AXX 9840 INSITE

System Administration Guide

ERICSSON

ERICSSON

1 Installation of AXX 9840 INSITE

1.1 Introduction

AXX 9840 INSITE

AXX 9840 INSITE is the Network Management System (NMS), supporting and automating the network level fulfilment and assurance processes.

AXX 9840 INSITE scales from managing enterprise networks from a centralised, single PC to manage nationwide networks in a distributed environment, supporting multiple users and roles 24*7. Furthermore, it features an intuitive and powerful graphical representation of network and nodes, with their corresponding conditions, resulting in immediate overview and control.

AXX 9840 INSITE offers extensive integration possibilities with 3rd party EM systems (southbound) and OSS (northbound) through the AXX 9880 MEDIATOR plug-in framework.

AXX 9840 INSITE is based on AXX 9800 TMN

- the platform and framework on which the Ericsson AXXESSIT management system product portfolio is built. AXX 9840 INSITE is designed with a multi tiered and component- based architecture, providing a tremendous flexibility in functionality and integration.

This is a prerequisite for a short time-to-market when it comes to functionality and integration. Furthermore, it ensures that Ericsson AXXESSIT management products keep up with the pace of network technology development.

AXX 9840 INSITE is based on the New Generation OSS (NGOSS) initiative from the TM Forum, and provides Ericsson AXXESSIT with the ability to offer a comprehensive and full management solution for Ericsson AXXESSITs expanding portfolio of leading edge network products.

1.2 This chapter

This chapter describes the tasks involved in installing and upgrading client software and server software of the AXX 9800 TMN Products.

The installation and upgrade is provided and supports several alternative installation or upgrade flows:

- Installing client and/ or server software
- Installing license
- Installing reconfigured database solutions for use by AXX 9800 TMN

The installation wizard is a guide through the installation process. See “Install License(s) with License wizard” on page 40.

See “Trademark Credits” on page 114.

1.3 Set Up Environment

1.3.1 Requirements

The system administrator must identify the required parameters to install or upgrade the AXX 9840 INSITE system, such as database solution, server host name and IP/ port address, number of clients and size of network.

1.3.1.1 SW and Licenses

AXX 9840 INSITE is a licensed product. See “Verification of Installation” on page 17.

Installation of licenses are an optional part of the installation wizard. The Install License Wizard tool can be started separately after the TMN system installation.

AXX 9840 INSITE installation packages:

- Downloading from AXXESSIT web page or FTP server. Access is controlled by user name and password

- An installation CD provided as a part of the purchased AXX 9840 INSITE system.

The product installation is implemented using Install Anywhere v.6.1.

1.3.1.2 HW

Client

See “Minimum system requirements - guideline” on page 5

Server

See “Minimum system requirements - guideline” on page 5

Screen settings

Recommended screen setting is: 1024*768 (32bits) True Color.

Minimum: 1024*768

1.3.1.3 Number of managed NEs

The required RAM is approximately 2MB pr NE.

For #NE > 100 no less than 2GHz is recommended.

1.3.1.4 Platforms

Client - Server

The client and server software platforms are installed with the versions and patches required by the AXX 9840 INSITE client and server. The client and server part of the system can be deployed on the following Operation Systems (OS):

- Windows[1]
- Solaris
- Linux

1. Tests has proven that the AXX 9840 INSITE improves stability running on the Windows Server 2003 Enterprise Edition. Running on this server, AXX 9840 INSITE serve a large number of clients without causing loss of events. Thus this server platform is recommendable

1.3.1.5 Database

The TMN database can be implemented by several 3.rd party database solutions. Apart from the embedded Derby database, the database solutions must be pre-configured to be accessed by the TMN installation package. The Derby is installed as a part of the TMN Installation , and does not require pre-configuration.

See “Database Support” on page 85.

1.3.1.6 Mobile radeon 7500 gfx chip

When AXX 9840 INSITE/Installation is run as the single java application on the system, and the application is closed, this may cause a blue screen on computers with the mobile radeon 7500 gfx chip. Please update the driver for the chip and/or reference Sun BugParade #4713003.

1.3.1.7 Performance on computers with specific power-settings

Some laptop computers may have power settings that reduce the graphical performance when running on battery. This may cause rendering problems in AXX 9840 INSITE.

To avoid this problem, the user has to change the settings on the computer, so the graphical performance is not reduced when running on battery.

Another work-around will be to add these properties in the lax -file.

SUN.JAVA2D.DDOFFSCREEN - This flag disables the use of DirectDraw and Direct3D for offscreen images.

sun.java2d.ddoffscreen=false

SUN.JAVA2D.D3D - This forcibly disables the use of Direct3D and avoids any Direct3D-specific problems.

sun.java2d.d3d=false

1.3.2 Minimum system requirements - guideline

AXX 9840 INSITE scales from managing enterprise networks from a centralized single workstation, to manage nationwide networks in a distributed environment, and supports multiple and concurrent users and user groups with a 24*7 system availability. This guideline is a reference to network scalability, system capacity, and technology compatibility, suggesting a minimum of system requirements given the size of network. The following guideline is PC-oriented, however, similar requirements will apply for Solaris platform.

(Legend: X - Supported, N/A - Not applicable)

	Minimum system requirements - guideline			**AXX 9840 INSITE**	**AXX 9820 CRAFT**
AXX 9800 TMN System requirements	**Server - small scale network**				
	Size of network		#NE	<20	
	Number of concurrent users (sessions)			2	
	Processor size			1GHHz	
	Memory (RAM)			512MB	
	Disk space available			50GB	
	Server - medium scale network				
	Size of network		#NE	20-100	
	Number of concurrent users (sessions)			3	
	Processor			1,25GHz	
	Memory			1GB	
	Disk space available			50GB	
	Server - large scale network[a]				
	Size of network		#NE	100- 500	
	Number of concurrent users (sessions)			5	
	Processor size			3GHz	
	Memory (RAM)			2GB	
	Disk space available			160GB	
	Client requirements				
	Clients			x	Standalone
	Processor size			1GHz	256MHz
	Memory (RAM)			256MB	128MB
	Diskspace available			200MB	200MB
	Display size (True Color)			1024*768	800*600
	Bandwidth[b]		per NE	32kbps	32kbps

