export PATH INGRES

NETEYEDIR=/home/neteye

```
export NETEYEDIR
MIBDIR=$NETEYEDIR/mib
MIBFILES=mib.txt:
XAPPLRESDIR=$NETEYEDIR/app-defaults
PATH=$NETEYEDIR/bin:$PATH
AFMPATH=/home/neteye/libs/afm
export MIBDIR MIBFILES XAPPLRESDIR PATH AFMPATH
POLLTIME=240
NETTIME=2
DELAYPOLL=1
MAXSNMPSOCKET=10
export POLLTIME NETTIME DELAYPOLL XSNMPSOCKET
REPORTCMD=lpr -Plaser
MAILCMD=mail %u < %f
FAXCMD=fax send -l %n %h %f
export REPORTCMD MAILCMD FAXCMD
```

-> Append these lines in /home/neteye/.profile ? (y/n) [N]y

-> NetEye services begin from ? [8000]

[Type carriage return to continue else type new value] >

```
# NetEye used services
neteye      8001/udp
netpoll     8002/udp
snmpmonitor 8003/udp
xevents     8004/udp
controlp    8005/udp
cmdsnp      8006/udp
configure   8007/udp
```

-> Append these lines in /etc/services ? (y/n) [N]y

```
Create Ingres User neteye
Net-Eye Data Base not found
Do you want install Data Base ? (y/n) y
creating database neteye -- please wait
loading relations
(1 tuple)
database neteye created
```

Bye

```
felix:/home/neteye/bin# su - neteye
felix:~$ export DISPLAY=skiddo:0.0
```

```
felix:~$ env
NETEYEDIR=/home/neteye
HOSTNAME=felix.softstar.it
LOGNAME=neteye
MIBFILES=mib.txt:
INGRES=/home/ingres
TERM=xterm
HOSTTYPE=i386
PATH=/home/neteye/bin:/home/ingres/bin::usr/ucb:/bin:/usr/bin:/usr/etc:/usr/bin/X11:/usr/TeX
HOME=/home/neteye
SHELL=/bin/bash
PS1=\h:\w\$
PS2=>
USER=neteye
MANPATH=/usr/man:/usr/man/preformat:/usr/X11/man:/usr/openwin/man
LESS=-MM
XAPPLRESDIR=/home/neteye/app-defaults
REPORTCMD=lpr -Plaser
FAXCMD=fax send -l %n %h %f
MAILCMD=mail %u < %f
MIBDIR=/home/neteye/mib
POLLTIME=240
DISPLAY=skiddo:0.0
AFMPATH=/home/neteye/libs/afm
OPENWINHOME=/usr/openwin
NETTIME=2
DELAYPOLL=1
BASH=/bin/bash
_=/usr/bin/env

felix:~$ neteye
```

## Chapter 3

# Starting NetEye and using network maps.

To start *NetEye* just enter `neteye` at the *Unix* command prompt. After a few seconds, the **Control Panel**, the **New Node Clipboard** and the **Main Map** will appear. If it is the first time you run *NetEye* the *Main Map* will be empty; the next sections will tell you how to populate it. If it is not the first time you run *NetEye*, the screen will be populated with the same set of network maps that were visible during your last session; each map will contain a color coded iconic representation of your network.

When you start *NetEye* the **Autodiscovery** feature will begin exploring your network and log in the *New Node Clipboard* all the objects it locates that are not yet supervised by *NetEye*. All objects that are capable of handling SNMP or ICMP are recorded and presented in the *New Node Clipboard* using their official name as returned by the name server of your network. At this point it is certainly usefull to check the correct configuration of your name server or the definition of the server (usually in file `/etc/resolv.conf`) if the host you are installing *NetEye* on only acts as a DNS client.

Please note that all the functionalities described in this section that refer to the **File** and **Edit** buttons in the **Control Panel**'s menu bar are duplicated for your convineince in the menu bar of each **Network Map**.

### 3.1 Creating Network Maps.

When you start *NetEye* for the first time you will see the *Clipboard* being populated very quickly with the names of all the network objects the *Autodiscovery* detects, but the *Main Map* will remain empty. This feature enables you to create a hierarchical view of your network by selectively placing objects on the maps. To create a map you must follow these steps:

1. click with the **Select** mouse button on one or more objects in the Clipboard; as you select them they turn red. To deselect an object just click again on it.
2. place the selected objects in the map by moving the mouse to the desired map and by clicking the **Select** button. All objects will be moved from the Clipboard to the map.

As objects are moved from the *Clipboard* to the map, they will be automatically connected if they are part of the same network. Objects with multiple interfaces will appear as connected to two or more networks.

Every time you move an object from the *Clipboard* to a map or manually add an object it will be recorded in the underlying database. This enables *NetEye* to permanently manage that object.

### 3.1.1 Grouping Objects.

To create a hierarchical view of your network we must introduce the concept of

**Physical Network Object** a device physically present on your network, that has an associated IP address and is capable of handling ICMP protocol and optionally SNMP.

**Logical Network Object** is composed of one or more physical network objects and can have more than one associated IP address. There is no physical equivalent of a logical network object on the network. Logical network objects are an artifact to help organize physical objects on maps. There is no limit to the nesting of logical network objects.

In the rest of this manual we will refer to logical network objects as **Groups**. By using groups you can create maps with various levels of details, such as the corporate network, the marketing network, the R&D network, office AC345 network. Because groups can be nested, you can easily drill down from the global network map to the specific physical network object.

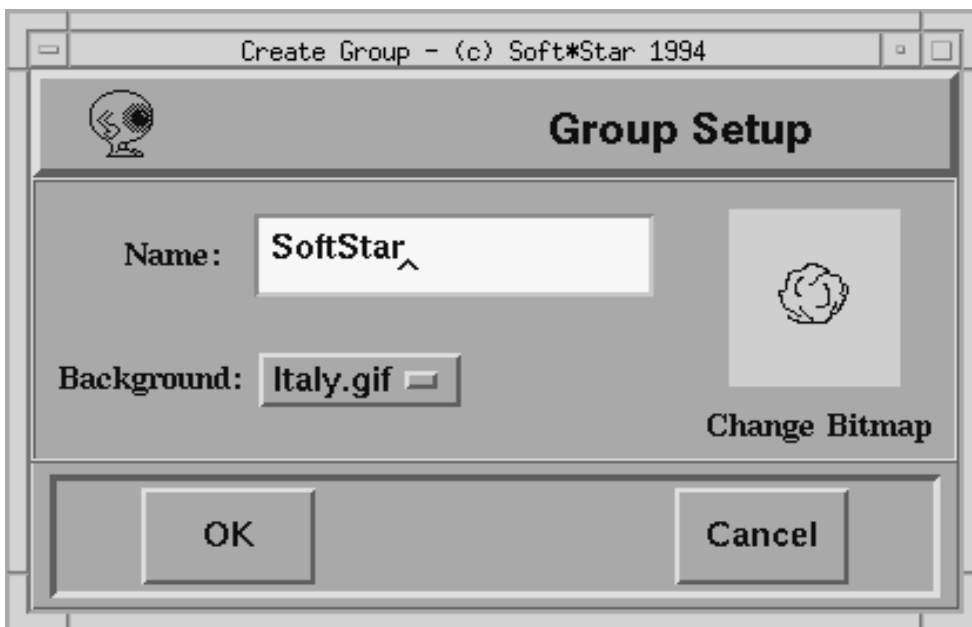


Figure 3.1 Create a Group dialog

To create a *Group* refer to the *Control Panel* and execute these steps:

1. select **Edit** from the menu bar.
2. select **Groups** and a cascade menu will appear; select **Group**.
3. a dialog box, as shown in figure 3.1. appears prompting you with then name of the new logical object. Enter the name you what to assign to the Group you are creating.
4. optionally you can associate a background to the Group by clicking on the **Background** button. It will be used when you will explode the Group.
5. to change the icon of the Group click on it's current icon; you will be shown all the available icons. Pick the one you prefer by clicking the **Select** mouse button on it. Figure 3.2 shows the icon selection dialog.
6. when you are satisfied with your data press the **OK** button.

At this point you are done inserting the data of the new Group; move the mouse to the map where you want to create it. By holding down the **Select** mouse button select an area of the map. When you release the mouse button all the objects contained in the selected area will be automatically removed from the map and inserted in the new Group. There is no limit ot the number of Groups you can create or to their nesting.

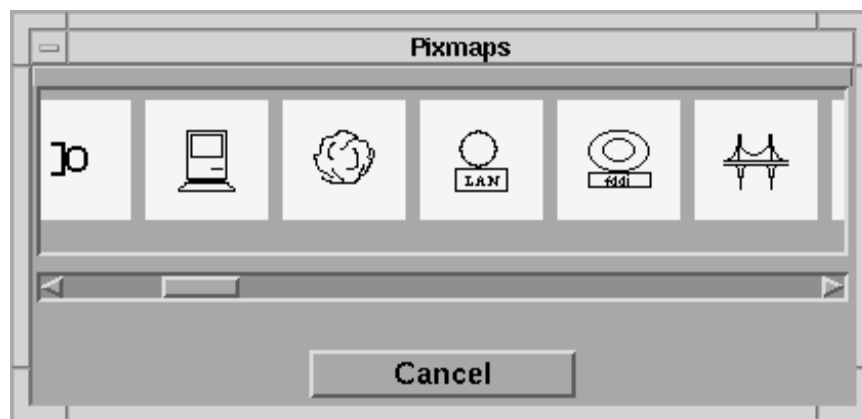


Figure 3.2: Icon selection dialog

### 3.1.2 Creating Physical Network Objects.

Sometimes it can be useful to place a *Physical Network Object* on a map even if it was not located by the autodiscovery mechanism. Maybe because the object was disconnected from the network or powered off; in anycase *NetEye* will automatically detect it when it is powered on again. To add a new object

1. select **Edit** from the *Control Panel* menu bar.

2. select **Objects** and a cascade menu will appear; select **Create**.
3. a dialog box, as shown in figure 3.3, appears prompting you with some information about the new object.
4. click with the mouse **Select** button on the position where you want to place the new object.

In this form, the only information you **must** supply is the IP address of the object in the text field labeled **IP Address**. You can use a symbolic name resolvable by your name server or a standard numeric ip address (w.x.y.z format). All the rest is optional and has the following use;

**Name** a symbolic name you can assign to the object; it will be used on the map to identify it instead of it's official IP address.

**Community/Party** the SNMPv1 community identifier or SNMPv2 party identifier.

**Frequency** the frequency, in seconds, with which *NetEye* must poll that object.

On the form there are two mutually exclusive buttons labeled **SNMP I** and **SNMP II**. They are used to select which SNMP protocol, SNMPv1 or SNMPv2, to use when polling the object you are creating. To select one or the other please refer to the object's User Manual. To change the object's icon click on it's current icon just as you did while creating a new Group.

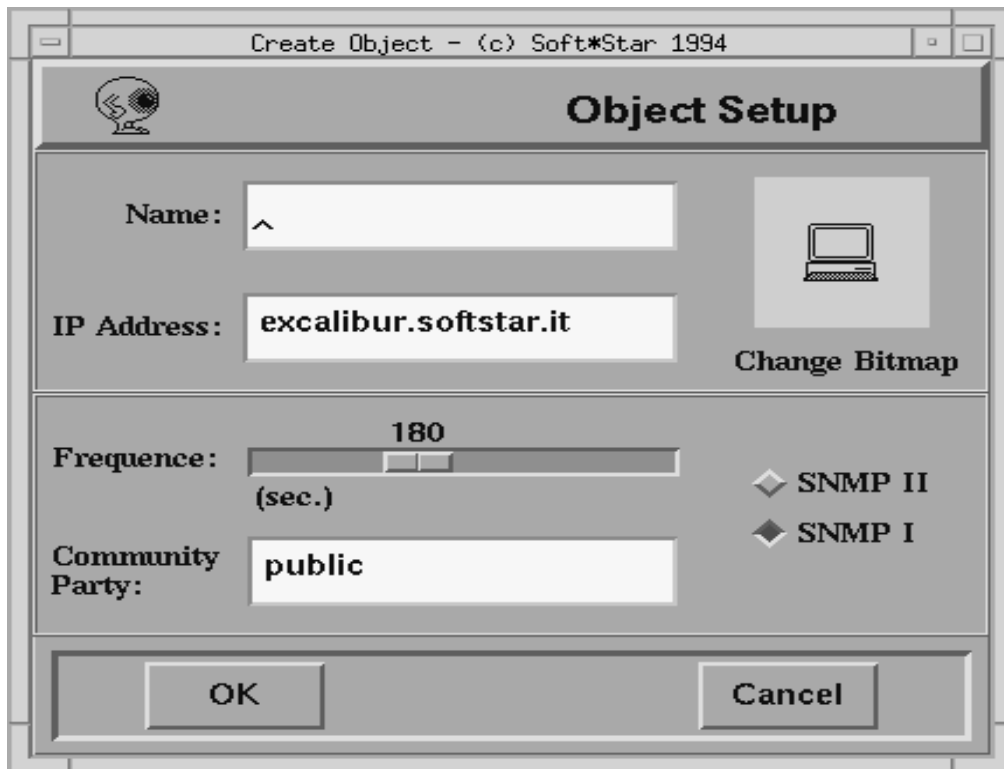


Figure 3.3: Create a physical network object

## 3.2 Objects Pop Up Menu.

Each object, both physical and logical, shown in a map has an associated Pop-Up Menu. It is activated by placing the mouse pointer over the object and pressing the **Menu** mouse button. All the operations that can be carried out on that object will be listed in the Pop-Up Menu and are activated by releasing the mouse button on the selected one. Figure 3.4 shows a map with an activated Pop-Up Menu on a physical object. Logical objects permit a limited set of operations

**Modify** to change characteristics of the Group. Selecting this option will pop up the same dialog used while creating the group.

**Interfaces** to list all interfaces of that Group.

**Ungroup** to destroy the Group but retaining all the objects it contains.

**Delete All** to destroy the Group and all the objects it holds.

The actions that can be carried out on physical objects are:

**Modify** to change characteristics of the Object. This button will pop up the same dialog described in section *Creating Physical Network Objects*.

**Interfaces** to list, add or remove an interface.

**Delete** to remove the object from the map.

**Telnet** to start an `xterm` with a *Telnet* session to that network node.

**Show** it will query the object for all the `mib-2.system` subtree and visualize the returned information in a readonly dialog box as shown in figure 3.5.

**Query** it enables you to query one or more SNMP variables with a `SNMP GET`, `SNMP GETNEXT` operations or set a single variable with a `SNMP SET` operation.

**Graph** this action is used to select one or more SNMP variable to continuously poll and display graphically.

**Actions** to handle the state transitions of that object.

**Clip** to remove that object from the map without destroying it. The object will be temporary placed in the *Clipboard* for future use.

The *Show*, *Query* and *Graph* actions can obviously only be used with an object that supports SNMP and are automatically disabled if the object is not capable of handling SNMP requests. No operations are permitted on the connections between objects because they are automatically maintained by *NetEye*.

## 3.3 Exploding a Group.

To see the contents of a Group just click with the **Select** button on it; a new map will be appear showing the internal structure of the Group. To close a Map select the **Close** option of the **File** button in it's Menu Bar. Figure 3.6 illustrates how you can create a hierarchical view of your network by grouping logically related objects and then exploding them in separate maps.

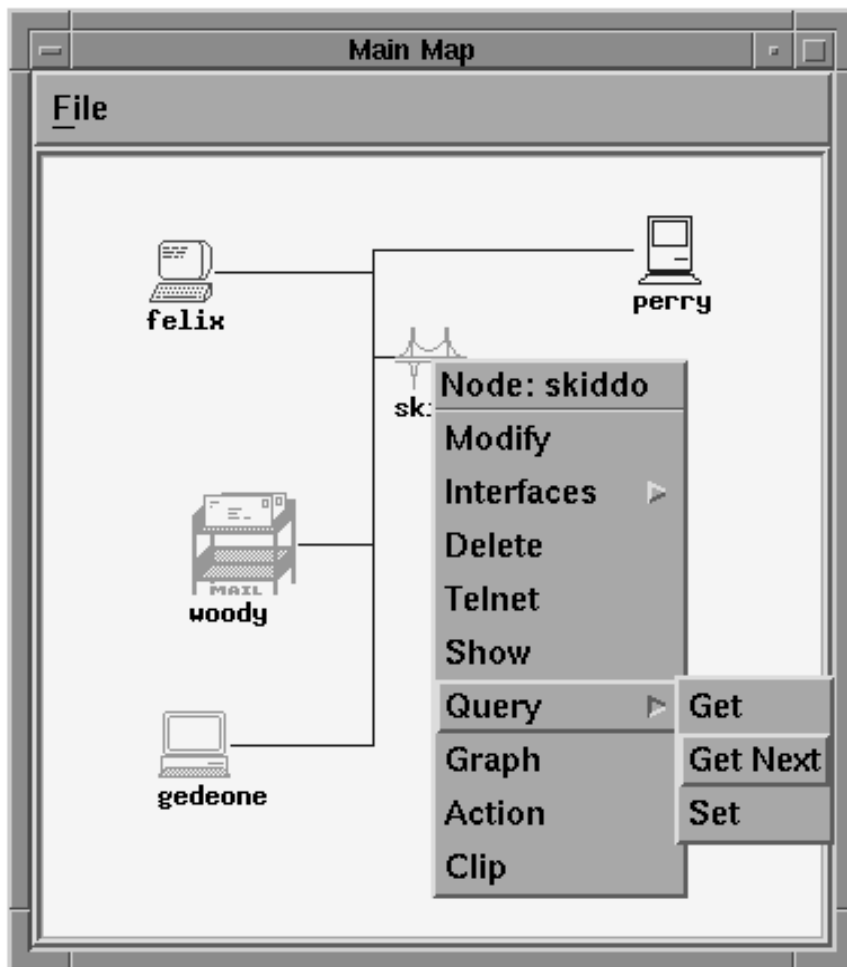


Figure 3.4: A Map with an Object Pop Up Menu

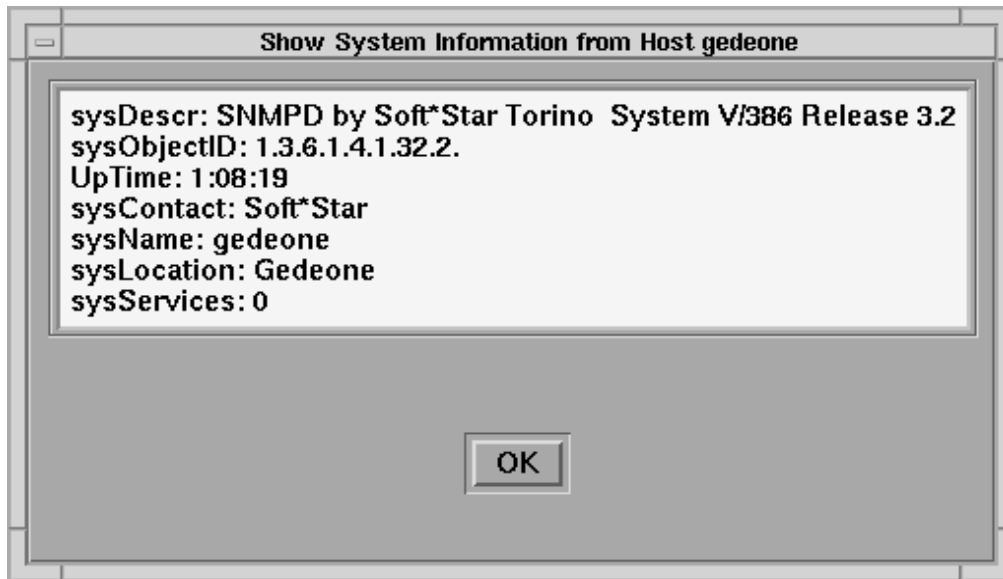


Figure 3.5: Output of a Show action on an object

### 3.4 Deleting objects.

The fastest way to delete a single object is to click the **Custom** mouse button on that object and selecting the **Delete** operation from the Pop-Up menu. You can also perform multiple deletions with these steps

1. click on **Edit** on the *Control Panel* menu bar.
2. from the pulldown menu select **Map** and **Delete** from the associated cascade button.
3. move to the map where you wish to delete the objects and select an area by holding down the **Select** button of the mouse.
4. when you release the mouse button all the objects in the delimited area will be highlighted to denote that they have been selected for deletion.
5. confirm the operation on the dialog that is popped up.

Multiple deletes can also be performed with the same sequence of steps by referring to the menu bar in the map and activating **Edit**, **Map**, **Delete** and then selecting an area on the map.

### 3.5 Moving objects.

Any object can be moved inside it's map. Just press the **Custom** mouse button while pointing at the object you want to move; now, without releasing the button, drag the object to the new position and release the mouse button. All the connections will be automatically adjusted.

If you want to move an object between maps you have to use the *Clipboard* as a temporary storage area for that object. The operations you must perform are:

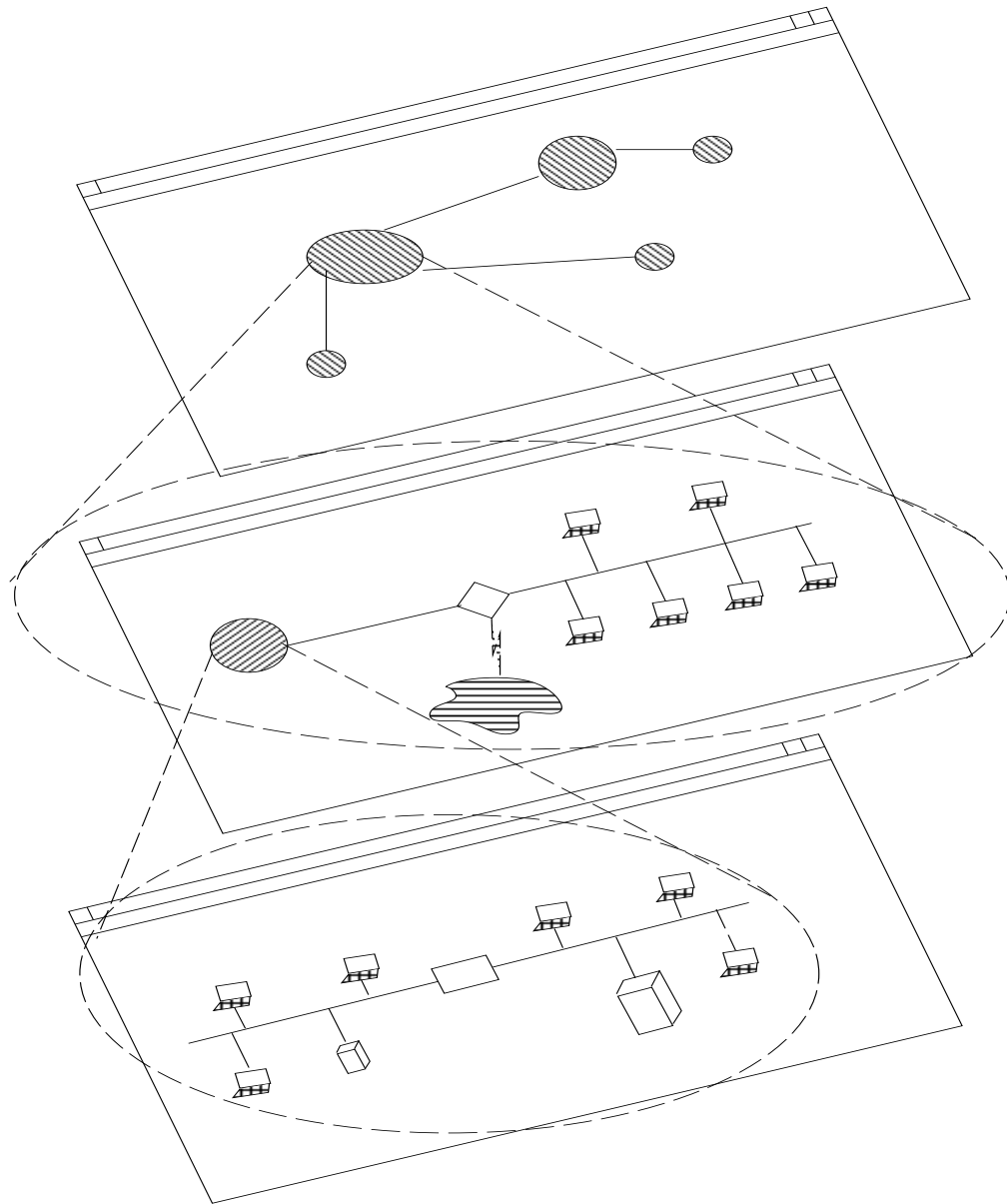


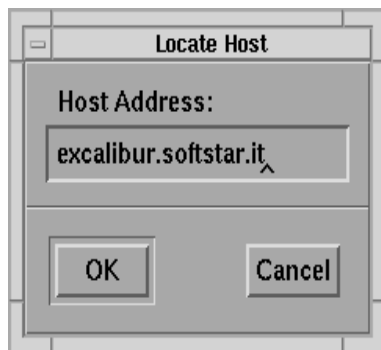
Figure 3.6: Hierarchical view and explosion of Groups.

1. click with the **Menu** mouse button on the desired object, select **Clip** and confirm. The object will be removed from the map and placed in the *Clipboard*. All the connections will be adjusted.
2. select the object from the *Clipboard* and place it inside the new map.

More than one object can be moved from one map to another with a similar procedure. Activate **Edit, Map, Clip** from the menu bar on the map or on the *Control Panel*, then select an area on the map. All the contained objects will be highlighted; when you confirm the dialog, the objects will be moved to the *Clipboard*.

### 3.6 Locating Objects.

Sometimes you may need to locate an object that is deep down a hierarchy and you don't want to explode all the Groups or maybe you forgot to which Group you assigned the object. You can easily locate any physical object by activating the **Network** button in the Control Panel menu bar and selecting the **Locate** action.



You will be prompted with the dialog box shown in figure 3.7; all you need to enter is the IP address of the object you are seeking. After you dismiss the dialog, if the object exits the map containing it will be automatically opened. You can enter either a symbolic name, that will be resolved using a name server, or the actual IP address.

### 3.7 Adding objects to a Group.

After you have defined a **Group** or Logical Network Object, you can add objects to it in two ways;

- select one or more objects on the Clipboard and then click the mouse **Select** button on the Group to which you want to add them.
- drag an object over the Group with the **Custom** button as if you were moving it into the Group.

### 3.8 Icons

All icons are in standard *X Window System* format and you can create your own using **bitmap**, the standard bitmap editor supplied with your implementation of X. After creating a new icon you must place it in the icon directory used by *NetEye*. For more information refer to *NetEye Installation and Configuration*.

### 3.9 Backgrounds.

The background of all maps can be made visible or invisible by activating the button labeled **Edit** on the Control Panel and choosing the **Backgrounds** option. You will then be prompted with a **Show** or **Hide** selection; choose the appropriate one. The use of a background can be activated or deactivated on a single **Network Map** by doing the same set of operations on that map's menus bar. When you exit *NetEye*, the current configuration of the backgrounds of each map will be save in order to restore it in the same way the next time you run *NetEye*.

The format of the backgrounds is *GIF*; in this way you can create new backgrounds using a scanner or any drawing tool capable of generating this format. Add all the backgrounds to the standard *NetEye* background directory. For more information refer to *NetEye Installation and Configuration*. Please keep in mind that adding colorfull backgrounds to your maps uses resources, and this can lead in slow startups and shutdowns especially if your machine is not very powerfull.

## Chapter 4

# Control Panel.

The **Control Panel** is a very important tool because it permits you to centrally control most of *NetEye*'s functionalities. It is shown in figure 4.1 and can be divided in three areas:

**Menu bar** used to activate other tools or perform editing and customizing jobs.

**Status line** that shows *NetEye*'s current status and activities.

**Counters** that summarize in a numeric fashion complex information collected from the other tools.

### 4.1 The Menu Bar.

The Menu Bar on the Control Panel contains the following buttons

**File** the pull down menu associated has only one button, **Quit** to exit *NetEye*. The confirmation dialog shown in figure 4.2 will be popped up. Before exiting *NetEye* will save the state of all your maps in order to automatically show them with the same the next time you start *NetEye*. The exact position of each map on the screen cannot be saved because it is controlled by the window manager.

**Edit** from the associated pulldown you can perform general editing functions such as creating or deleting objects and maps, resetting counters, showing or hiding backgrounds on the maps, setting actions on state transitions of individual objects.

**Network** to perform operations on the network, such as snmp **set** or **get** operations, start the exploration of a new network, dynamically plotting of variables, instruct the network explorer to analyse a specific network.

**Tools** to activate other tools such as the *Trouble Ticketing Subsystem*, the *SNMP Trap Query Facility* or the *Configuration Tool*.

**Help** to activate the interactive help facility, the color legend or get information about your *NetEye* license.

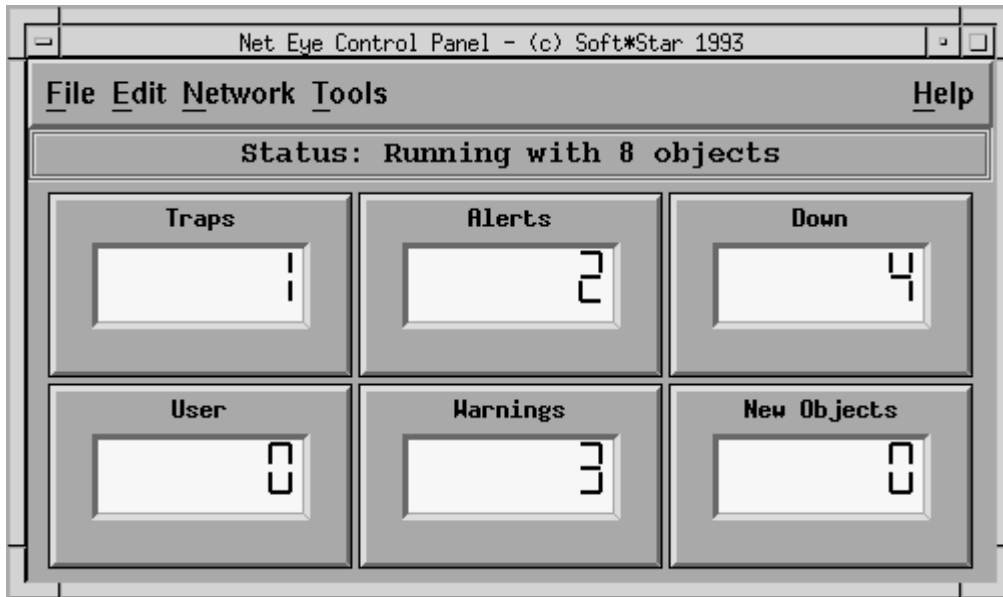


Figure 4.1: Control Panel



Figure 4.2: Confirmation dialog

## 4.2 Status Line.

The Status line shows *NetEye*'s current state or which operations are being carried on. When you are not interacting with *NetEye* the status line reports the number of physical objects that are been supervised; other information reported is

- Create Object
- Create Group
- Destroy Group
- Selected from ClipBoard
- Selecting Hosts
- Move Host
- Move Group
- Modify Object
- Delete Object
- Destroy Group
- Destroy Group and Objects
- Browse Mib
- Running
- Exit from Net Eye

## 4.3 Counters.

Using the color codings in the Maps and the Counters in the *Control Panel* you can have an immediate feeling of how your network is performing and what kind of events have occurred. Each Counter works interactively with the other tools and is incremented on specific events, some of which can be defined by you. The Counters in the Control Panel are:

**New** how many new objects have been detected by the *Autodiscovery* feature and are currently in the *New Objects Clipboard*. This is also the number of objects that are not yet under *NetEye*'s control.

**Down** the number of Objects, known by *NetEye*, that are currently inactive. This means they do not respond to ICMP or SNMP messages; probably they are disconnected from the network or powered off.

**Alert** how many user defined *Alert Events* have occurred.

**Warning** how many user defined *Warning Events* have occurred.

**Traps** the total number of *SNMP Traps* received by *NetEye*.

**Alarms** the number of user selected *SNMP Traps* that have occurred.

### 4.3.1 Resetting Counters.

When you are tracking a specific problem it may be useful to reset one or more counters in order to see how many events occur in a certain time interval. You can reset all the Counters at once or selectively one at a time by activating the **Edit** button in the Control Panel and selecting **Counters**. Select the Counter you want to reset from the cascade menu that appears.

## 4.4 Using Interactive Help.

*NetEye* includes an interactive help facility that you can consult any time you need more information on a particular subject. To activate this facility press the **Help** button on the *Control Panel* and select **NetEye**. A window, as shown in figure 4.3, will be popped up with some introductory text. All the subjects for which some help is available are highlighted using a different foreground color, that is controlled by the X resource `highlightColor`. To set the foreground text to red add to your `.Xdefaults` a line containing `*ControlPanel*hman*highlightColor:red`. The shape of the cursor will change to a small hand when you place the cursor over the highlighted text; this means that if you click on that text you may obtain more information on that subject. Every time you click on a new subject, *NetEye* records your request so that you can move back and forth. To move between pages that you have already visited use the buttons under the text area

**First** to return to the top of the Help system.

**Previous** to go back one page.

**Next** to go forward one page.

Use the **Load** button to dynamically add help pages to the browser. The interactive help facility can be easily customized to include your own help information; for more information refer to *NetEye Customization*.

## 4.5 NetEye version information.

The version of *NetEye* and other licensing information can be viewed by selecting **Version** from the **Help** pull down menu in the *Control Panel*. A small dialog, as in figure 4.4, will be posted; it contains

- version number
- a copyright notice
- the name of the licensee

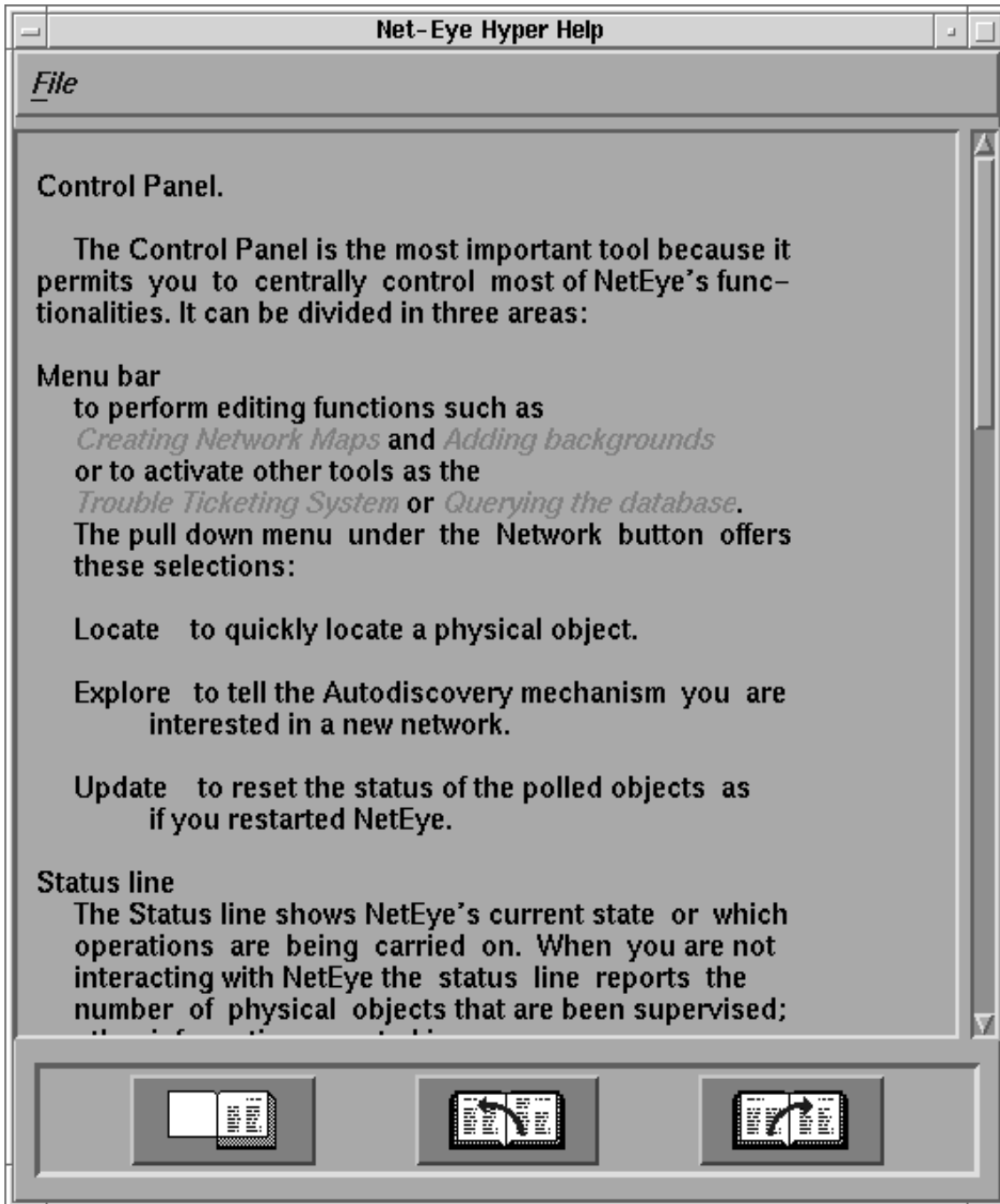


Figure 4.3: Interactive Help facility



Figure 4.4: Version and licensing information

- the expiration date of the license

Most of the information is contained in `$NETEYEDIR/etc/neteye.lic` that also contains the activation key.