Minimum system requirements - guideline				AXX 9840 INSITE	AXX 9820 CRAFT
3 rd. party Technology compatibility	**Operating system**	HW Platform	Release		
	Sun Solaris	Sun SPARC	10	x	x
	Sun Solaris	Sun SPARC	9	x	x
	Sun Solaris	Sun SPARC	8	x	x
	MS Windows	IA-32[c]	XP	x (Clients only)	x
	MS Windows	IA-32	Server 2003	x	x
	MS Windows	IA-32	2000 Professional	x (Clients only)	x
	MS Windows	IA-32	Server 2000	x	x
	MS Windows	IA-32	NT WS	x (Clients only)	x
	MS Windows	IA-32	NT Server	x	x
	Gnu Linux	IA-32	Debian Stable (Sarge)	x	
	Gnu Linux	IA-32	Red Hat EL v4	x	
	Gnu Linux	IA-32	Suse Linux ES 9	x	
	Gnu Linux	IA-32	Fedora Core 4	x	
	Gnu Linux	IA-32	Ubuntu 5.10	x	
	Database		Version		
	Oracle 10g		10.2.0.2	x	
	Oracle 9i		9.2.0.7	x	
	Oracle 8i		8.1.7	x	
	mySQL		4.1.20	x	
	mySQL		4.0.23	x	
	mySQL connector(recommended)		3.0.10 stable	x	
	Derby		10.1.2.1	x	

a. For #NE> 500, please see “Scalability” on page 7

b. Bandwidth requirement is given per network element for management communication.
c. Intel Architecture 32 bit

1.3.2.1 Scalability

The AXX 9840 INSITE is by design equally suitable for the entire range from small to large network sizes. Above are recommendations for some standard server system configurations, regarding typical platform configurations suitable for each product applied on some network sizes. Such recommendations are based on theoretical calculations, benchmarking tests, and other experience. Due to the heterogeneous nature of network design and real life applications, these recommendations do not provide any absolute limits neither for maximum nor minimum requirements. The actual load on the server and management communication network (MCN) is highly dependent on the type of NE, the stability of the environment, and the actual usage. Tasks such as continuous collection of a large number of PM counters increase the load, while a high percentage of “small nodes” normally reduces the actual load. For cases that, for network scale or server cost reasons, do not seem to fit into any of the recommended standard configurations, Ericsson AXXESSIT is happy to contribute in the analysis and calculation of actual platform requirements. Tailored recommendations may e.g. include solutions for distributed servers, with division of the network into regions and domains.

1.4 Installation of AXX 9840 INSITE

The Install Wizard will guide your installation. You can cancel the installation up until the wizard starts to install the files.

This procedure presents installation of AXX 9840 INSITE to the same computer, and uses the embedded database for illustration purposes.See “Install AXX 9840 INSITE Server” on page 16 and “Install AXX 9840 INSITE Client” on page 16 and see “Database Support” on page 85 .

Warning!

A new SW version of an AXX 9840 INSITE must be installed in a directory different from the existing AXX 9840 INSITE. Use the Data Migration wizard to move system data from the previous installation to the new installation.

Warning!

<u>Do not uninstall</u> to upgrade SW. See “The Data Migration wizard” on page 29.

1.4.1 Install AXX 9840 INSITE

The server and client are installed on the same computer.

1. **Insert** the SW **CD** in desired Drive on target computer.
2. Run **`INSITE.exe`** and the Install shield launches.

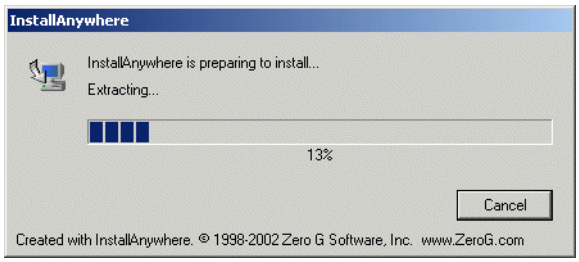


Figure 1 Install shield preparing installation wizard

The wizard opens the installation start page and following information is presented:

- Contained product
- Contained products versions
- Own build ID (Base Line)
- Own build date
- Initial instructions for the installer

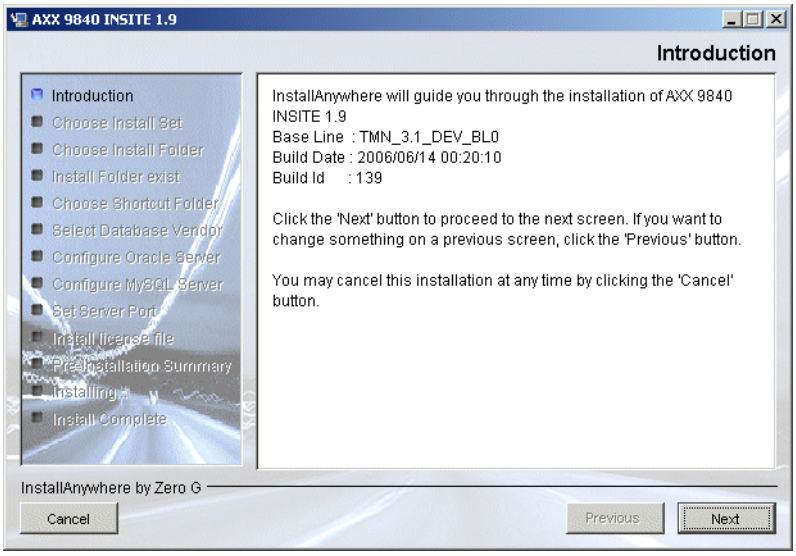


Figure 2 Introduction

3. Choose **Install** set and press **Next** to continue.

This example uses ‘ALL’ to install both server and client on the same computer.

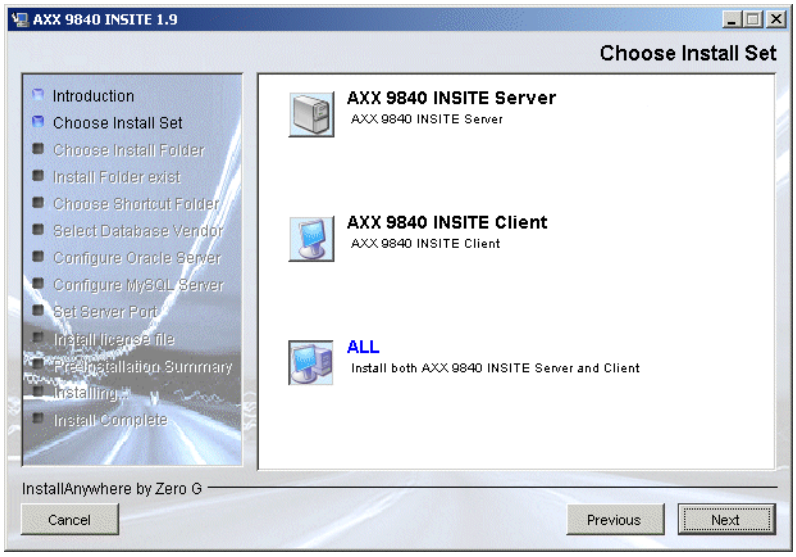


Figure 3 Install set All

See “Install AXX 9840 INSITE Server” on page 16 and “Install AXX 9840 INSITE Client” on page 16.

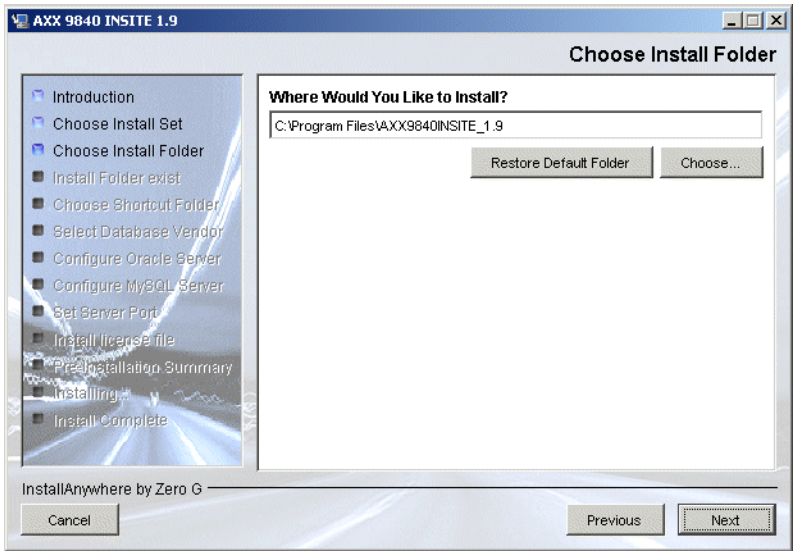


Figure 4 Install Folder

The number of required location parameters depends on OS and type of installation set (client/ server). Default choices are offered.

4. Choose **Install Folder** and press **Next** to continue.

NOTE! The wizard will detect any present AXX 9800 TMN installation, see Figure 5 If an Install folder exist, press Prevoius and change Install folder before proceeding.

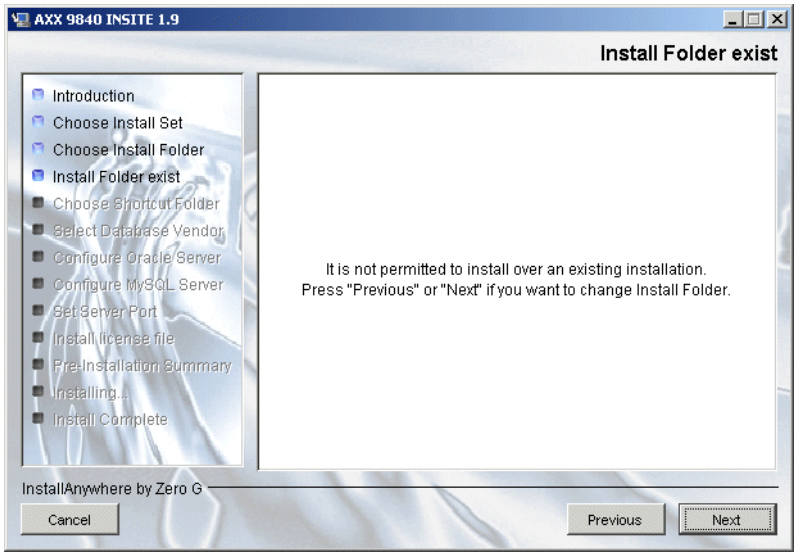


Figure 5 Install folder exist

In the next step you shall provide information on how the TMN executables shall be available on the platform:

- Started automatically from start-up scripts
- From icons in an new or existing program group
- In a start menu
- On a graphical desktop
- In a quick launch bar
- From a command line interface

NOTE! Input depends on the platform and is not described in the procedure

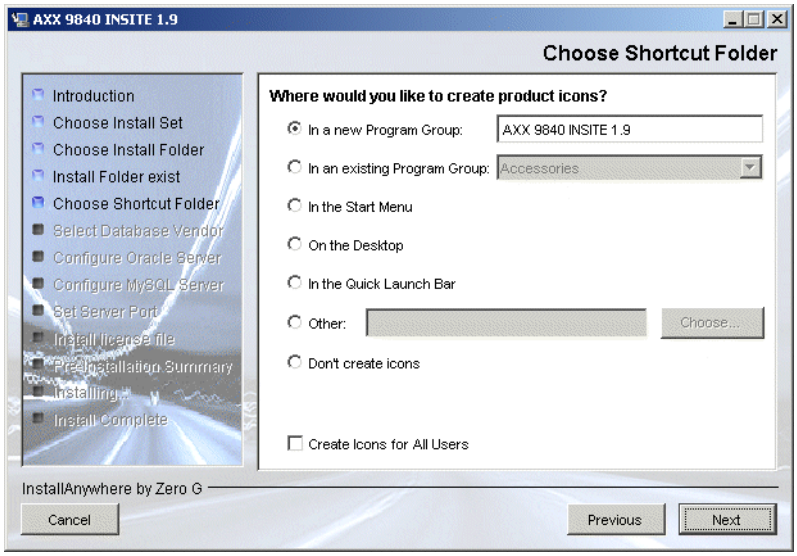


Figure 6 Shortcut Folder

5. Choose **Shortcut Folder** and press **Next**.

The system offers a set of startup options.

6. Choose **Database Vendor**.

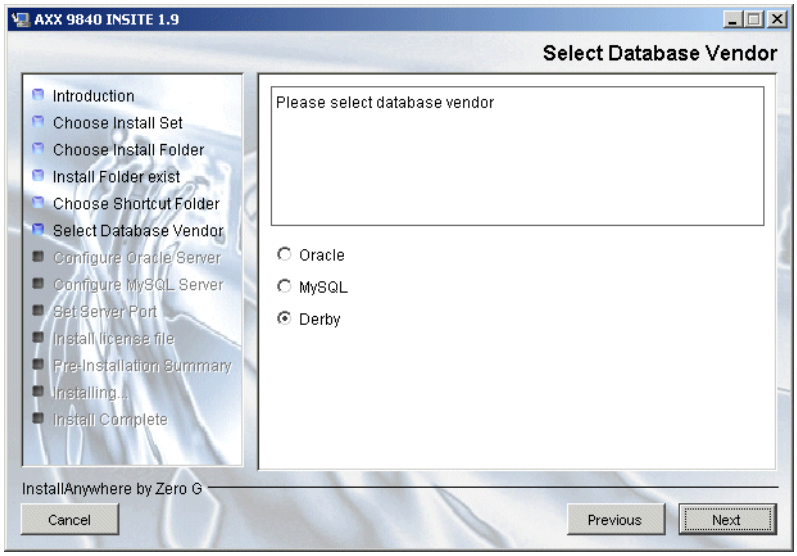


Figure 7 Select Database vendor

See page 5-85 for recommended database support

7. Press **Next**.

8. Select **Server port**.

The system presents 13170 as default server TCP port.