## 4.6 The Clipboard.

As you have already seen the *Clipboard* is where the *Autodiscovery* tool places all the new objects as it locates; figure 4.5 shows the *Clipboard* with some new objects. The *Clipboard* contains two areas:

**Networks area** containing the list of networks currently known to *NetEye* but still unexplored by the *Autodiscovery* facility.

**Object area** containing all the objects currently discovered. They appear as buttons labeled with their official name as returned by the local name server.

From the *Clipboard* you can also control the network *Autodiscovery* tool by starting or stopping it by clicking on the toggle button labeled **Network Explorer**.

### 4.6.1 Basic use of the Clipboard.

As objects on your network are located by *NetEye*, you will want to move them from the *Clipboard* to one of your maps. To accomplish this you must perform these steps:

1. select one or more objects by clicking on the buttons, on the right side of the *Clipboard* that represent the objects. This action will change the buttons' color to red, meaning that you have selected that object. If you don't have many objects in the *Clipboard* a global selection is available by clicking on the button labeled **Select All**.

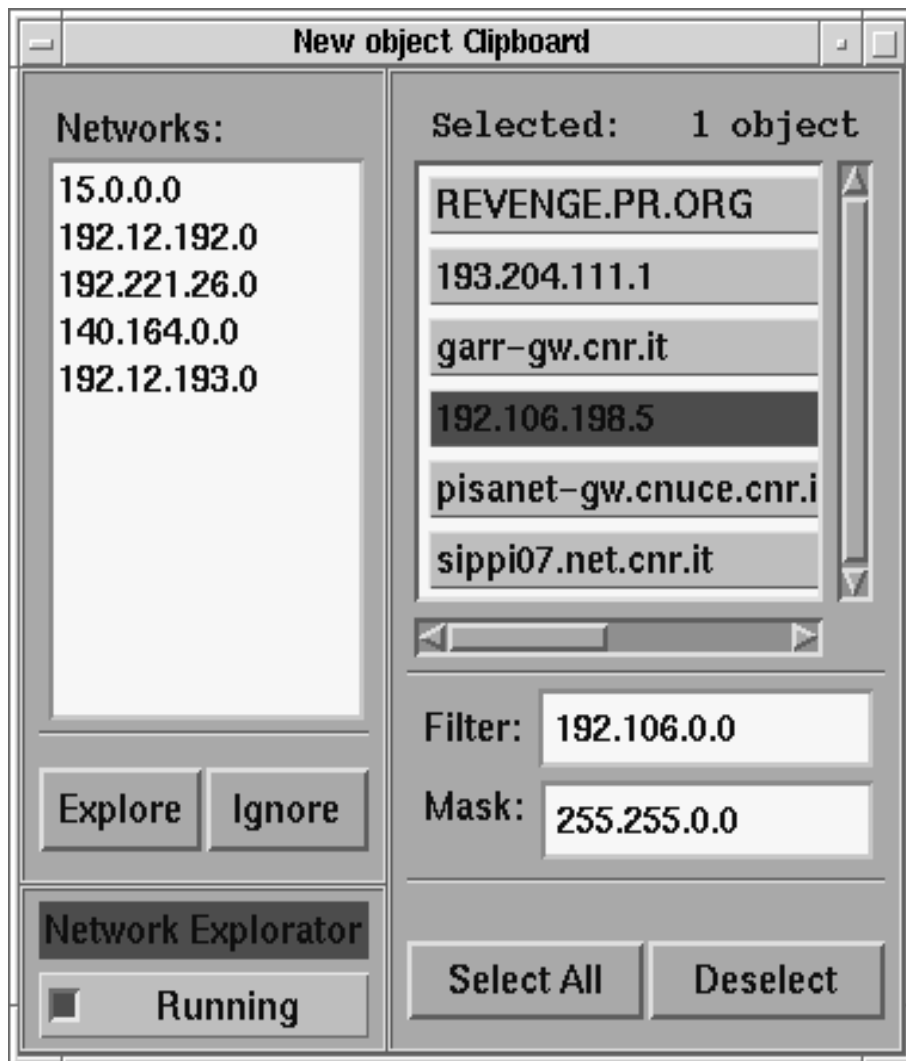


Figure 4.5: The Clipboard

2. if you selected an object, then change your mind and want to deselect it just click on it again. A global deselection is available by clicking on the **Deselect** button.
3. move to the map where you want to insert the selected objects and click the **Select** mouse button. All the objects will be removed from the *Clipboard* and placed in the selected map. Objects can be placed in a map even if it is not currently open; just click on it's icon and all your selected objects will be moved there without opening the map.

#### 4.6.2 Advanced selections.

More complex selections from the *Clipboard* can be performed by using the **Filter** and **Mask** fields in the object area of the *Clipboard*. In the **Filter** field you can specify any range of IP addresses using the standard numeric format, for example 193.42.2.0. In this case, when you press **Select All**, you are asking the *Clipboard* to pick only those network objects that are on network 193.42.2.0. If your networks are subnetted you may use the **Mask** field to specify a **subnetmask** to use with the value specified in **Filter**. By default the **Filter** value is empty and the **Mask** is 255.255.255.0. Both these values are also considered when executing a **Deselect** operation.

As an example consider a class C network with address 193.42.2 that has been divided in four subnets 193.42.2.0, 193.42.2.64, 193.42.2.128 and 193.42.2.192. If you want to select only those objects on subnet 193.42.2.64 (all IP addresses from 193.42.2.64 to 193.42.2.127) you will enter 193.42.2.64 in the **Filter** field and 255.255.255.192 in the **Mask** field.

#### 4.6.3 Handling interconnected networks.

As soon as a new object is located, the *Autodiscovery* facility will query this object for the number of interfaces it has. If the network object has more than one interface (excluding `localhost`) it will be probably connected to more than one network and maybe be acting as a router. *NetEye* will query the device for the IP address of all it's interfaces; these addresses will be displayed in the **Network area** of the *Clipboard*. If you have properly configure the file `/etc/networks` *NetEye* will display the official network name, otherwise only the IP address in numeric format.

As *NetEye* discovers new networks the **Network area** will fill up. You can handle them in two ways:

- Explore a network.
- Ignore a network.

These two operations are activated by first selecting a network from the list then pressing the **Explore** or **Ignore** buttons. Remember that *NetEye* will not start exploring new networks until told. Exploration of a network can also be started from the *Control Panel* menu bar by activating the **Network** button and selecting **Explore**. From the list of known networks select the one you are interested.

#### 4.6.4 The Clipboard as a temporary object store.

As you build your network maps or when the physical layout of your network changes you will need to move one or more objects around the maps. These operations are accomplished by using the *Clipboard* as detailed in the following two steps:

1. cut the objects from the current map and place them in the Clipboard.
2. selectively move the objects from the Clipboard to the new maps.

If you want to move a single object just activate it's associated Popup Menu and select **Clip**, then answer **Yes** to the confirmation dialog. To move more than one object into the Clipboard do the following

1. click on the **Edit** button on the menu bar of the map or of the *Control Panel*.
2. select **Map** then **Clip** from the a cascade menu.
3. move to the map you want to edit and select an area of the map by holding down the **Select** mouse button. When you release the mouse button all the objects in the delimited area will be selected for moving to the Clipboard.
4. on the dialog that appears select **Yes** to confirm the operation.

As you move objects from the maps to the clipboard they will still be tracked by *NetEye* so you won't loose their data.

When you exit *NetEye* and there are still objects in the *Clipboard* these will **not** be saved in the internal database and will be permanently lost until the *Autodiscovery Facility* doesn't detect them again.

#### 4.7 Color codes legend.

Every object on the maps is color coded. You can permanently post a color code legend by clicking on **Help** in the Control Panel menu bar and selecting **Legend**. The legend as in figure 4.6 will be posted; it will remain there untill you deactivate the **Legend** toggle button with the same sequence of actions.



Figure 4.6: Color codes legend

## Chapter 5

# The MIB Browser.

The *MIB-2* standard defines about 170 variables to which we must add all those in the `private.entreprises` subtrees. Remembering their names, types and values is quite a task. To simplify the navigation of the MIB *NetEye* has a **MIB Browser**, shown in figure 5.1, that can be activated by all the tools that require you to input a SNMP variable. On the **MIB Browser** you can identify these areas:

**MIB description** where the optional descriptive comment is held to help you understand the use of that Object Identifier.

**MIB tree** where you move around in the MIB.

**Selection** where the exact name of the selected Object Identifier is shown. The prefix `iso.org.dod.internet.mgmt.mib-2` (1.3.6.1.2.1) is omitted. If you are already skilled with SNMP you can directly type the name of the SNMP variable in the *Selection* area. If you don't specify the instance number of the Object Identifier *NetEye* will supply zero by default. To change this value append the correct instance value before activating any button in the low part of the **MIB Browser**.

**Command buttons** where you select the operations to perform.

The most important part of the **Browser** is the *MIB Tree* area; it is composed of a set of parallel scrolled lists that are used to build the complete identification of a SNMP variable. The list on the left contains the root of all subtrees under `mib-2` and `private.entreprises`.

The browsing operation works from left to right; every time you single click on an element its name will be added to the *Selection* area. If this Object Identifier has an associated comment it will be loaded in the *MIB Description* area. This type of operation can be useful when you are exploring the private MIB of a new device on your network.

When you double click on any element in the *MIB Tree* area that is not a leaf node in the MIB, the corresponding subtree is loaded in the scrolled list immediately to the right. This operation will continue until a leaf node has been reached. As you work your way down the tree, the *Selection* area will contain the fully qualified name of the Object Identifier; the elements of the scrolled list that you have selected will remain highlighted. The global horizontal scroll bar

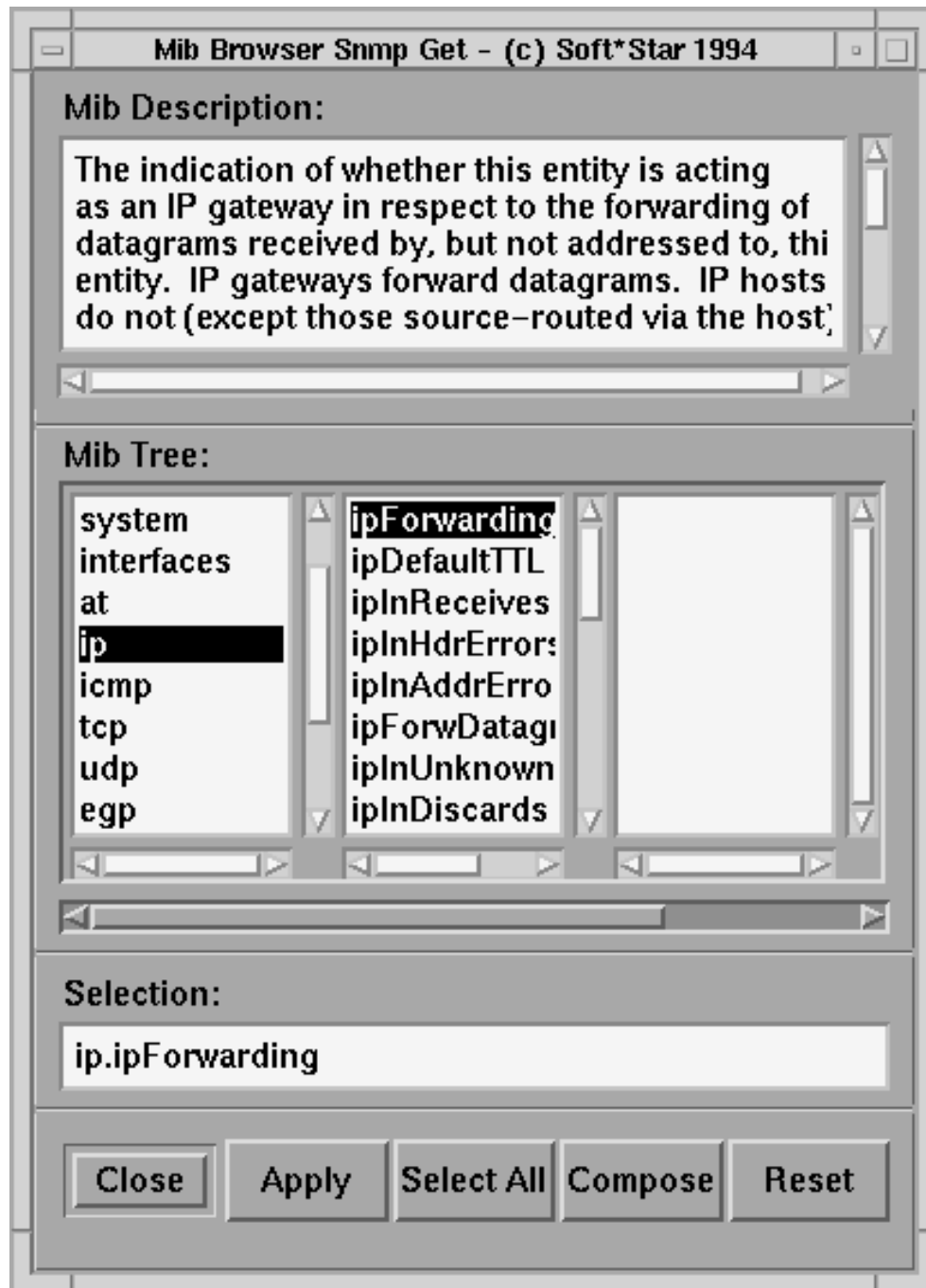


Figure 5.1: MIB Browser

right under the *MIB Tree* area enables you to move around in the various scrolled lists when the variable nesting gets quite deep.

If you choose to enter the Object Identifier directly in the *Selection* area, rather than using the **MIB Tree** scrolled lists, you can either omit the prefix `iso.org.dod.internet.mgmt.mib-2` (1.3.6.1.2.1) and it will be supplied by *NetEye*, or enter the full name of the Object Identifier starting from the root. In this case you must prefix your string with `."` (dot) to tell *NetEye* you are referring to the root of all Object Identifiers. For example, to query the complete **system** subtree you would type `.iso.org.dod.internet.mgmt.mib-2.system` in the *Selection* area.

## 5.1 MIB Browser buttons.

At the bottom of the **MIB Browser** there is a set of five buttons that control how the selected Object Identifiers are used.

**Close** to dismiss the **MIB Browser**.

**Apply** to select the Object Identifier described in the *Selection* area. If it is not a leaf node of the MIB, *NetEye* will perform a **snmp walk** of the entire subtree with that Object Identifier as it's root.

**Select All** select all those Object Identifier that are leaf nodes, all others will be ignored.

**Compose** pop up a **Compose** window that will hold all the Object Identifiers you select. In this way you will be able to request from the snmp agent unrelated Object Identifiers. Using the **Compose** window will enable you to edit the instance identifier associated with each variable. You can perform normal text editing and cut&paste as supported by *OSF XmText* widget.

**Reset** reset the **MIB Browser** to it's initial state.

## 5.2 Loading private MIBs.

When you start *NetEye* it automatically loads all the MIB files, contained in directory **MIBDIR**, as specified by environment variable **MIBFILES** set up at configuration time. For example the following c-shell command `setenv MIBFILES mib.txt:cisco:novell` will tell *NetEye* to automatically load the standard MIB-2 and the mibs from *Cisco* and *Novell*. At any moment you can load a new MIB by pressing the **Tools** button on the Control Panel menus bar and activating **Add Private Mib**. A file selection box will be displayed showing all the mibs available in the directory specified by environment variable **MIBFILES**; select the one you want and it will be loaded in the Browser. This MIB will be maintained for the current *NetEye* session; if you prefer it to be always loaded move it in directory **MIBDIR** and include is in the **MIBFILES** environment variable.

## Chapter 6

# Controlling devices.

Careful monitoring of network performance can identify problems before a serious failure occurs. This means that you must monitor not only network errors or *trap* conditions broadcast from devices in your network, but you should poll them for statistical and performance information. *NetEye* lets you easily poll one or more SNMP variable of any device on your network and plot it dynamically. The polling rate is user definable. The user interface and device polling is performed by process `snmplot`. Since there can be more than one graph displayed at a certain time, all operations on the graphs will be generally performed by first selecting the operation and then the graph on which you want to perform it. To abort the selected operation press the **Menu** mouse button.

### 6.1 Plotting SNMP variables.

To start polling an SNMP variable of a device place the mouse pointer over it's icon, press the **Custom** button and select **Graph**. These actions will pop up the **MIB Browser** described previously. Every time you press the **Apply** button in the **MIB Browser** and an Object Identifier is specified in the **Selection** area of the **MIB Browser** a new graph will be created.

If all the graphs you are requesting are relative to the same host, they will be stacked vertically in the same window with a vertical scroll bar. A new **Graph** window will be created for each graph on a different host. If you want to plot a particular instance of an Object Identifier, remember to specify it in the **Selection** area. From the **MIB Browser** you can request any number of variables to plot.

Without leaving the **MIB Browser** you can start plotting a variable of a new device by simply popping up it's associated popup menu and selecting **Graph**; from now on all the selections made in the **MIB Browser** will be sent to the newly selected host. When you are done defining your graphs, dismiss the **MIB Browser** with the **Close** button.