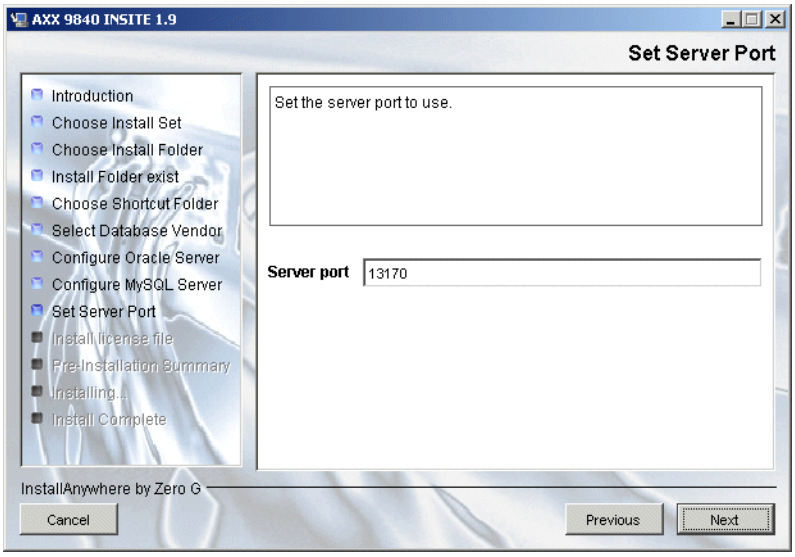


Figure 8 Set server port

9. Install **license** file.

NOTE! You can choose to start Install License Wizard immediately after the installation is completed. See “License Activation” on page 38.

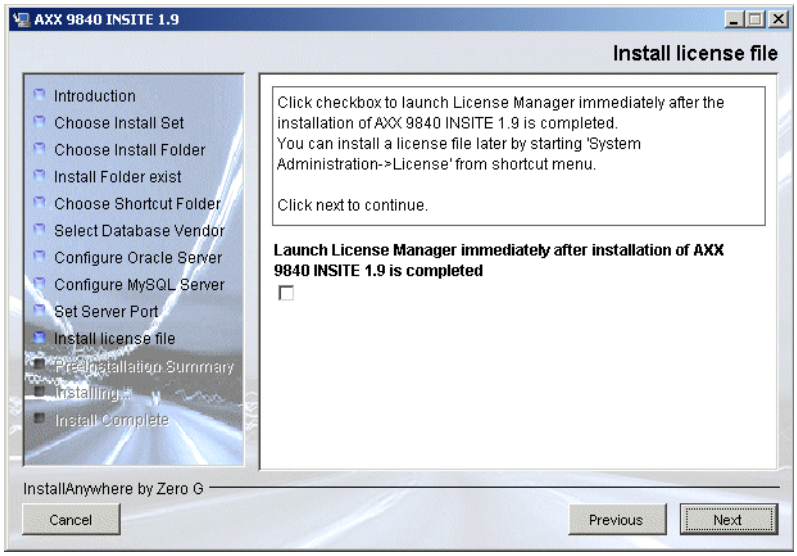


Figure 9 Install license file

10. Review **Summary** of your installation settings.

11. Press Previous to change input (Optional).

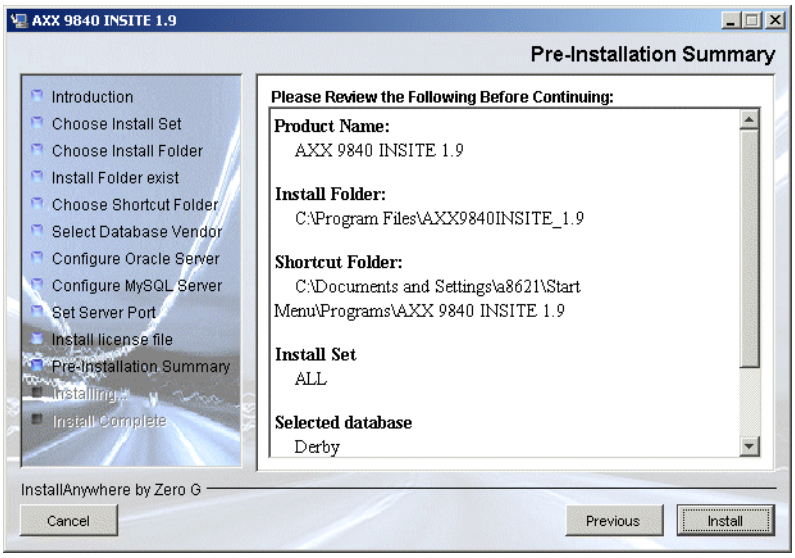
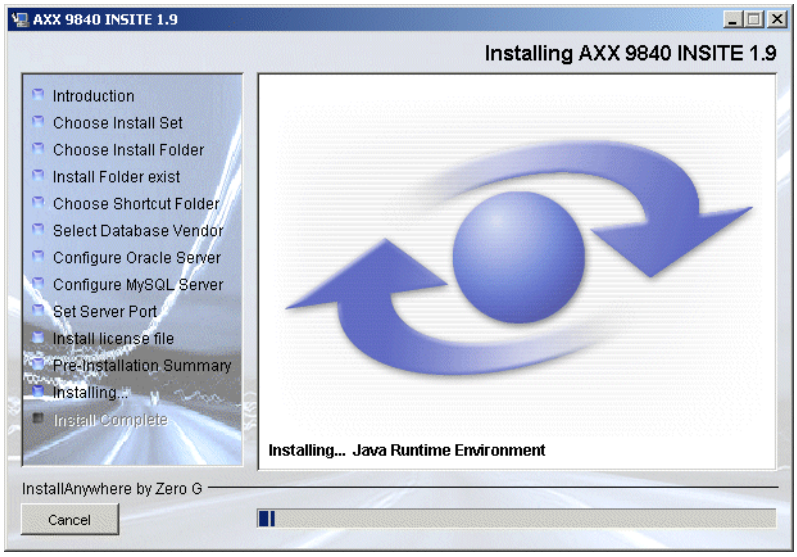


Figure 10 Pre- installation Summary

12. Press **Install** and view progress.

The AXX 9840 INSITE installation process is started.