Shortly after you press the **Apply** button in the **MIB Browser**, a window, as shown in figure 6.1, will be posted on the screen. It is the *Monitor* window and it is divided in several sections depending on the number of SNMP Variables or Graphs you want to plot. Immediately under the menu bar you will find an area common to all graphs containing

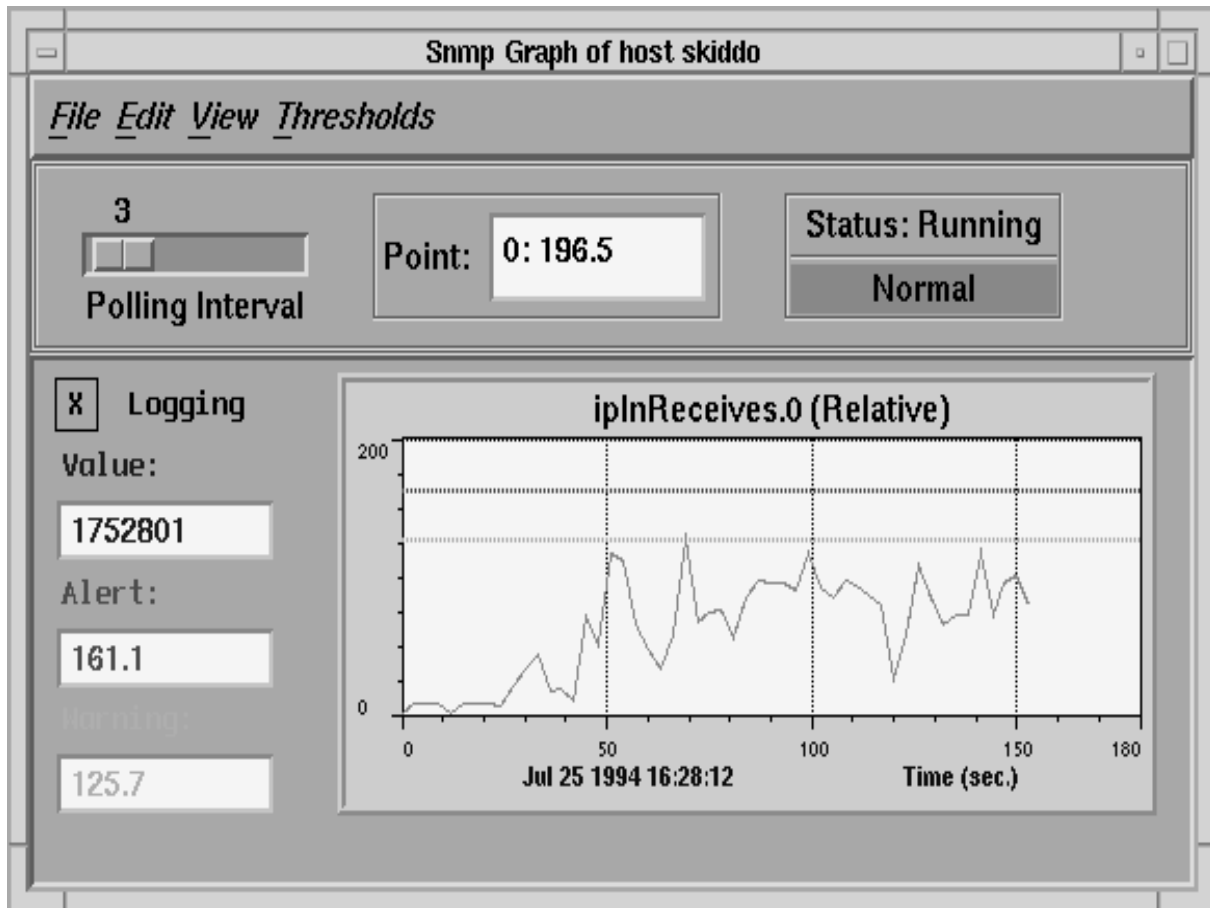


Figure 6.1: Plotting absolute and relative values of SNMP Variables

**Polling Interval slider** you can change the polling period using this slider. When you set a new poll interval it will be used for all the variables of that physical object; this is done in order to maintain the spatial relationship between the various graphs.

**Point** this field will report the X,Y coordinates of the mouse pointer when it enters a graph. This information helps you precisely place alert and warning thresholds as describe further on.

**Status** displays the current state of the *Monitor* window for a certain physical device. Generally the Status is **Running** but it will sometimes contain a hint of your next action, when you are operating on one or more graphs.

On the left of each graph you will find these informations:

**Logging** the box is ticked when you activate the logging function for that garph.

**Value** the absolute value of the SNMP variable being plotted.

**Alert** the value on the Y axis that currently identifies the *Alert* threshold.

**Warning** the value on the Y axis that currently identifies the *Warning* threshold.

On the top of each graph the exact name of the MIB variable is reported and if it is being printed in absolute or relative form. The X axis reports the number of seconds that have passed since the initial polling of that graph. Immediately under the the X axis is recorded the date and time of when the graph was started.

## 6.2 Plotted values.

By default all points displayed in a graph are plotted in *relative* mode and are computed as  $\Delta = value_i - value_{i-1}$ . You can change the plotting to *absolute* mode with these operations:

1. click on the **View** button of the menu bar, from the **Mode** cascade select **Absolute**.
2. move to the graph you wish to modify. The mouse cursor changes into a small pointing hand.
3. click the **Select** mouse button on the graph.
4. to abort the operation press the **Custom** mouse button.

## 6.3 Multiple plots.

If you want to start plotting the same set of SNMP variables on more than one object you can do this by repeating the procedure previously described for each host. There is a short cut and it is done with the following steps

1. click on **Network** on the Control Panel menu bar.
2. select **SNMP** and **Graph** from the cascade menu that follows.
3. move to the map that contains the objects you want to plot. While keeping the **Select** mouse button pressed select an area of the map.

When you release the mouse button all the objects in the delimited area will be highlighted and you will be requested to confirm the operation. A new plot window will be created for each selected object. This operation is obviously available only on SNMP capable devices; if in the area you select there are some devices that don't support SNMP they will be ignored.

## 6.4 Resetting and deleting graphs.

Graphs must be reset all at the same time; if this were not done the spatial relationship between the various graphs would be lost. To reset all graphs that plot snmp variables of a certain host press the **Edit** button of the *Monitor* menu bar and select **Reset**.

When you no longer need a graph press the **Edit** button of the *Monitor* menu bar and select **Delete**. You can destroy all the graphs at once by selecting **All** from the cascade or selectively one at a time. With this last option the mouse pointer will change to a pirate symbol when you pass over each graph; clicking destroys the graph.

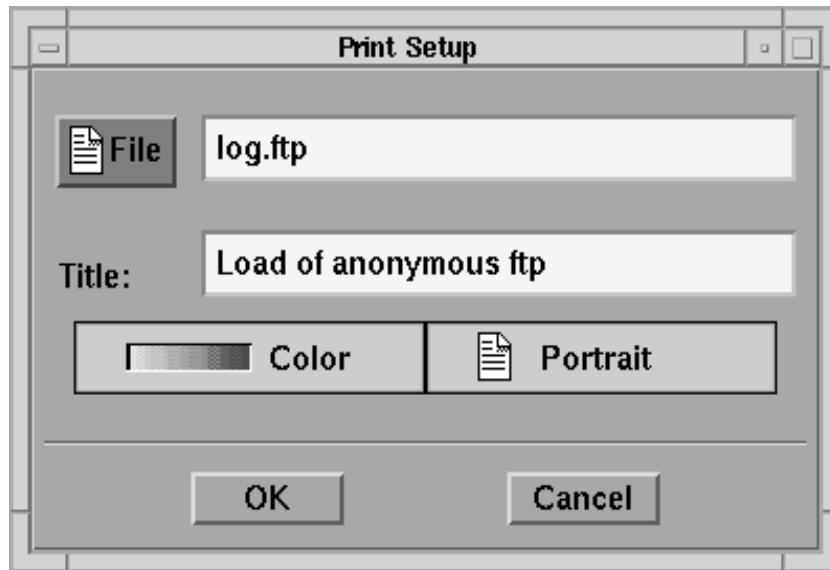


Figure 6.2: Selecting print options.

## 6.5 Printing graphs.

To create a *PostScript* hardcopy of a graph you must press the **Edit** button on the menu bar and select **Print**. A selection form as in figure 6.2 will be activated.

To specify the name of the file in which you want to save the output insert it in the text area to the right of the **File** button. By pressing this button a file selection box will be shown so that you can move around in the filesystem. To add a title to your hardcopy enter it in the **Title** text field. There are two toggle buttons for the selection of portrait or landscape format and color or monochromatic printing.

## 6.6 Zooming.

When you are polling a device you might be interested in having a close look at a small set of polled values. You can zoom a portion of the graph by activating the **View** pull down menu and selecting **Zoom**. You must now choose on which graph you want to execute the zoom; select the graph by clicking the **Select** mouse button. Once you have selected the graph, choose the area to zoom by clicking and dragging the mouse; when you release the mouse button a new window with the zoomed area will appear on the screen. Use the **Print** button to create a hardcopy of the zoomed graph.

## 6.7 Saving plotted data.

If you need to save the values of the plotted SNMP variables for later analysis, click on **File** button and select **Start** from the **Logging** cascade menu. The mouse cursor will change to a pointing hand; click on the graph whose data you wish to save. In the file selection box that

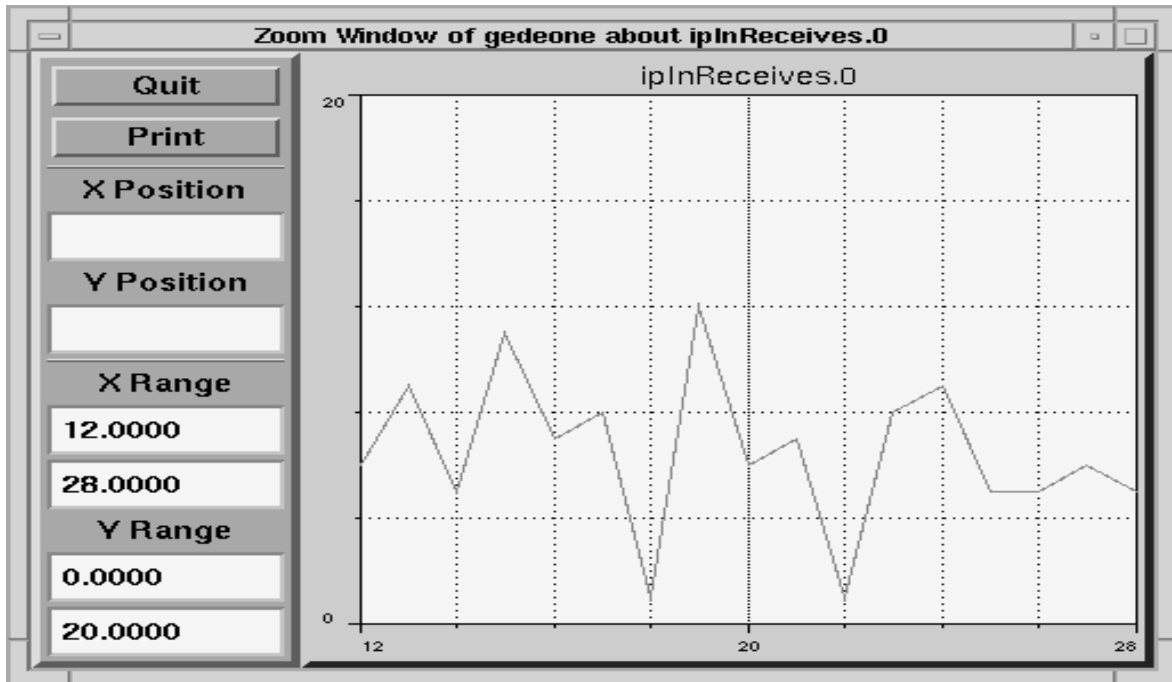


Figure 6.3: Zooming on a graph.

will be displayed enter the name of the file in which the values will be logged. If the file already exists you will be prompted if you want to save the data in append mode or if the file should be truncated. As soon as you dismiss the file selection dialog all data will be saved in the file until you select **File, Logging, Stop** .

The data save in the file consists of a small header that contains the name of the host being polled, the name and instance number of the SNMP variable. Every next line has a touple of values separated by the pipe symbol; currently the saved values are polling time and variable value. An example of saved data is

```
Host: gedeone
Variable: icmp.icmpInMsgs.0
Apr 5 1994 17:58:25|7761
Apr 5 1994 17:58:29|7764
Apr 5 1994 17:58:30|7767
Apr 5 1994 17:58:34|7770
Apr 5 1994 17:58:37|7773
Apr 5 1994 17:58:40|7776
```

## 6.8 Switching representation.

You can dynamically switch from a graph type of representation to a numeric type by clicking on the **View** button of the *Monitor* menu bar, select **Switch** and then click on the graph who's

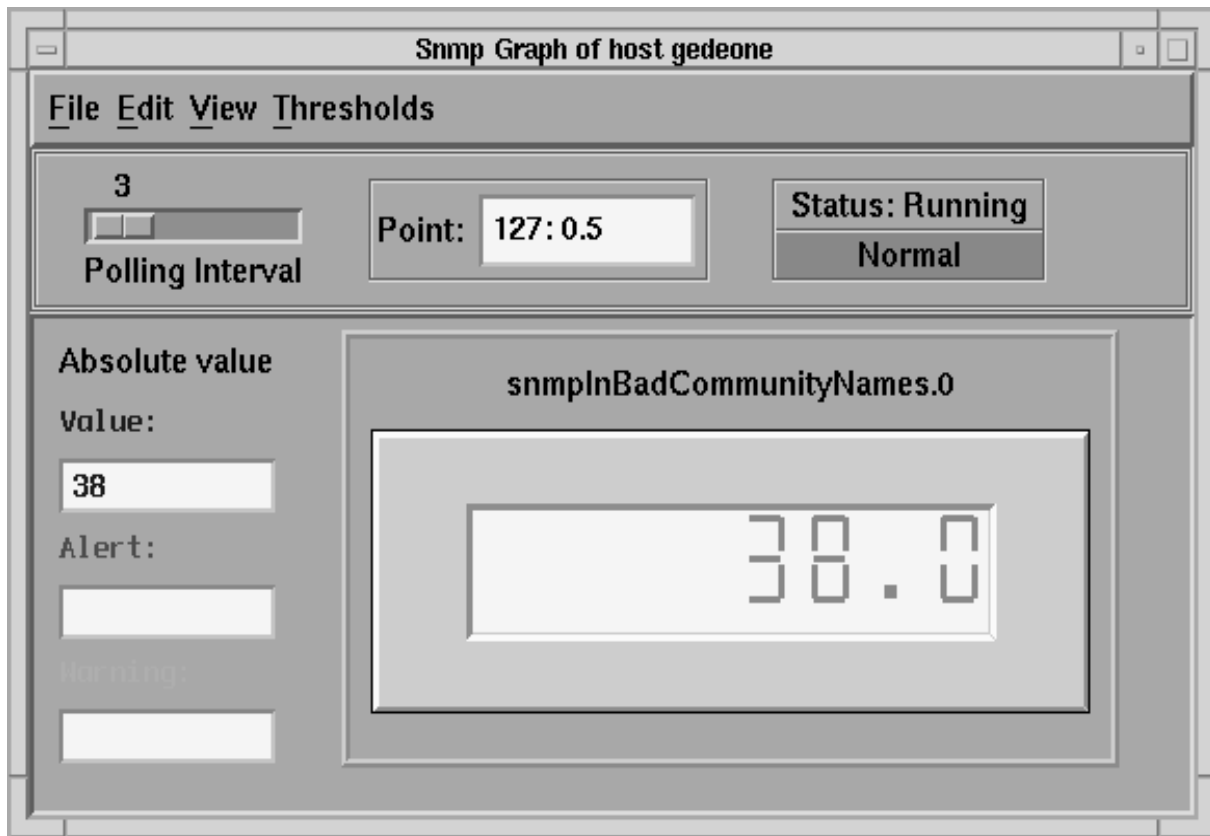


Figure 6.4: Displaying the absolute value of SNMP Variable

form you want to change. The format of the display will change as shown in figure 6.4. Perform the same operations to switch back to the time plot type of representation.

## 6.9 Setting Thresholds.

The scope of continuously polling a device is not just to print a pretty graph but to trigger some event when the value of a polled variable exceeds a certain threshold. *NetEye* defines two types of thresholds

- Warning
- Alert

and they are set by clicking on the **Thresholds** button on the *Monitor* menu bar and selecting **Thresholds**. Now every time you enter one of the graphs in the monitor window the cursor will change to a little hand pointer. As you move the mouse its X,Y coordinates will be reported in the **Point** box so that you have a numeric feedback of where you are placing the threshold. When you have found the threshold value you desire click the **Select** mouse button; a horizontal colored line will be plotted on the graph to indicate the presence of a threshold. From now on

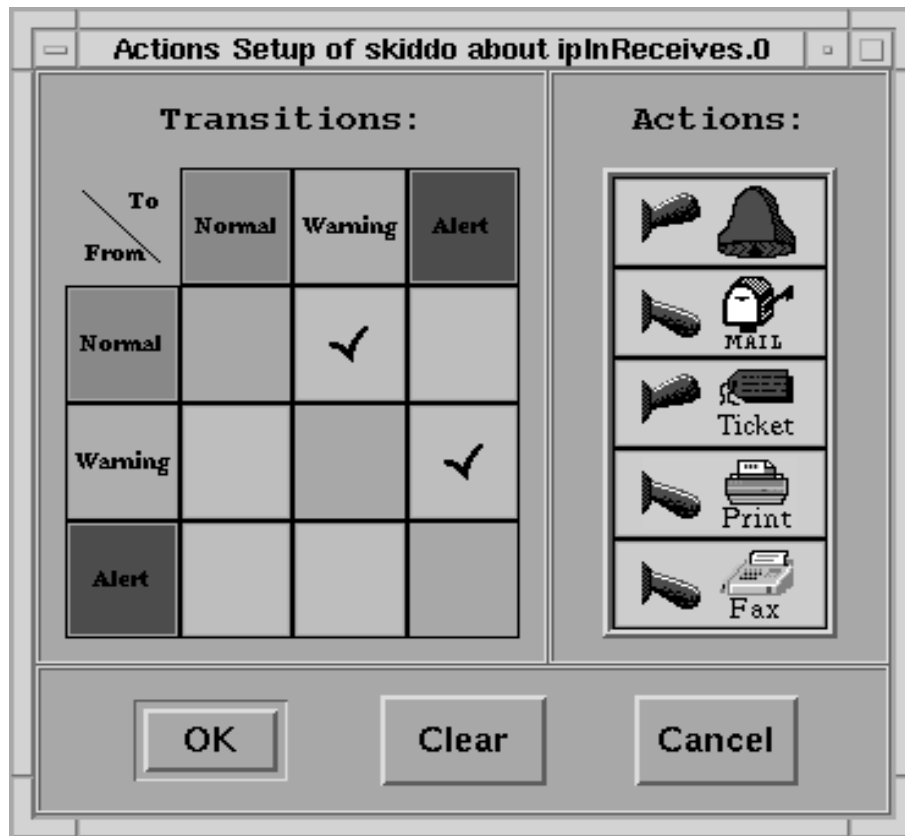


Figure 6.5: Setting actions for a graph.

every time a variable exceeds a threshold, the **Status Box** will report which threshold was passed.

Everytime a threshold is exceeded a message is sent to the Control Panel so that it can increment it's counters. If you want to remove a threshold click on the **Thresholds** button in the menu bar and select **Remove**. From the cascade menu select the type of threshold you want to remove and then click on the specific graph.

## 6.10 Actions.

After defining a threshold you can associate an action to it so that it will be performed everytime the threshold is exceeded. To set an action press the **Thresholds** button on the *Monitor* menu bar, select **Actions** then click on the graph where you want to set an action; the window in figure 6.5 will be popped up. This window has a color coded transition matrix that contains all possible state transitions.