NOTE! Cancel will exit the Installation process and close down the wizard.



AXX 9840 INSITE installation process

The installation process establishes:

- Directory structure if not already present
- Platform environment variables, depending on the software platform
- The system files areextracted and installed in the designated locations

The system files:

- Client and/or server executables and libraries
- Desktop icons and images
- Predefined reports and/or report templates
- Install License Wizard
- Administrator Tool
- Uninstall script

The administrator is requested to start the Install License Wizard to install the required licenses now or later, see “Install License(s) with License wizard” on page 40.

The installation is completed when the following window is displayed.

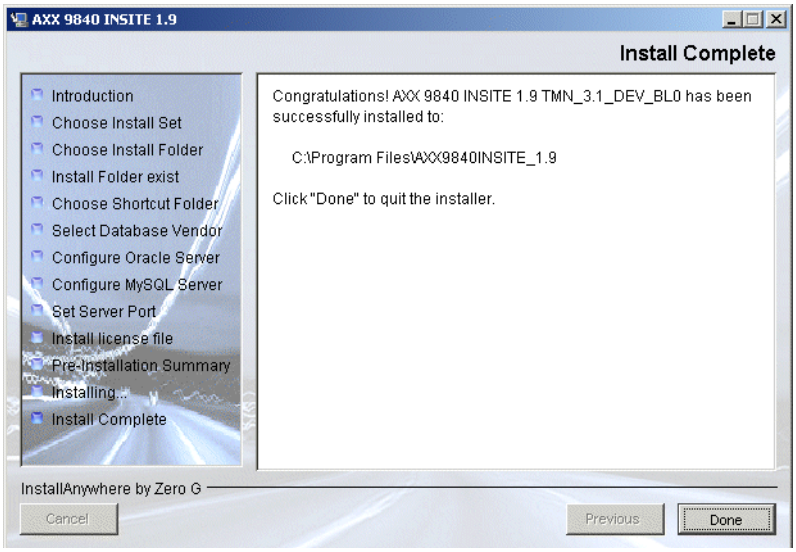


Figure 11 Install Complete

You will be notified about unsuccessful installation. See “Verification of Installation” on page 17.

13. Press **Done** to exit the Wizard.

Server and client are installed on the same computer, using Derby database. You can now logon and start the AXX 9840 INSITE, See “Start program” on page 2.

NOTE! See “Log-size” on page 86 . Recommended reading when installation is complete.

1.4.1.1 Install AXX 9840 INSITE to Solaris

1. **Insert** the SW **CD** in desired Drive on target computer.

2. Run **`AXX9840INSITE.bin`** and the Install shield launches.

See “Install AXX 9840 INSITE” on page 8.

1.4.2 Recommended Installation of AXX 9840 INSITE

The server and client are installed on different computers with reference to distributed client- server architecture.

1.4.2.1 Install AXX 9840 INSITE Server

1. Follow the guidelines. See “Install AXX 9840 INSITE” on page 8.
2. Choose **Server** Install Set.

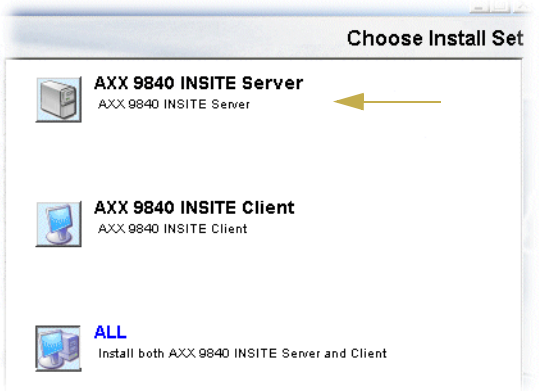


Figure 12 Install Client - install server

3. Follow the guidelines. See “Install AXX 9840 INSITE” on page 8

Pre- installation summary list Server as installed set.
The database is configured for use by the server, the server is installed and ready to be started.

NOTE! No network elements exists after a clean server installation. Only user(s) defined in the system after a clean installation is a system administrator with a default name and password.

If the installation was an upgrade, the configuration of the previous installation remains - including the user names and passwords.

1.4.2.2 Install AXX 9840 INSITE Client

This procedure is repeated to all client computers in network.

1. Follow the guidelines. See “Install AXX 9840 INSITE” on page 8.

2. Choose **Client** install set.

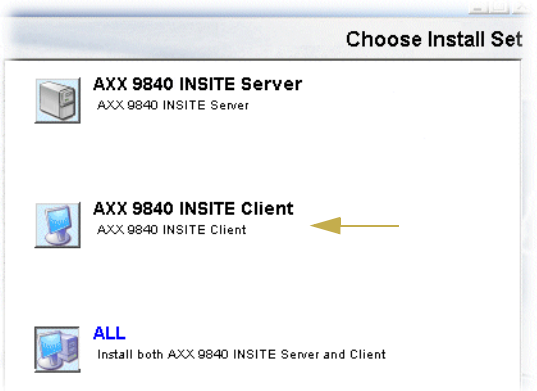


Figure 13 Install Client install set

3. Follow the guidelines. See “Install AXX 9840 INSITE” on page 8.

Pre- installation summary list Client as installed set to computer.
The client is installed and ready to be connected to a AXX 9840 INSITE server, providing that a valid User ID and password, the server host name or IP address is known. See “Logon” on page 2.

1.4.3 Verification of Installation

The installation should be verified for both server and client. Review the installation log and test a user logon, see “Logon” on page 2.

1.4.3.1 Installation Log

1. Go to `<installdirectory>`

The location of program files is according to installation settings.

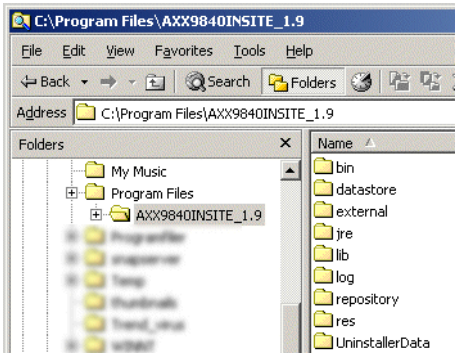


Figure 14 Install directory- example (Windows)

2. Open **Installation log** from **root directory**.

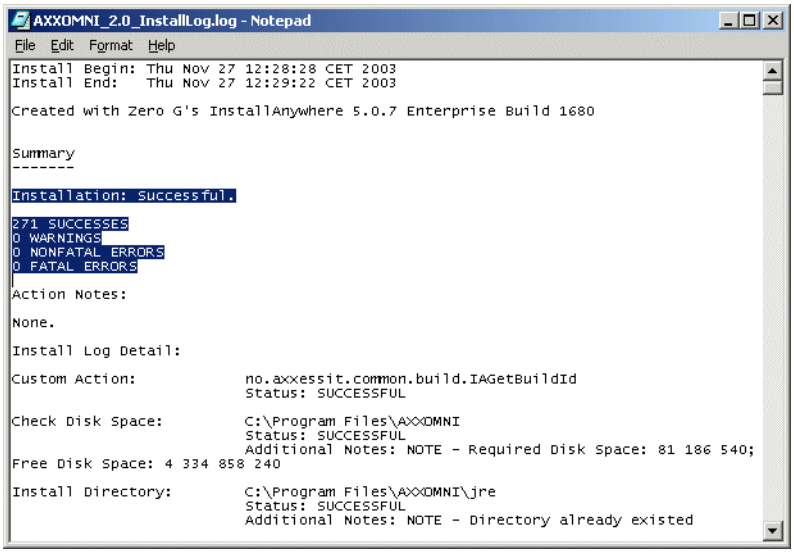


Figure 15 Example of installation log for a successful installation

3. **Verify** installation of selected Install Set.

1.5 Solaris and Gnu/Linux considerations

1.5.1 Installation directory

The AXX 9840 INSITE directory could be placed in the home directory of a normal user, for instance a user created specially for the purpose of holding the AXX 9840 INSITE files. For instance, with a user **`AXX 9840 INSITE`**, the installation directory will then be:

`/home/AXX 9840 INSITE/AXX 9840 INSITE`

The server still must be executed with root authorithy, but the installation will be simpler, as the user can run an Xserver.

Other good places to have the install directory would be

`/usr/local/lib/AXX 9840 INSITE`

or

`/opt/AXX 9840 INSITE`

1.5.1.1 External applications

Activating entries in the Help menu will issue the command **acroread** to present pdf files and the command **netscape** to connect to the Ericsson web site. If you prefer other programs for these functions, for instance xpdf or gpdf for pdf viewer, or firefox for html viewer, create links to those applications somewhere in your PATH.

TIP! We recommend gnome pdf , alternatively xpdf for viewing PDF files. Launching PDF files with ggv may fail.

The links can be created in users **$HOME/bin** directory, provided that this directory is in the users path.

Check that **~/bin** is part of your path:

```
env | grep PATH
```

create a link in a normal user's home directory:

```
jur@blade:~/bin$ ln -s /opt/csw/bin/firefox netscape
```

or

create a link in **/usr/local/bin** (as root):

```
root@blade:/usr/local/bin#
ln -s /opt/csw/bin/firefox netscape
```

1.5.2 Installation needs X server

The installer needs access to an X server to run. If you don't run your X server as root, which is a bad habit anyway, there are two possibilities:

- run the installer as a normal user
- run the installer as root, supply the necessary access rights to the X server

Note that in the world of the X Window System, the X-server is the program that owns the screens and controls the wondows. The actual user programs are called clients.

1.5.2.1 Run the installer as a normal user

Use an existing normal user id, or create one for the purpose. Install ether in the user's home directory, or make sure you have write access to the directory you want to install in. For instance, to install AXX 9840 INSITE in **`/usr/local/lib/AXX 9840 INSITE`**, do this as root:

```
root@blade:~# cd /usr/local/lib
root@blade:/usr/local/lib# mkdir AXX 9840 INSITE
root@blade:/usr/local/lib# chown jur AXX 9840 INSITE
```

when the installation is finished, you can and must run the server as root. Running the server does not require access to the X-server.

1.5.2.2 Supply access rights for root to X server

If you choose to run the installer as root, you need access to the X-server, which is running with the authority of a normal user. Without access to the server, you will get a message like:

X11 connection rejected because of wrong authentication.

When an X server starts, it will create a random string, called a cookie, in the **`users .Xauthority file`**. This file is readable by the owner only. Any program wishing to create windows on the X server, needs to present the cookie to the server. This is a minimalistic security measure. In addition, you need to tell each client where the server is with the environment variable **`DISPLAY`**.

1.5.2.3 Change to root without the "-" parameter

If your X server is running on the same machine, the easiest way to give axxess to the root user is to use the su command without the optional "-" <minus> parameter. Then most of the environment variables, including **`HOME`** and **`DISPLAY`** will be retained. The **`DISPLAY`** will point to the correct X-server and **`HOME`** will allow the programs to find the normal user's **`.Xauthorithy file`**. Since root can read any file, it can find the necessary cookie.

```
jur@blade:~$ echo $DISPLAY
blade:11.0
jur@blade:~$ echo $HOME
/space/home/jur
jur@blade:~$ su
Password:
# bash
root@blade:~# echo $DISPLAY
blade:11.0
```

```
root@blade:~# echo $HOME
/space/home/jur
root@blade:~# xclock
root@blade:~#
```

1.5.2.4 Give authority using xauth

This also works for an X-server running on another machine. In a shell where X clients already can run, issue this command:

```
johan@bread:~$ xauth list $DISPLAY
bread/unix:0  MIT-MAGIC-COOKIE-1
3fefdc3dcc129db15230e4e2c8dc1578
```

The current cookie is extracted from the X server and printed in hexa-decimal form. This entry points to a unix socket, but in the add command on the server we have to specify an ip address where the X server runs, or a dns name that resolves correctly.

In the root shell where you want to run X programs, run this:

```
root@fire:~# xauth add 10.20.43.38:0.0 .
3fefdc3dcc129db15230e4e2c8dc1578
```

Note that the hexadecimal string is copied. Then issue the command:

```
export DISPLAY:10.20.43.38:0.0
```

This will tell the X client programs where the X-server is located.

1.5.2.5 Use ssh

With ssh, secure shell, set up on the machines, you can tunnel the X traffic through the ssh connection, and ssh will automatically make sure that authorization and the **DISPLAY** variable is set up correctly. This may be the best option, when the X server is on the local machine or on a remote machine. This will work also when your X server does not listen on tcp.

On localhost:

```
ssh -Y root@localhost
```

With this, you will have X and the normal environment variables for root.
Start it from another machine:

```
ssh -Y root@fire
```

1.5.3 Running the server

1.5.3.1 Manual start

The server has to run as root, because it needs to open privileged ports.