On the right of the transition matrix there is a set of buttons with an iconic representation of the possible actions, that are

**Bell** ring the bell on the computer to attract your attention.

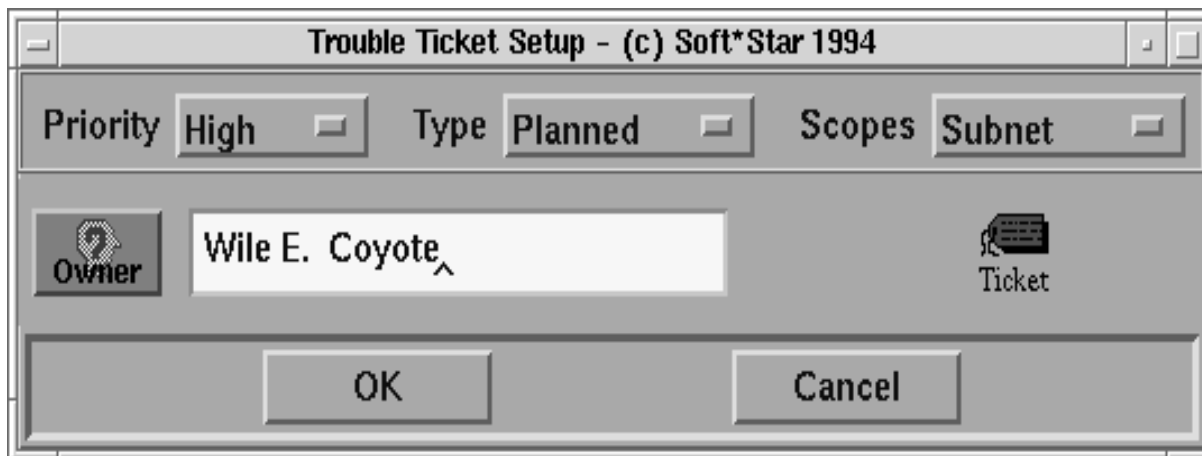


Figure 6.6: Threshold related data for trouble tickets.

**Mail** automatically send a email message to one or more persons with selected data about the event.

**Print** automatically generate a *PostScript* hardcopy of the graph.

**Ticket** automatically generate a *Trouble Ticket*.

On the matrix you select the transitions by clicking on the box at the intersection of the *From* and *To* states; a check will be displayed in the box to show your selection. In the example, the transitions from *Normal* to *Warning* and from *Warning* to *Alert* where selected. Click on the action buttons to define one or more actions to be performed.

When you select the **Mail** action you will be prompted for the email address of the destination party. If you press the **User** button you will be shown the list of all email addresses available in *NetEye's* address book. You can select one of the available email addresses or input a new one; in this case, the address will not be saved in the address book. Be sure that all email addresses are correctly spelled using the format `name@domain` otherwise the mail subsystem will not be able to deliver the message and it will be returned to your personal mailbox. The environment variable `MAILCMD` is used to determine what command must be invoked to send mail; for more information refer to section *Optional environment variables*.

If you select the **Ticket** action you will be prompted with a dialog as in figure 6.6 for the following information:

**Priority** to assign a priority to the generate trouble ticket.

**Scopes** the area affected by this trouble ticket.

**Type** the type of trouble ticket.

**Owner** your name or other identification of who requested the trouble ticket.

By pressing the **Owner** button you will be shown the list of all current persons in *NetEye's* address book. As with the mail addresses you can define the owner on the fly even if it's not in the address book; in this case it will be uses as a temporary reference.

For more information on how to work with trouble tickets please refer to section *Trouble Tickets*.

When you select the print action you will be prompted with the same dialog described in section *Printing graphs* to set up the print format.

## Chapter 7

# Get, GetNext and Set of SNMP variables

Not all variables defined by a snmp agent can be plotted because their value is not numeric, for example a description of the agent or a routing table. SNMP protocol defines three operation that can be performed on any variable regardless to their type or value. These operations are

**Get** to retrieve a specific variable.

**GetNext** to retrieve the variable lexicographically following the named instance.

**Set** to modify the value of the named instance.

From within *NetEye* you can perform any number of **Get** or **GetNext** operations on one or more network objects with an snmp agent. All **Set** operations are limited to only one snmp variable at the time.

### 7.1 Get and GetNext operations.

To execute a **Get** or a **GetNext** operation on a single network object do these steps

1. click the *Menu* mouse button on the object in the map.
2. from the popup menu select **Query**
3. then select the operation you want to perform from the cascade.

These actions will pop up the **MIB Browser**. As you have previously seen the **MIB Browser** can work on these different ways:

1. on a leaf node of a specific SNMP group.
2. on all leaf nodes of a certain SNMP group.
3. on all the subnodes starting from a certain Object Identifier.
4. on a set leaf of different SNMP Groups.

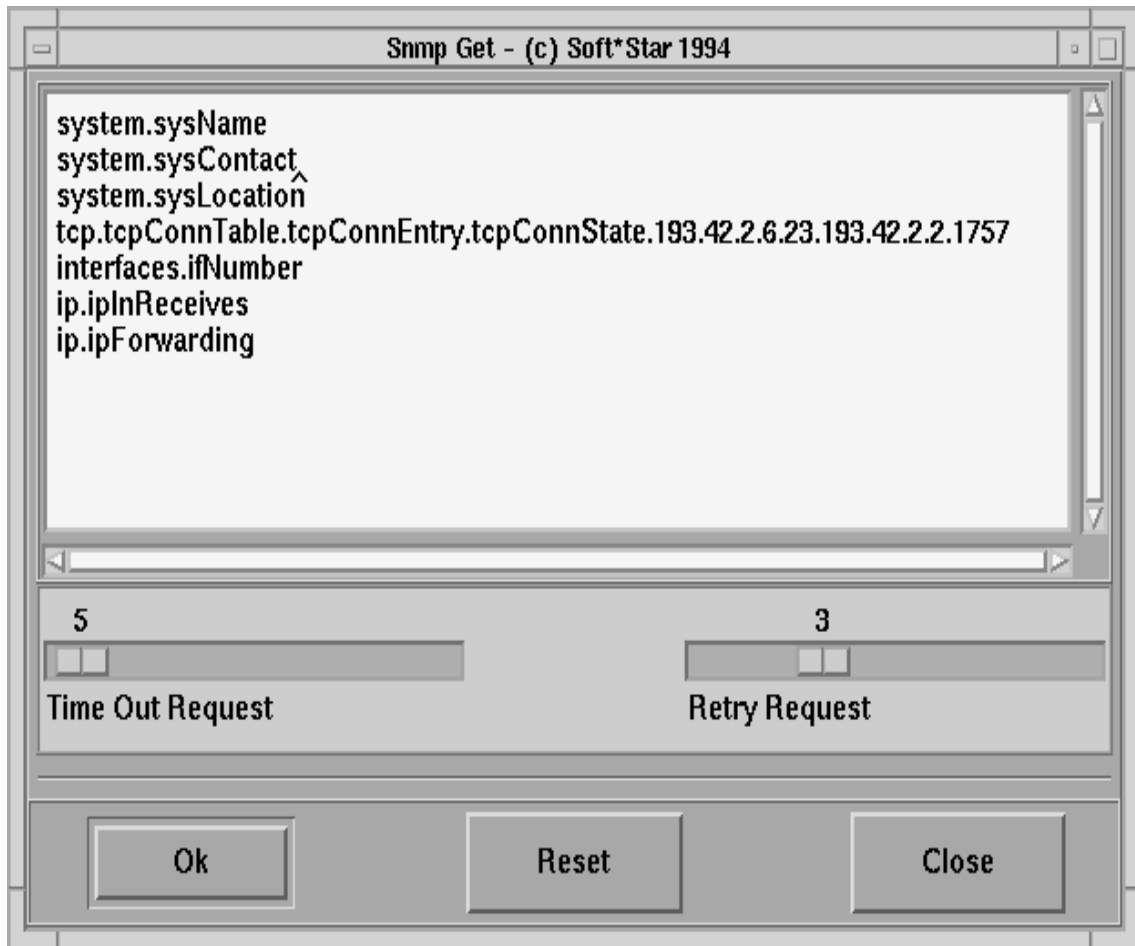


Figure 7.1: SNMP Get, GetNext Compose window

The first mode of operation is obtained by selecting a leaf Object Identifier by double clicking in the in the **MIB tree** area of the **MIB Browser** and pressing the **Apply** button. To query all leaf nodes of a SNMP group double click on it's name and press **Select All**. The third way of operating is to perform an **SNMP walk** of an entire subtree. To do this, select an Object Identifier that isn't a leaf node and press **Apply**. To query different Object Identifier in different SNMP Groups you must first press the **Compose** button in order to pop up the **Compose Window** as shown in figure 7.1.

When using the **Compose** mode you operate normally on the **MIB tree** by double clicking, pressing **Apply** or **Select All** but your requests will not be sent immediately to the network device but added to the **Compose** window. When you are done selecting the variables you need to query press the **OK** button in the **Compose** window. You will notice that this window has two sliders **Time Out Request**, to program the number of seconds before the request times out, and **Retry Request** that represents the number of retries *NetEye* will execute before giving up. These two sliders are usefull when requesting many and complex SNMP variables from a remote device. To clean the contents of the **Compose** window press **Reset**.

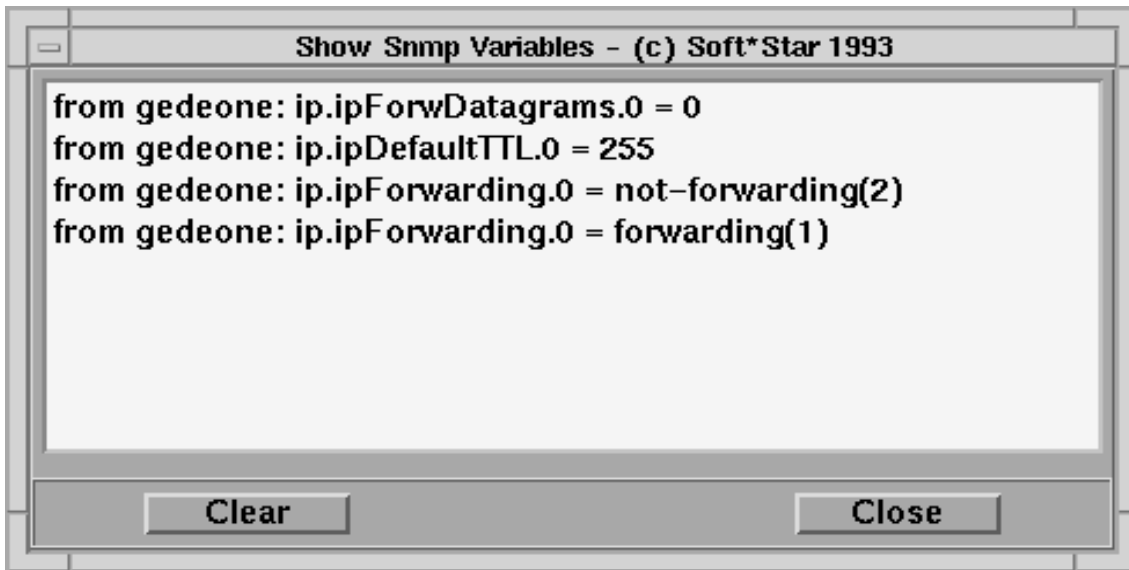


Figure 7.2: result of SNMP Get, GetNext or Set operations

All results of single or multiple Get or GetNext requests will be displayed in a read only window as shown in figure 7.2. You can cut&paste data from this window to the **Compose** window to create complex queries.

If you just purchased a new SNMP capable device you may explore all it's Object Identifier tree by simply entering `.iso` in the *Selection* area and pressing the **Apply** button. Since `.iso` is not a leaf node, *NetEye* will walk all the Object Identifier tree.

Remember that you can refocus your queries from one network device to another, without closing the **MIB Browser**, by simply selecting the appropriate action from it's popup menu.

## 7.2 Set operations.

Set operations are a bit more complex because you must also tell *NetEye* the type of variable and it's value. The steps you must take are

1. click the *Menu* mouse button on the object in the map.
2. from the popup menu select **Query**
3. select **Set** from the cascade.

The **MIB Browser** and the window show in figure 7.3 will be popped up. After selecting the SNMP variable you want to set from the **MIB Browser**, input it's value in the text field just under. There are two row of buttons that you use to set the type of the variable; available type are

- integer
- byte values in hexadecimal.

**Snmp Set - (c) Soft\*Star 1993**

**MIB Variable**

**Mib Value**

Integer     
 Hex bytes     
 Object Id     
 Dot IP Addr  
 String     
 Dec bytes     
 Time Ticks     
 Null Object

**Time Out Request**     
 **Retry Request**

Figure 7.3: SNMP Set operation form

- and Object ID.
- IP address in dot notation.
- a string of characters.
- byte values in decimal.
- time ticks
- null object

You can change the timeout after which the set request fails and the number of retries with the two sliders. To execute the request press **OK**. When the operation terminates the result will be shown in the **Result** window; if the request succeeded the new value is displayed, otherwise the reason of the failure. All the result codes are listed in the window and can be erased by pressing the **Clear** button.

## Chapter 8

# Controlling the state of a device.

Every object that is shown on the maps is color coded in order to show it's state. You can reconfigure the colors associates with the state of each Physical Network Object by changing the value in the *Application defaults* file; the default settings are

Color	State
Blue	Unknown, probably disconnected or powered off
Red	Active and responding to ICMP only
Green	Active and responding to SNMP
Yellow	Active, was previously Green but now responds to ICMP only

If the object is a Logical Network Object it's color is computed as following

Color	State
Blue	all objects are Blue
Green	all objects are Green
Red	there are Blue and Red objects, but no Green ones
Yellow	there are Green and Yellow objects

### 8.1 Actions on state changes.

An object's transitions from one state to another might indicate the existance of some sort of hardware or performance problem. *NetEye* has a tool that can be associated with each physical object so that an action can be performed when a particulare set of state transitions occur. The tool is shown in figure 8.1 and is similar to the one used to associate actions to threshold on SNMP variable plots.

It can be activated only for physical devices by pointing with the mouse on the icon representing the device, pressing the **Menu** button and selecting **Actions**. To select a transition click on the buttons at the intersection of the **From** and **To** in the transition matrix and a check will appear at each selected transition. There is no limitation to the number of transitions you can select. After defining the state transitions you want to monitor, you must choose the action to be performed. To do so you use the set of iconic toggle buttons located on the right side of the transition matrix; the available actions are:

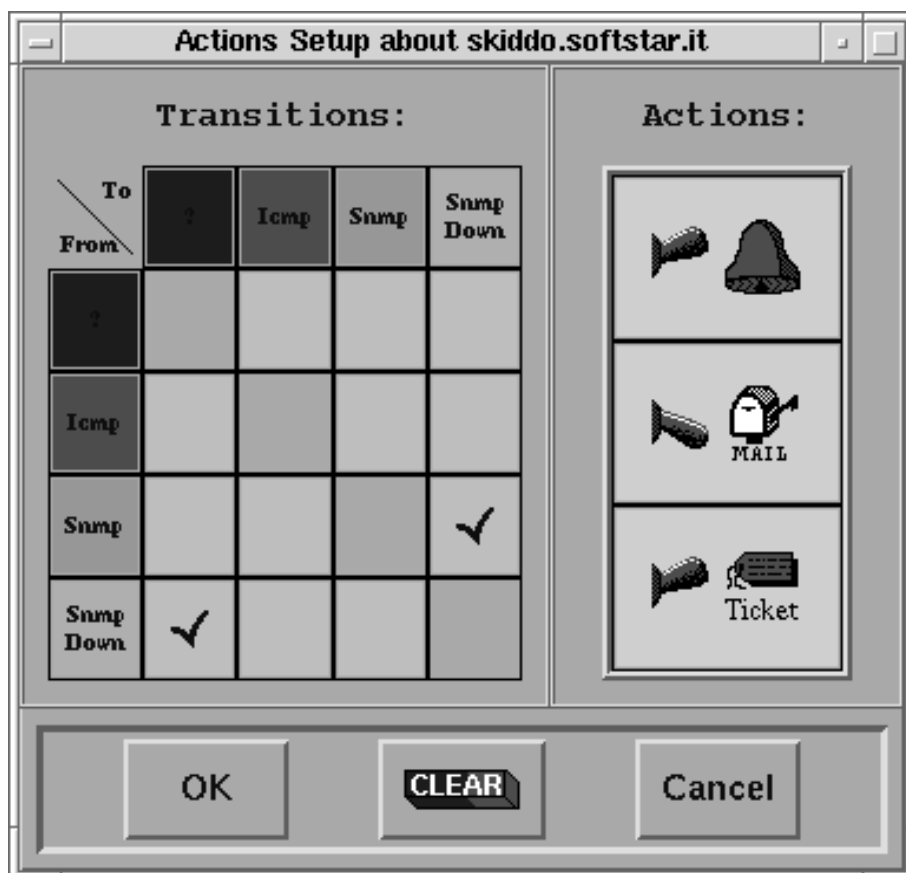


Figure 8.1: Setting actions for object state transitions

**Bell** ring the bell on the computer to attract your attention and automatically open the map that contains the object.

**Mail** automatically send an email message to one or more persons with selected data about the event.

**Ticket** automatically generate a *Trouble Ticket*.

When you select the **Mail** action you will be prompted for the email address of the destination party. If you press the **User** button you will be shown the list of all email addresses available in *NetEye*'s address book. You can select one of the available email addresses or input a new one; in this case, the address will not be saved in the address book. Be sure that all email addresses are correctly spelled using the format `name@domain` otherwise the mail subsystem will not be able to deliver the message and it will be returned to your personal mailbox.

If you select the **Ticket** action you will be prompted with a dialog for the following information:

**Priority** to assign a priority to the generate trouble ticket.

**Scopes** the area affected by this trouble ticket.

**Type** the type of trouble ticket.

**Owner** your name or other identification of who requested the trouble ticket.

By pressing the **Owner** button you will be shown the list of all current persons in *NetEye*'s address book. As with the mail addresses you can define the owner on the fly even if it's not in the address book; in this case it will be used as a temporary reference.

All the actions you define during a work session with *NetEye* will be saved when you exit in order to enable the same actions the next time you start *NetEye*.

## Chapter 9

# Handling SNMP Traps.

Every SNMP capable device will generate a *Trap* when it needs to inform the network management station of some event. SNMP protocol standard defines seven different types of traps that are

- coldStart
- warmStart
- linkDown
- linkUp
- authenticationFailure
- egpNeighborLoss
- enterpriseSpecific

for the specific meaning of each trap or for more information refer to *RFC 1157 A Simple Network Management Protocol (SNMP)*.

*NetEye* handles all the traps with the **Trap Handler** shown in figure 9.1. It's basic function is to simply log in the topmost list window all the traps that are broadcast from the devices on the network. The data that is reported in the window is

- host identification and IP address.
- type of trap; if enterpriseSpecific it's value.
- the time the SNMP Agent of the reporting device has been active.
- an agent specific information; typically system.sysDescr.0

By double clicking on a trap you will activate *NetEye*'s **Locate** function and the map, containing the physical device that generated the selected trap, will be opened. If you want to save the traps in the associate RDBMS must activate the *Logging* feature; to do so click on the **File** button in the menu bar, activate the **Logging** toggle button. To terminate the data logging execute the same steps but deactivate the **Logging** toggle button.

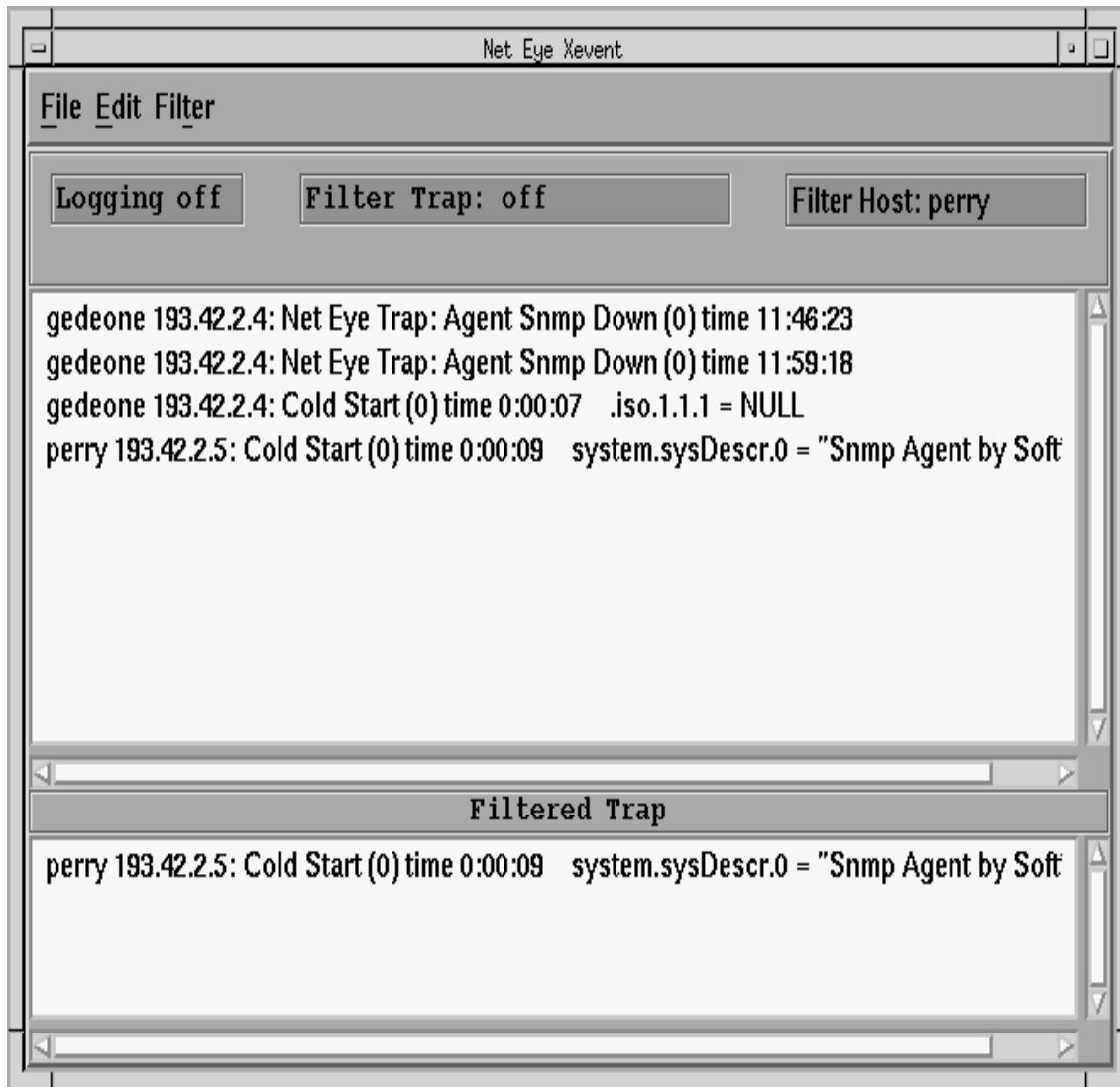


Figure 9.1: SNMP Trap handler

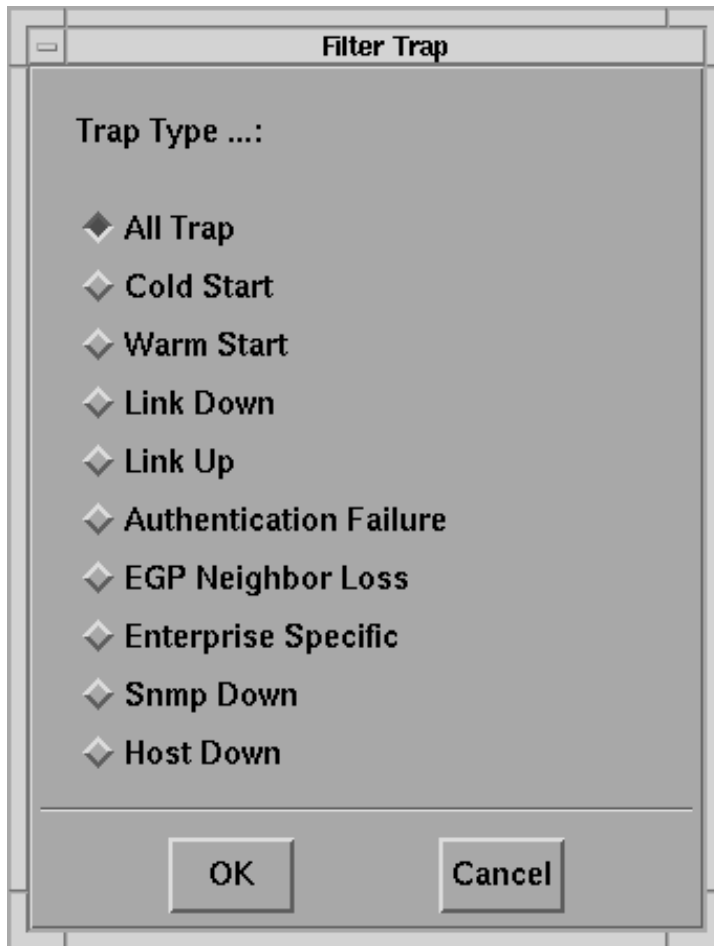


Figure 9.2: Types of SNMP Traps

## 9.1 Filtering traps.

If you want to concentrate yourself on specific traps you can use the filter function that visualizes in the lower scrolled window only the traps that match certain criteria. You can filter by

- trap type
- IP address

To activate the filtering function click on the **Filter** button on the menu bar and select the modality you desire. When you select **Address** you will be shown all the IP addresses currently used by the filter. To add one or more IP address enter them in the **Selection** area; to remove items select them in the **Item List** and press the **Delete** button.

If you want to filter by trap type select **Trap Type** and you will be prompted with a dialog as in figure 9.2. Click on the button according to the type of trap you wish to locate. To stop selecting a particular trap select **All** from the dialog.

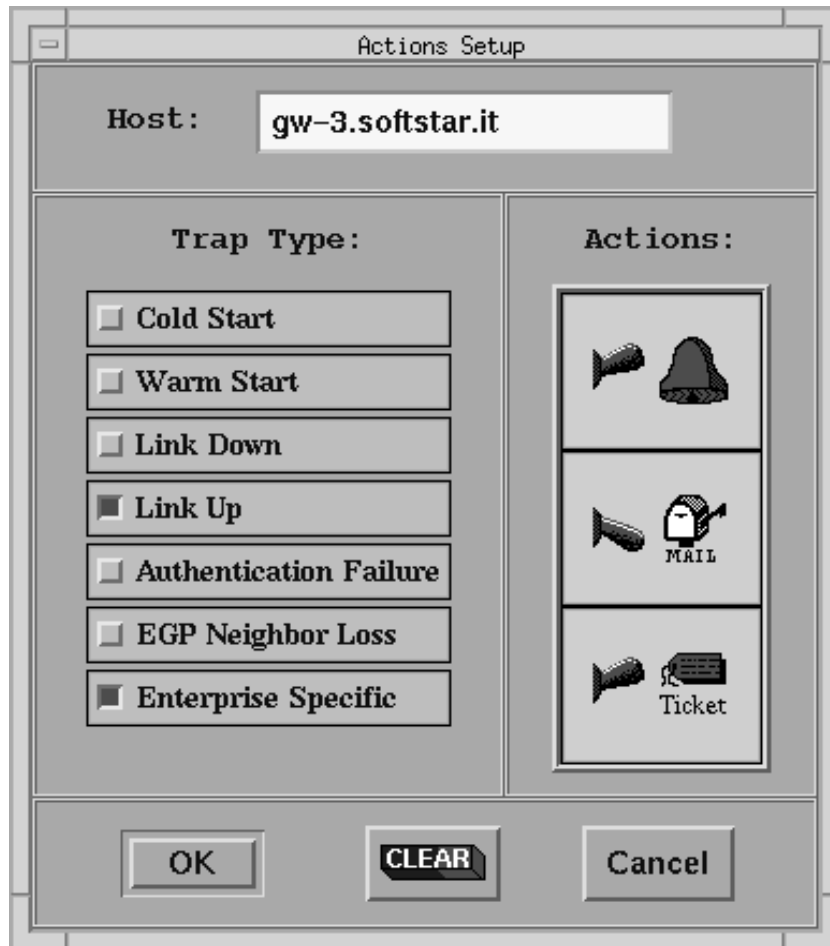


Figure 9.3: Actions on SNMP Traps

## 9.2 SNMP traps and actions.

As with state transitions and thresholds, you can associate an **Action** with SNMP traps. To do so, select the **Actions** option in the **Edit** cascade menu. the window of figure 9.3 will be shown. From this dialog window you can select the trap types and the IP address that will trigger the execution of the specified actions. As before you can select from

- Bell
- Mail
- Trouble ticket

### 9.3 Interrogating the trap database.

As you have seen previously you can save all the trap in the RDBMS associated with *NetEye*. To query the database you must activate the **Trap Query tool** by pressing the **Tools** button on the Control Panel, activate the **Query** cascade menu and select **Traps**; the window in figure 9.4 will be displayed.

You can create complex queries by filling in the form on the screen, in particular you can define

**time interval** by clicking on the **Start time** and **Stop time** buttons that will pop up a dialog as in figure 9.5 from which you can set date and time.

**trap type** click on the **Trap** button and you will be prompted with the same form as in the filter function.

**IP address** click on the **Host** button and you will be asked the IP address of the device that generated the trap.

When you are satisfied with the selection criteria press the **Done** button and all the entries in the database that match your conditions will be shown in the scrolled area; you will also be told the exact number of lines that were matched. To delete some records from the RDBMS define the selection criteria as before, then press **Delete**. All the records that satisfies your conditions will be erase from the RDBMS. To clear the query from press **Clear** and **Exit** to dismiss **Trap Query tool**.

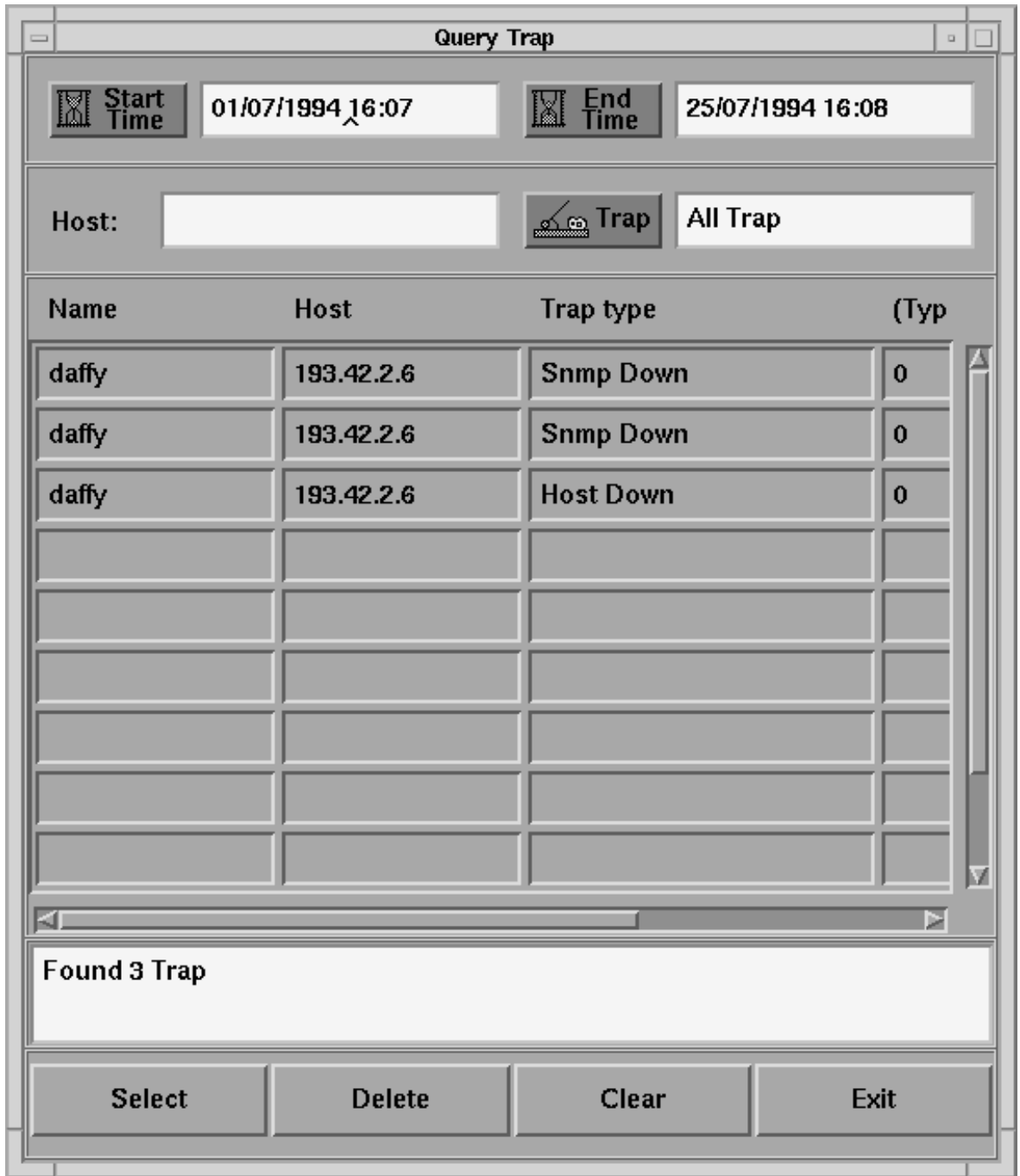


Figure 9.4: Interrogating the RDBMS for saved traps

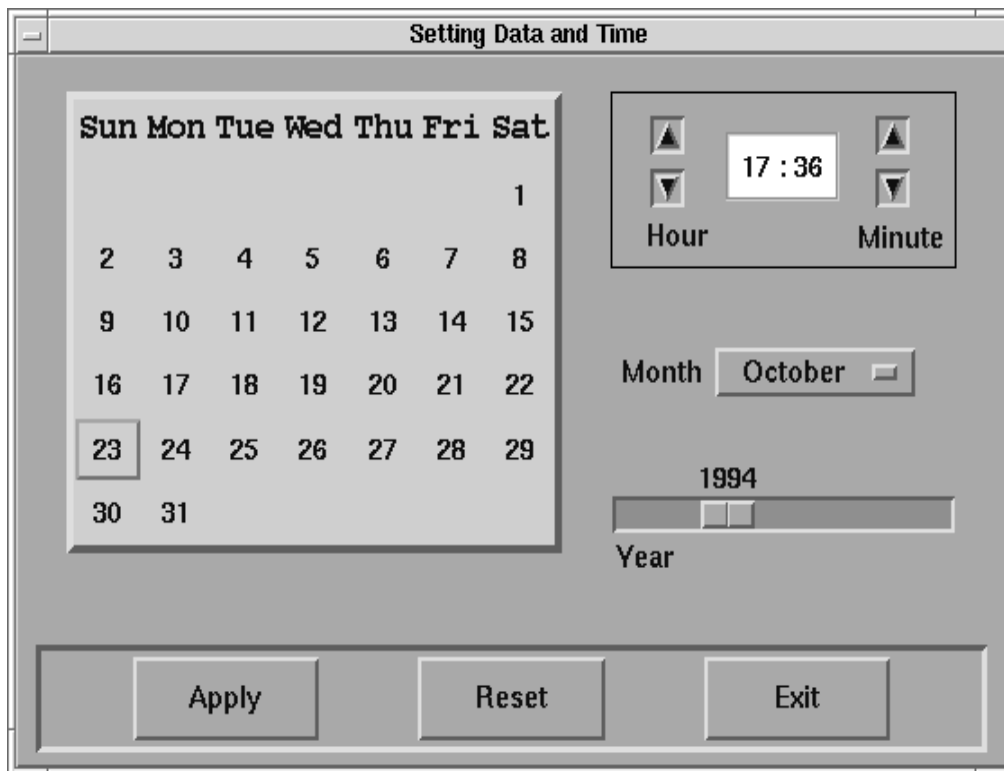


Figure 9.5: Setting date and time for a query

## Chapter 10

# Trouble Tickets.

A **Trouble Ticket** does not have to be related to hard faults such as hardware problems. It is probably more useful when you use a Trouble Ticket to track problems such as excessive use of resources or slow responses on your network. You typically provide the following information:

- The manufacturer or system supplier.
- The contact person's name and phone number.
- An explanation of the problem.
- The actions that you have taken to fix the problem.

To activate the Trouble Ticket subsystem press the **Tools** button on the **Control Panel** menu bar, select **Query** then **Trouble**; the window in figure 10.1 will pop up.

The **Trouble Ticket button box** contains a set of buttons with an iconic representation of their actions; you have these choices:

- Insert
- Select
- Update
- Delete
- List
- Alarm
- Print
- Fax
- Close

To close the **Trouble Ticket** subsystem press the **Close** button.



Figure 10.1: Trouble ticket button box

## 10.1 Structure of a Trouble Ticket.

**Trouble Tickets** are complex objects and contain quite a bit of information. To correctly deal with Trouble Tickets you must understand all the data that is saved in the associated RDBMS.

Every ticket has an associated **Ticket Number** that is generated automatically and that you cannot change in any way; this number will be always shown in the top part of the windows whenever you access a ticket. When a ticket is created it will record the time and date of creation; the date can be automatically entered by *NetEye*.

All tickets are classified by *Priority* (high, normal or low) and by *Type* (planned, unplanned, new site). If you are notified that a site will be down for maintenance that is a *planned ticket*; if the site crashes or goes down for scheduled work but you were not notified that's an *unplanned ticket*.