The server can be started in a terminal window by mentioning its executable name:

```
root@fire#/usr/local/lib/AXX 9840 INSITE/bin/AXX 9840 INSITE_Server
and then
<ctrl-z>
bg
```

to push it into background

The problem now is that the server will be killed when the terminal window is closed. Using this command:

```
root@fire#(/usr/local/lib/AXX9840INSITE/bin/
AXX9840INSITE_Server &)
```

- will send it to the background, change the parent process to init, and disconnect it from the terminal. Then you can close the terminal window and the server will continue.

1.5.3.2 Manual shutdown

The server can be stopped using <ctrl-c> or closing the terminal window(if it is in running the foreground), or sending it the term signal

```
root@fire:~# ps -ef | grep AXX
root 22725 18701  0 15:42:19 pts/19   0:00 grep AXX
root 22686 18701  0 15:38:28 pts/19   0:31
/usr/local/lib/AXX9840INSITE_1.9/bin/../jre/bin/java -
Xmx536871872 -Xms268435936 com.
root@fire:~# kill -TERM 22686
```

But the standard way to stop the server is with the supplied Console program

```
root@fire:~#
/usr/local/lib/AXX9840INSITE_1.9/bin/Console
Starting application "AdminTool"
Successfully logged on
Looking up ManagementServer
Application "AdminTool" is ready
```

```
Menu
1. List active users
2. Stop server
x. Exit
Choice:
```

Then enter “2” to stop the server. If you start the Console program from a normal user, you will be prompted for a password first:

```
jur@fire:~$
/usr/local/lib/AXX9840INSITE_1.9/bin/Console
Starting application "AdminTool"
user id (default admin):
password:
server (default fire):
Successfully logged on
Looking up ManagementServer
Unable to connect to notification stream; will not
receive notifications

Menu
1. List active users
2. Stop server
x. Exit
Choice:
```

NOTE! The Console program will exit before the server exits. The server needs about ten seconds to stop.

The server can also be stopped with the client program:

1. Right click on the **SystemManagement node** in the Management tree.
2. Then select **Shutdown Server**.

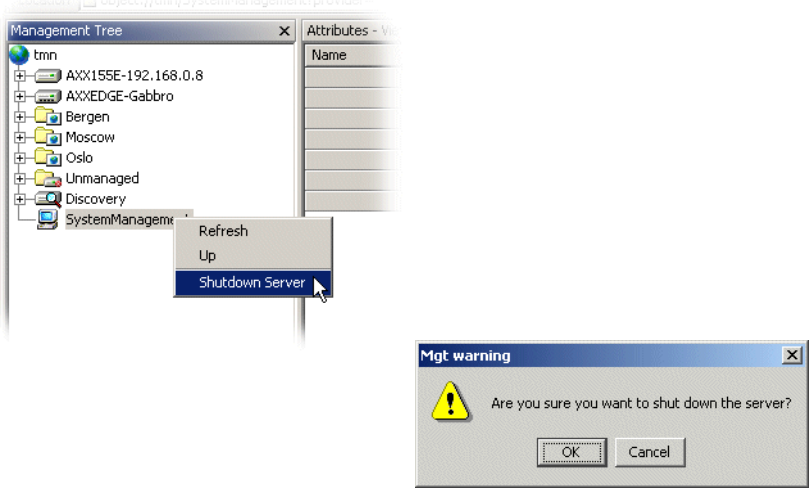


Figure 16 Management tree - Shutdown server

You have to be logged on as a user with SystemAdministrator role to be able to do this.

1.5.3.3 Automatic start and shutdown

The goal with this effort is to start the server automatically after the Solaris or Gnu/Linux operating system starts on the machine, and to stop the server just before the system stops after a shutdown command. The normal way to do this is to create a script to start or stop a server in the **`sysv`** service directory, normally **`/etc/init.d`**. Then a link to the script is created in **`/etc/rc3.d`** and another in **`rc0.d`**. The init system will run the script with parameter 1 set to “start” when the link starts with S, and parameter 1 set to “stop” when the link is called something starting with K.

For instance, a script **AXX9840INSITE** to both start and stop the server:

```
#!/bin/sh
case "$1" in
    start)

/usr/local/lib/AXX9840INSITE/bin/AXX9840INSITE_Serv
er &
        ;;

    stop)
        if  lsof -i :13171 | grep LISTEN  ; then
            /usr/local/lib/AXX9840INSITE/bin/Console -
stop
            /usr/bin/sleep 20
```

```
        fi
        ;;

    *)
        echo "Usage: /etc/init.d/AXX9840INSITE { start
| stop }"
        ;;
esac
```

Again, the Console -stop command does not need a password when it is running as root.

This script also checks if the server is in fact running, before stopping it.

Then the links:

```
cd /etc/rc3.d
ln -s ../init.d/AXX9840INSITE S99AXX9840INSITE
cd /etc/rc3.d
ln -s ../init.d/AXX9840INSITE K00AXX9840INSITE
```

1.5.4 Running the client

The client needs to run on the same machine as the X server. This is because it makes heavy use of the X protocol, and the latency in the client to X-server communication will quickly be a problem. Some parts of the client is more impossible then others. If you want to try, avoid using ssh forwarding, as this adds latency.

Note that other networking options are available: The client can be Windows on a Unix server and vice versa. Also, you can run a vnc (virtual network computing) X-server on the same machine as you want to run the client on, and view that vnc Xserver on another machine.

1.5.5 Miscellaneous

1.5.5.1 Find processes that are blocking the ports

If you see a log message indicating that the tftp port, the snmp trap port or the server ports could not be opened, find out which process holds the port:

```
root@fire:~# lsof -i :tftp
COMMAND PID USER   FD   TYPE        DEVICE SIZE/OFF NODE
NAME
inetd   184 root   22u  IPv6 0x30001b8fb38      0t0  UDP
*:tftp (Idle)
```

, in this case, the **inetd** is holding the tftp port

```
root@fire:~# lsof -i :162
COMMAND  PID USER   FD   TYPE        DEVICE SIZE/OFF NODE
NAME
java    2197 root   25u  IPv4 0x30003428668      0t0  UDP
*:162 (Idle)
```

The snmp trap port is held by process 2197

In Gnu/Linux the fuser command can also be used:

```
Firoot@orkl:~# fuser -n udp 69
69/udp:               719
```

Here the port 69 is used by process 719

1.5.5.2 Find the current runlevel

This command find the current runlevel:

```
who -r
```

In Gnu/Linux, you can also use the command:

```
runlevel
```

1.5.5.3 Decide if a server is alredy running

This works if your install-directory contains the string AXX

Solaris:

```
ps -ef | grep AXX
```

Gnu/Linux:

```
ps aux | grep AXX
```

You can also use the fact that the server listens to TCP ports 13170, 13171 and 13172.