Tickets can be generated in several ways, manually because you were informed of a problem over the phone or can be automatically created by *NetEye* and it's tools when certain events occur.

The *Scope* of a Ticket indicates who is affected by the problem; usually it is a single host but one or more sites or a complete subnetwork can be affected by the problem.

The evolution in time of the ticket is represented by its *Status* and it can be *open*, *closed*, *hold* or *needinfo*. Normally tickets are either open or closed. However, some tickets get into the state where you are waiting for a response from the original complaint; for these cases you can use the needinfo state. Some times you may decide that a problem will not be address yet and you don't need to allocate it on the list of the open problems; for this situation you can assign a hold status.

When you have solved a problem or it no longer exists you will generally *close* the ticket and assign a close code that indicates what has been the problem. When closing a ticket you should remember to assign the time and date you solved the problem.

## 10.2 Creating tickets.

When you define a new ticket all the information describe previously, except the *Close Code* and *Close Date* will have to be entered in the ticket database. You will input the data using the form activated by pressing the **Insert** button in the **Trouble Ticket button box**. The form is shown in figure 10.2; most of the selections are through option menus.

To describe the problem you are handling press the **Problems** button to visualize the **Problem Dialog** box shown in figura 10.3.

The window contains a editable **Combobox**, a couple of buttons to add and remove problems, a list of all the problems you have selected up to now and the buttons to accept your selections or to dismiss the dialog. You can either select some pre-edited problems from the Combobox or enter a new problem by hand in the editable area of the Combobox. Every time you add a new problem description or modify an existing one, you will be asked if the problem you typed will be permanently saved in the database so it will be available during all future sessions.

Create Trouble Ticket - (c) Soft\*Star 1994

Ticket Number

Priority:  Status:

Source:  Type:

Scopes:  Close Code:

Site  Owner

Opened  Closed

Problems:

Lightning struck microwave link tower

Figure 10.2: Creating a new Trouble Ticket

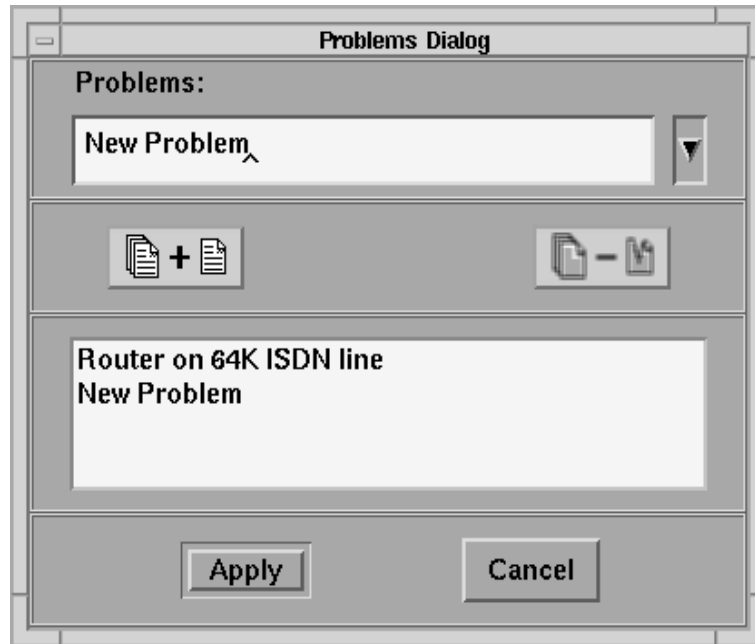


Figure 10.3: Problem Dialog

### 10.3 Searching for tickets.

You have several ways for searching in the Trouble ticket database, each one giving you a different grain of detail and they are

**Select** you get complete control on the conditions used to query the database.

**Alarm** it will extract all the tickets with **Priority = High** and **Status = Open**.

**List** will list all open tickets.

When you use the *Select* button you will be prompted with the form in figure 10.4. If you know the ticket number you are looking for just enter that in the *Ticket Numebr* field and press *Ok*. In case you are interested in a set of tickets just select the appropriate conditions using the option menus; a *don't care* condition is set using the <all> label. If you don't qualify the detail, the search the tool will extract all the tickets currently in the database.

With the **List** and **Alarm** buttons you will not have to describe the search condition because it is fixed as described previously.

When all the data is extracted from th RDBMS it will be show in a window as in figure 10.5. If you click on a line of data you will obtain a detailed view of the selected line in a window similar to the one used for the **Select** operation; all the text fields will be read-only and the options menus insensitive.

Select Trouble Ticket - (c) Soft\*Star 1994

Ticket Number

Priority:  Status:

Source:  Type:

Scopes:  Close Code:

Site  Owner

Opened  Closed

OK Cancel

Figure 10.4: Trouble ticket query preparation dialog

Number	Owner	Site	Open Date
21	FERRU IGNAZIO	skiddo.softstar.it	12/10/1994 16:37
22	FERRU IGNAZIO	skiddo.softstar.it	12/10/1994 16:37
23	PENENGO MAR	iunet-gw.iunet.it	19/09/1994 09:56
24	PENENGO MAR	iunet-gw.iunet.it	18/10/1994 10:00
25	NetEye NMS	woody.softstar.it	19/10/1994 12:34
26	NetEye NMS	woody.softstar.it	19/10/1994 12:35
27	NetEye NMS	woody.softstar.it	19/10/1994 12:35
28	NetEye NMS	woody.softstar.it	19/10/1994 12:35

**Done**

Figure 10.5: List of tickets that satisfy a query condition

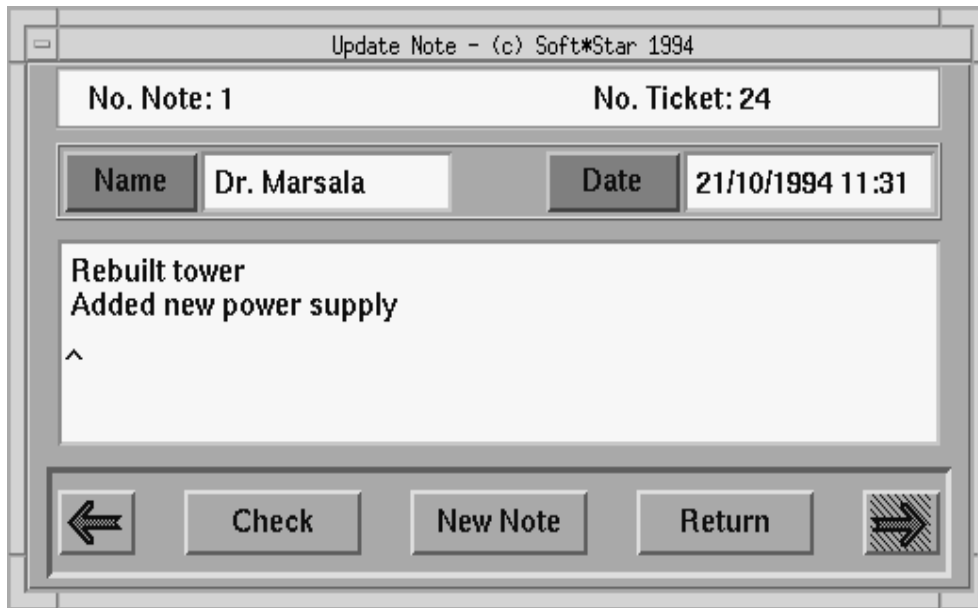


Figure 10.6: Adding a note to a trouble ticket

## 10.4 Updating tickets.

Every time you take an action on your network trying to solve a problem you will need to update the information held by the trouble ticket associated with the problem. Updating a trouble ticket involves popping up the update form by pressing the **Update** button on the trouble ticket button box. The form is similar to the one used to select information. You can either enter a ticket number, that uniquely identifies a ticket or enter a search condition and then press **OK**. From the list of selected tickets click on the one you want to update and the update form will be posted. You will not be able to edit all the fields; the table shows which elements are readonly and which are read-write

Readonly	Read-write
Priority	Scope
Source	Status
Type	Close Code
Site	Closed
Owner	Notifications
Opened	Problems

To browse the notes associated with that particular trouble ticket press the **Notes** button in the lower part of the window and you will get the Notes window as in figure 10.6. You can move back and forth between the different notes by pressing the two arrow buttons; while browsing the notes all the data will be readonly. If you want to create a new note, explaining the operations you did to fix a problem, press the **New Note** button and you will obtain an editable note page; it will be save in the RDBMS when you press **OK**.



Figure 10.7: Preparing the list of target notifications

After you have updated a trouble ticket you may want to notify one or more persons at a remote NOC or at a response center. *NetEye* is capable of sending notifications by email or by fax. To send a notification press the **Notify** button; the window in 10.7 will be shown.

Every trouble ticket can be sent to more than one person both by fax and by email. If it is the first time you send a notification for a specific ticket the two lists contained in the notification window will be empty. If you already have sent notifications for that specific ticket the lists will contain all the persons involed in the last notification. *NetEye* will always save the complete list of all the persons involved in the last notification. The list of all the possible receivers of a notification is extracted from *NetEye's Address Book* and the names are listed in the combobox. Use the two toggle buttons to select if you want to send a fax or and email or both to the selected person, then press the add button to include that person in the appropriate list. To remove persons select their names on the list and then press the remove button. As soon as you press the **Apply** button the request will be queue on the application server .

## 10.5 Deleting tickets.

To delete one or more trouble tickets from the database press the **Delete** button in the **Trouble Ticket button box**. As usual you will be prompted with the query preparation dialog where you will setup the conditions for the deletion. After you press the **Ok** button you will be shown all the tickets that match those specific requirements. Click on the the tickets you want to delete; as you select the items you want to delete they will be highlited. Clicking on one of these highlighted items will reset it to it's normal unselected state. When you are satisfied with your selections press **Done** to effectively destroy the selected tickets. You will be prompted with a confirmation dialog because this operation is irreversible; all selected data will be lost unless

you have backed-up your database.

## 10.6 Printing tickets.

To print one or more trouble tickets from the database press the **Print** button in the **Trouble Ticket button box**. As usual you will be prompted with the query preparation dialog where you will setup the conditions to use when extracting tickets from the database. After you press the **Ok** button you will be prompted for the print command to use; the default value is obtained from `REPORTCMD` environment variable. Once the print command has been determined, *NetEye* will compute the number of elements that will be printed and ask you confirmation if it should proceed in the operation or not. Here is the sample output after printing a trouble ticket with two associated notes.

```
Ticket Number      : 24
Ticket Priority     : High          Ticket Status      : Open
Ticket Source      : Phone         Ticket Type        : Unplanned
Ticket Scopes      : Many Sites   Ticket Close Code  : Other
```

```
Site/Line          : iunet-gw.iunet.it
Ticket Owner       : PENENGO MARCO
```

```
Ticket Opened      : 18/10/1994 10:00   Ticket Closed     : 25/10/1994 11:34
```

```
Problem Description:
Lightning struck microwave link tower
```

Actions or Notes:

```
Note Number        : 1              Note Date : 21/10/1994 11:31
Note Owner         : Dr. Marsala
Note Description   :
```

```
Rebuilt tower
Added new power supply
```

+++++

```
Note Number        : 2              Note Date : 22/10/1994 11:33
Note Owner         : Gedeone Gatto
Note Description   :
```

```
Measured SWR
```

New feeder

+++++

## 10.7 Faxing and emailing tickets.

As you have seen in the *Updating tickets* section you can notify a remote person with one or more trouble tickets you generated or updated. To start a notification press the **FAX-MAIL** button on the **Trouble Ticket button box**. As usual you will be prompted with the query preparation dialog where you will setup the conditions to use when extracting tickets from the database. After you press the **Ok** button you will be prompted for the for the list of destinations. As explained in section *Updating tickets* insert all the names of the persons you want to send the notification to and if you want to use email or fax or both. When you are done press the **Apply** button and all the requests will be queued with the application server.

## Chapter 11

# User and configuration management.

In *NetEye* all configuration aspects that are not strictly related to network management are handled by the **Application Server**. This process is a daemon that is used to update the **Address Book**, change some parameters at run time without restarting *NetEye* and handling the outgoing queues of fax and email messages.

To activate the Application Server press the **Tools** button in the *Control Panel* and select **Configure**. The Application Server will pop up the window shown in figure 11.1. From this main window select where you want to operate by pressing the related button.

### 11.1 Address Book.

All handling of names, addresses, fax numbers, email addresses etc. of the persons with which you interact are done in the **Address Book** window shown in 11.2.

This window is divided in three areas; on the left you have a list of all the current persons loaded in the database. When you click on one of these names the right side is loaded with all the information related to that person. In this side of the window you perform all the maintenance activities by using the buttons with iconic representation of these operations:

**Add** fill in the form and add the new record to the database.

**Delete** select a name from the list and press this button to remove it from the database.

**Modify** to save in the database the changes you have made in the form.

**Clear** reset the form to a clear state.

### 11.2 Output queues.

The **Application Server** handles the outgoing fax and email queues; when a request is queued it will try to serve it immediately. If for some reason it fails the request will remain queued until

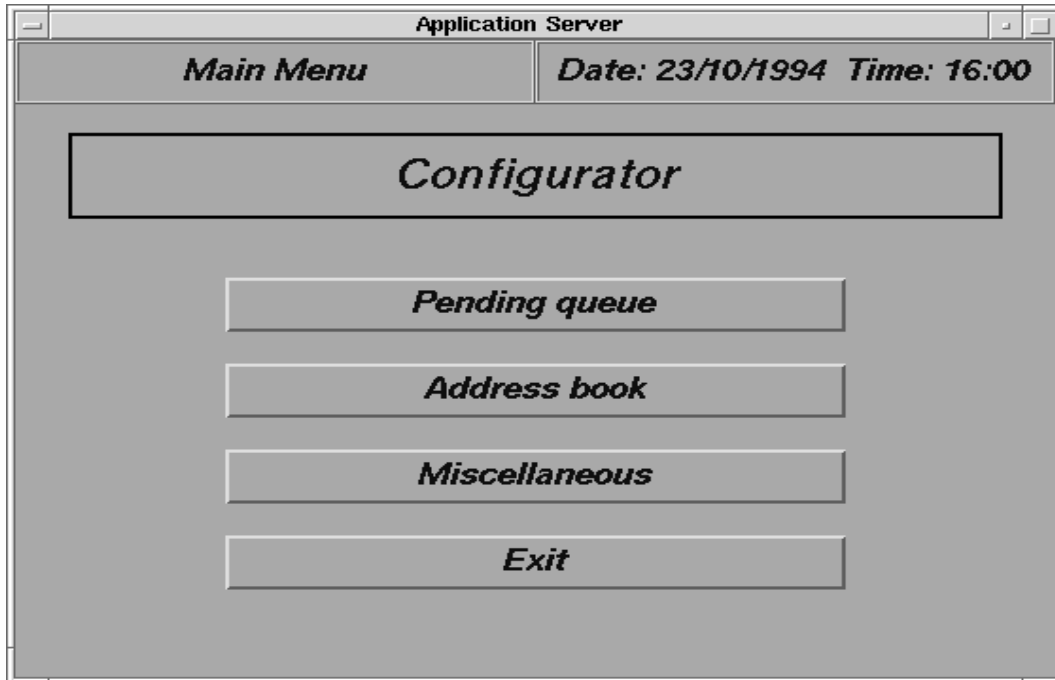


Figure 11.1: Application Server startup window

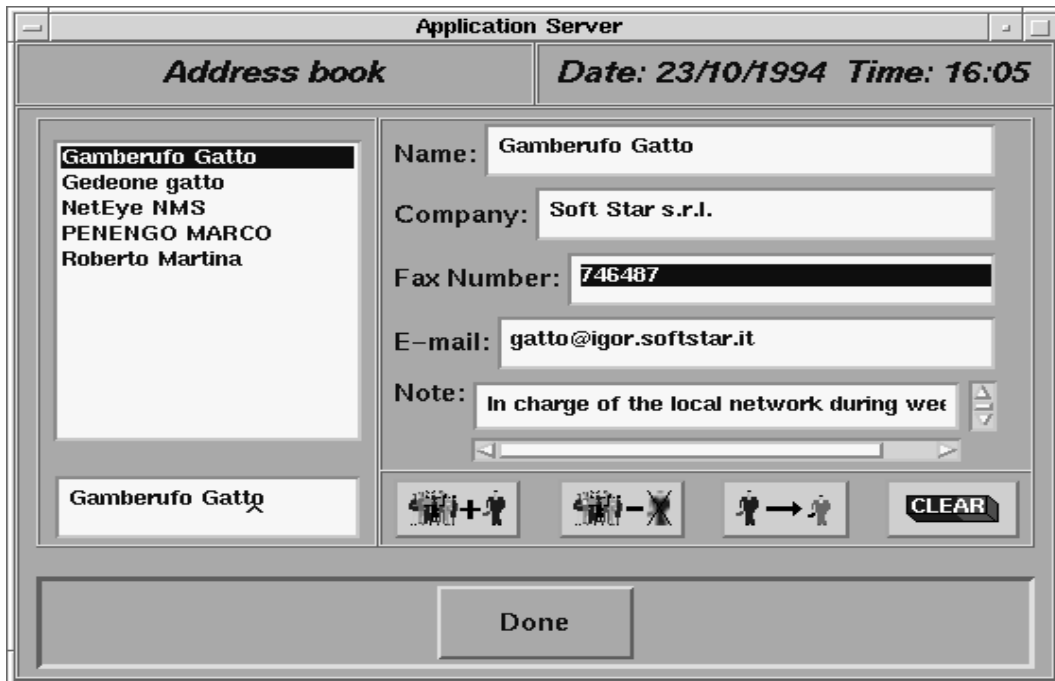


Figure 11.2: NetEye's address book



Figure 11.3: List of queued faxes and emails

the next request. By pressing the **Pending Queue** button you will be shown all the queued requests in a window as the one shown in figure 11.3. The information reported is

- ticket number
- owner of the ticket
- type of deliver (fax or email)

To handle the tickets waiting to be delivered, select one or more tickets from the list then use the buttons at the bottom of the window to perform one of these operations:

- immediately try to send. If the operation succeeds the selected tickets will automatically be removed from the queue.
- delete from the output queue
- list the contents of the selected tickets in a separate window, shown in figure 11.4

### 11.3 Miscellaneous configurations.

With the **Miscellaneous** button you display the window, shown in figure 11.5, used to dynamically reconfigure the following parameters used by the Application Server:

- email command
- fax command

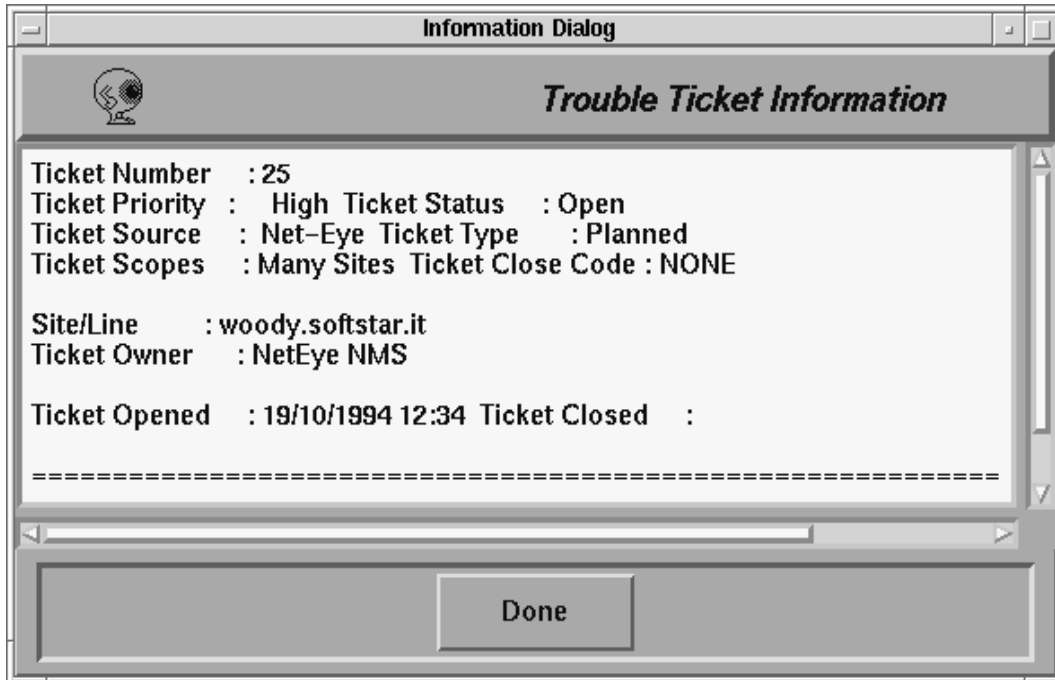


Figure 11.4: Contents of selected tickets waiting on output queue

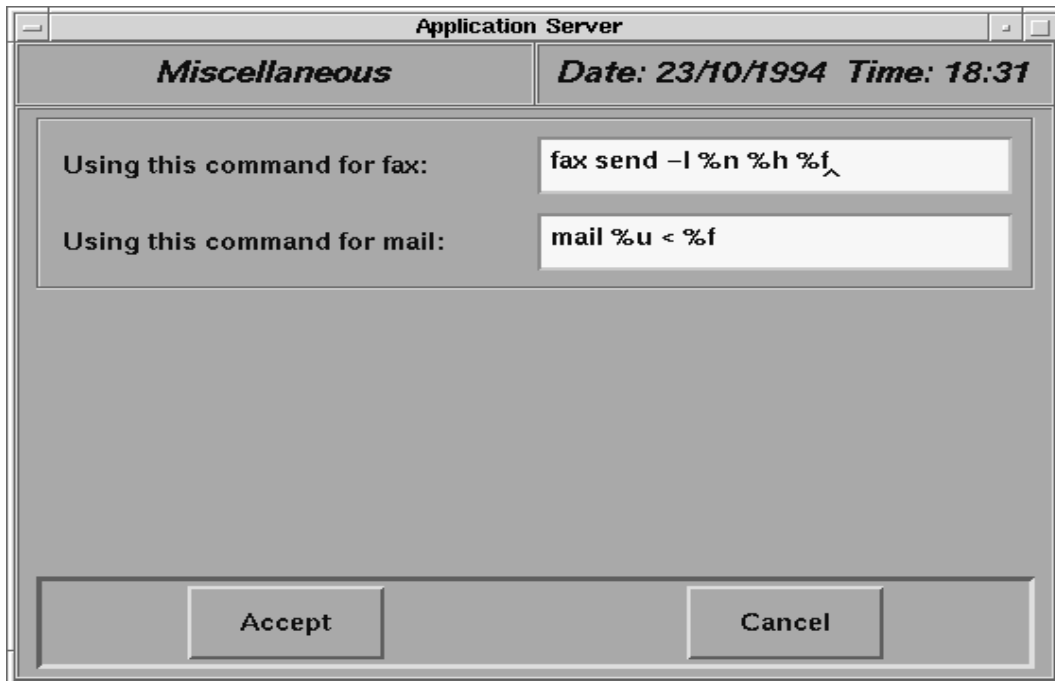


Figure 11.5: Application server parameter reconfiguration

## Chapter 12

# Application defaults.

All application defaults are located in directory `$NETEYEDIR/app-defaults` and are

**ControlPanel** for the Controlp Panel.

**TRap** for the Trap Query Tool.

**TRouble** for the Trouble Ticketing System.

**XEvent** for the Trap Visualization. Xevent

**SnmpMon** for the variable plotting tool.

All the resources that can be customized are listed in the next sections. The following resources can be specified for the map and for the windows that monitor and graph statistics.

**statusNormalColor** Specifies the color for the **Normal** state.

**statusWarningColor** Specifies the color for the **Warning** state.

**statusAlertColor** Specifies the color for the **Alert** state.

**statusSnmpUpColor** Specifies the color of an object that is manageable with SNMP.

**statusSnmpDownColor** Specifies the color of an object that was previously manageable with SNMP but that now only responds to ICMP.

**statusUnknownColor** Specifies the color of an object that is not currently managed or active.

**statusIcmpColor** Specifies the color of an object that is only manageable with ICMP

**valueForeground** Specifies the foreground color of the digits in the Control Panel counters.

**valueBackground** Specifies the background color of the digits in the Control Panel counters.

**upperBound** Specifies the maximum value a counter can reach. The default setting is 99999.

**lowerBound** Specifies the minimum value a counter can reach. The default setting is 0.

**valueLength** Specifies the number of digits displayed by a counter in the Control Panel. The default value is 5.

**maxPollIntervall** Specifies the maximum polling interval. This value must be greater or equal to one.

**priorityNormalColor** Specifies the foreground color to use when displaying the retrieved lines with normal priority from the Trouble Ticket database.

**priorityLowColor** Specifies the foreground color to use when displaying the retrieved lines with low priority from the Trouble Ticket database.

**priorityHighColor** Specifies the foreground color to use when displaying the retrieved lines with high priority from the Trouble Ticket database.

## Chapter 13

# Appendix.

### 13.1 Directory structure.

**app-defaults** all application defaults are in this directory. You can copy them to your standard app-defaults directory.

**bin** all executable are located here.

**bitmap** this directory contains the bitmaps used to represent object on the maps

**etc** various support files including `neteye.lic` that contains the activation key.

**gif** put all the backgrounds you want to use in your maps in this directory. The format must be gif.

**help** this directory contains the files used by the hyper help.

**libs** at the moment there is only one subdirectory, `afm`, that contains the fonts used by the graphing and plotting tools.

**log** various log information is recorded in this directory.

**mib** add all new mibs in this directory. Remember to update environment variable `MIBFILES`.

### 13.2 Database structure.

The schema of the database used by *NetEye* is shown in figure 13.1; the notation used is **IDEF-1X**. All commands to create the database are included in file `installdb` shell script; these commands depend on the RDBMS uses in your installation.

### 13.3 Internal protocols.

All elements of *NetEye* exchange information in order to perform their tasks. These communications take place using **UDP** protocol and the following set of services are defined in `/etc/services`

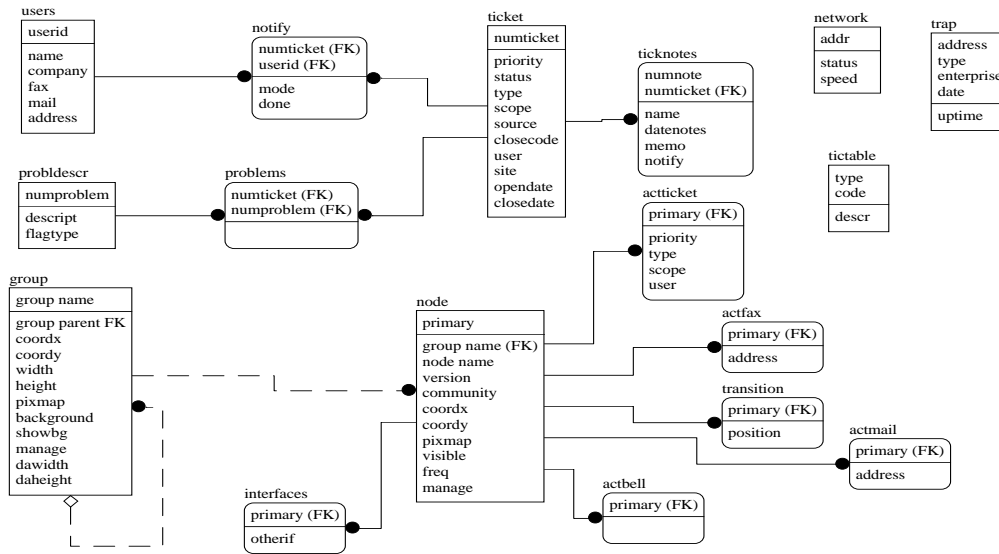


Figure 13.1: Database schema

```
# services used by NetEye
neteye      8001/udp
netpoll     8002/udp
snmpmonitor 8003/udp
xevents     8004/udp
controlp    8005/udp
cmdsnp      8006/udp
configure   8007/udp
```

### 13.4 Structure of help files.

The structure of the help files used by *NetEye* is very simply because they are flat ASCII. The only addition is that the words that will be highlighted and used to reference other help pages must be inside curly brackets. There is no limitation to the length of the highlighted text but they must be included in the index file `man.index` located in the `$NETEYEDIR/help` directory. The index file contains the referenced text followed by a column and then the name of the file containing the specific topic.

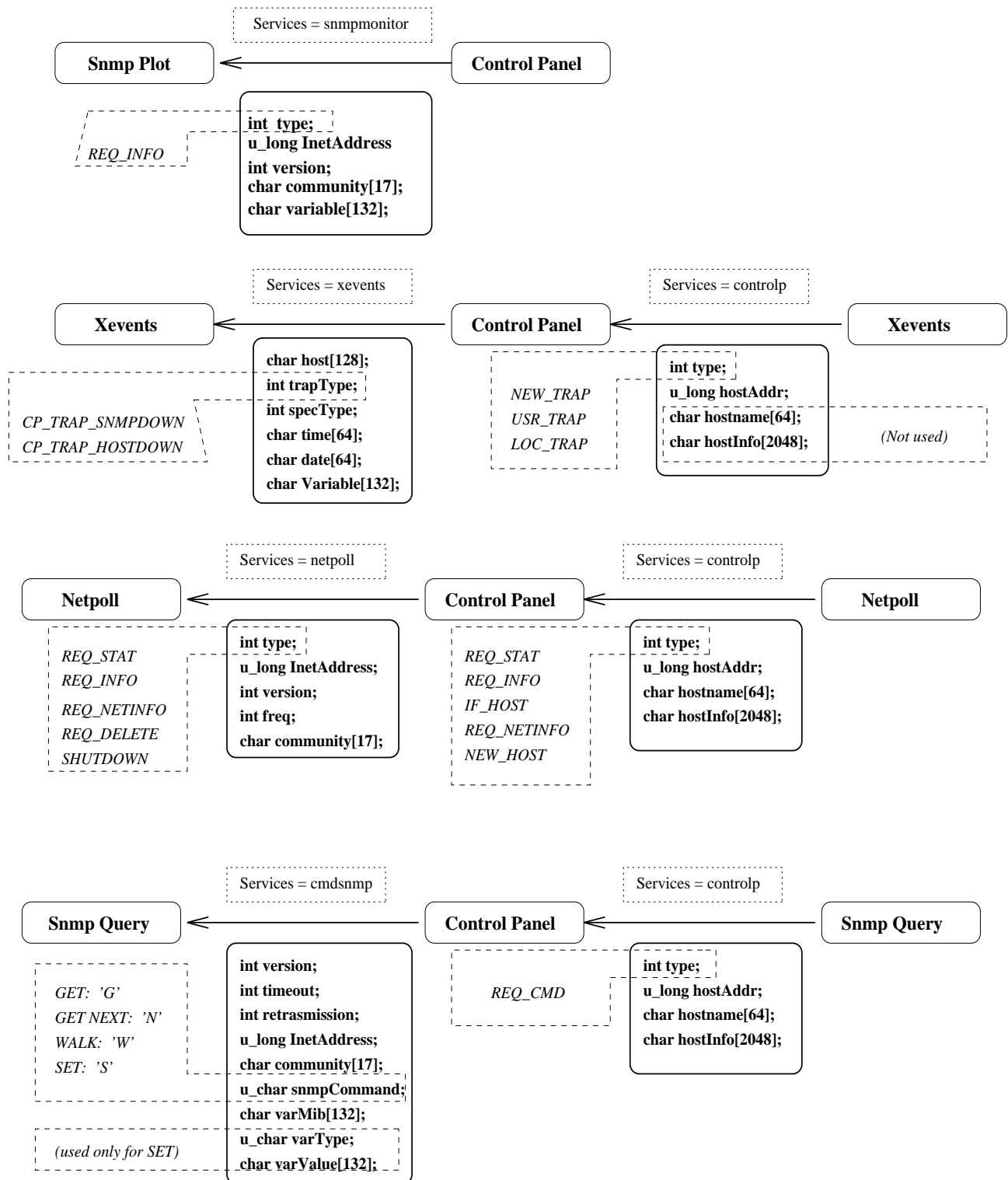


Figure 13.2: Protocol used to exchange information between various processes

# Index

- <all>, 65
- /etc/resolv.conf, 14
- /etc/services, 78
- \$INGRES/files/users, 9
  
- absolute mode, 39
- Action, 57
- Actions, 18, 43, 51
- activation key, 29
- Adding objects to a Group, 22
- Address, 17, 56
- Address Book, 69, 72
- address book, 44, 53
- AFMPATH, 8
- agent system, 1
- Alarm, 65
- Alarms, 27
- Alert, 26
- Alert threshold, 38, 42
- Application Server, 72
- application server, 69, 71
- applserver, 2
- Autodiscovery, 8, 14, 26, 29
  
- Background, 16
- Backgrounds, 23
- Bell, 43, 53
- bitmap, 22
- Browser, 34
  
- Cascade Menu, 3
- Clip, 18
- Clipboard, 14, 20, 29
- Close Code, 63
- Close Date, 63
- Command buttons, 34
- Community, 17
  
- Compose, 36, 47
- Control Panel, 2
- controlp, 2
- ControlPanel, 76
- coordinates, 38, 42
- Counters, 26
  
- database, 72
- DELAYPOLL, 8
- Delete, 18
- Delete All, 18
- Down, 26
  
- email, 74
- Emulate3Buttons, 5
- Explore, 31
  
- fax, 74
- FAXCMD, 7
- FAXHDR, 7
- Filter, 31, 56
- First, 27
- Frequency, 17
  
- Get, 46
- get, 1
- GetNext, 46
- GIF, 23
- Graph, 18, 37
- Group, 16
- Groups, 15
  
- Help, 27
- Hide, 23
- hierarchical view, 18
- highlightColor, 27
  
- ICMP, 26

- Ignore, 31
- Interfaces, 18
- IP address, 58
- iso, 48
  
- landscape, 40
- leaf node, 46
- license, 29
- List, 65
- Load, 27
- Locate, 22, 54
- Locating Objects, 22
- Logging, 38, 40
- Logical Network Object, 15, 51
- lowerBound, 77
  
- Mail, 44, 53
- MAILCMD, 7, 44
- Management Information Base, 1
- management system, 1
- manager, 1
- Mask, 31
- maxPollIntervall, 77
- MAXSNMPSOCKET, 8
- Menu bar, 3
- MIB, 1
- MIB Browser, 37, 46
- MIB description, 34
- MIB tree, 34
- mib-2, 34
- mib.txt, 6
- MIBDIR, 6
- MIBFILES, 6
- Miscellaneous configurations, 74
- Modify, 18
- mouse buttons, 3
- Moving objects, 20
- Multiple deletes, 20
  
- name server, 14
- nesting of logical network objects, 15
- NETEYEDIR, 6
- netpoll, 2
- NETTIME, 8
- Network Explorer, 29
  
- Networks area, 29
- New, 26
- Next, 27
- Notes, 68
- notification, 69, 71
- Notify, 69
- null object, 50
  
- Object area, 29
- Object Identifier, 34, 47
- Object Identifiers, 36
- Output queues, 72
- Owner, 44, 53
  
- Party, 17
- PATH, 6
- Physical Network Object, 15, 51
- Polling Interval slider, 37
- polling rate, 37
- POLLTIME, 8
- Pop Up Menu, 18
- portrait, 40
- pre-edited problems, 63
- Previous, 27
- Print, 44
- print, 70
- Priority, 44, 53, 63
- priorityHighColor, 77
- priorityLowColor, 77
- priorityNormalColor, 77
- Private Mib, 36
- private.entreprises, 34
- Problem Dialog, 63
- Pull Down Menu, 3
  
- Query, 18
- queue, 74
- queues, 72
- Quit, 24
  
- relative mode, 39
- REPORTCMD, 7, 70
- Resetting Counters, 27
- Retry Request, 47
- RFC-1155, 2

- root, 36
- schema, 78
- Scope, 63
- Scopes, 44, 53
- Select All, 29
- services, 78
- Set, 46
- set, 1
- Show, 18, 23
- SMI, 2
- SNMP, 1
- snmp agent, 46
- SNMP GET, 18
- SNMP GETNEXT, 18
- SNMP SET, 18
- SNMP Traps, 27
- SNMP walk, 47
- SnmppMon, 76
- snmpplot, 2
- snmpquery, 2
- snmptrapd, 2
- Start time, 58
- Status, 38, 63
- Status Line, 26
- statusAlertColor, 76
- statusIcmpColor, 76
- statusNormalColor, 76
- statusSnmppDownColor, 76
- statusSnmppUpColor, 76
- statusUnknownColor, 76
- statusWarningColor, 76
- Stop time, 58
- Structure Of Management Information, 2
- subnetmask, 31
- Switch, 41
- Telnet, 18
- Thresholds, 42
- Ticket, 44, 53
- Ticket Number, 63
- time interval, 58
- Time Out Request, 47
- Tools, 58
- transition matrix, 43, 51
- TRap, 76
- trap, 1
- Trap Query tool, 58
- trap type, 58
- Traps, 27
- traps, 54
- trigger, 42
- TRouble, 76
- Trouble Ticket, 61
- Trouble Ticket button box, 61
- tt, 2
- Type, 44, 53, 63
- Ungroup, 18
- upperBound, 76
- Value, 38
- valueBackground, 76
- valueForeground, 76
- valueLength, 77
- variable types, 48
- version, 27
- Warning, 27
- Warning threshold, 39, 42
- XAPPLRESDIR, 6
- Xconfig, 5
- XEvent, 76
- xevent, 2