1.5.5.4 AXX 9840 INSITE does not find printer - Solaris

When client is installed on Solaris, AXX 9840 INSITE does not find the printer.

Workaround

Print to file, print with default tool (ggv)

CUPS= Common Unix Printing System

Demonstration that CUPS on the machine works:

```
jur@blade:~$ lpq
Rico is ready
no entries
jur@blade:~$ echo lala | pr | lpr
jur@blade:~$ lpq
Rico is ready and printing
Rank    Owner   Job     File(s)                         Total Size
active  jur     101     (stdin)                         1024 bytes
jur@blade:~$ lpq
Rico is ready
no entries
jur@blade:~$
```

It is not enough that the commands lpstat and lp for CUPS is first in $PATH.

The default commands for Solaris must be renamed, then linked to the CUPSYS equivalents:

```
cd /usr/bin
mv lp lp.solaris
mv lpstat lpstat.solaris
ln -s /opt/csw/bin/lp .
ln -s /opt/csw/bin/lpstat .
```

1.5.6 Virtual memory

When another process is started from the system, there has to be at least as much free virtual memory as the current virtual memory usage. After a new process is started, the new process's requriement for space is lowered.

This is relevant for the pdf-viewer and browser in the client and for the Notification gateway in the server.

1.5.6.1 Solaris

In Solaris additional virtual memory space can be added with the **`mkfile`** and **`swap`** commands.

1.5.6.2 Gnu/Linux

In Gnu/Linux extra virtual memory space can be added with the **`dd`**, **`mkswap`** and **`swapon`** commands.

You can also create scripts to add swap space when the operating system starts.

1.6 Upgrade AXX 9840 INSITE

This section describes main tasks in the upgrade process of your AXX 9840 INSITE system. It is recommended that you read through the whole section before you start.

Source installation refers to your current installation, while **Target installation** refers to the new installation (new version). If you are

upgrading, use the Data Migration wizard, see section “The Data Migration wizard” on page 29.

NOTE! Always back up your database before an upgrade by using vendor specific database tools.

NOTE! It is wise to test an upgrade first on a test database before main system upgrade. The system will automatically create database schema.

Warning!

A new SW version of a AXX 9840 INSITE must be installed in a directory different from the existing AXX 9840 INSITE. Use the Data Migration wizard to move system data from the previous installation to the new installation

If the new AXX 9840 INSITE requires another database vendor than the previous (e.g from Derby to MySQL), the existing data must be moved to the new database using a third party tool.

1.6.1 The Data Migration wizard

1. **Shutdown** the server in your source installation.
2. Back-up your database before an upgrade by using vendor specific database tools.
3. Optionally, copy configuration files that have been manually changed.
4. Install **a new target** installation.

NOTE! This is exactly the same procedure as a new installation. E.g. database and licenses must be specified during the installation. Target installation shall be installed in a different directory than source installation. The database schema will most likely be changed the first time during server startup. It is not possible to automatically restore database schema.

Procedure: Start the wizard

1. Select Program>AXX 9840 INSITE Product>System Administration>**Data Migration from target installation.**

This wizard will help you to manage your AXX 9840 INSITE configuration.No changes will be performed until you press **Finish** and you can abort the wizard at any given time by pressing the **Cancel** button.

2. Click **Next** to continue to Step 1 **Task Selection**.

Select desired task. Every task and involved steps are described in:

- “Product Data Migration” on page 30
- “Display current Data Source information” on page 31
- “Modify settings for database” on page 32

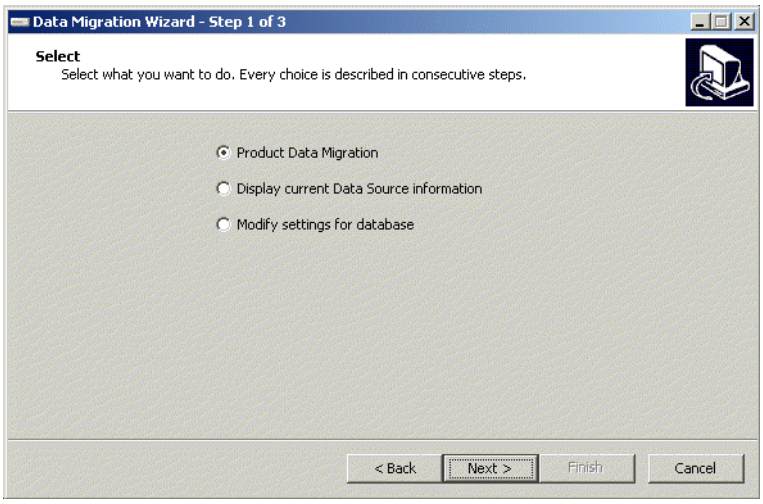
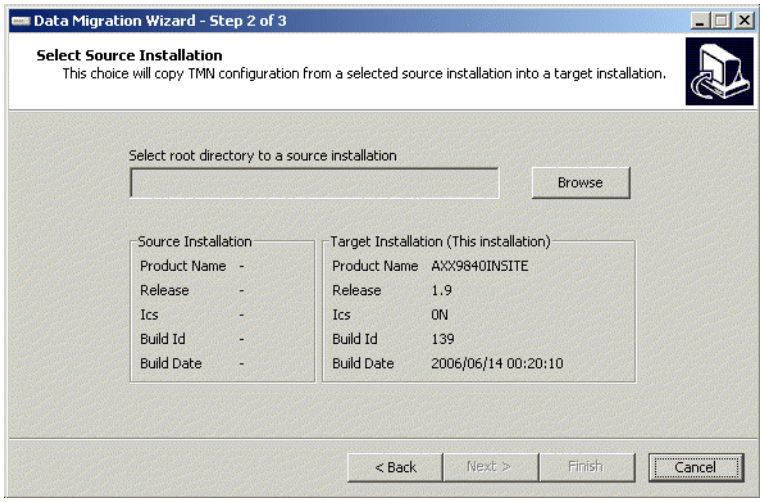


Figure 17 Data Migration Wizard - Task selection

1.6.1.1 Product Data Migration

1. Select **Product Data Migration** and click **Next**,

2. Browse for **root directory** of the **Source Installation**.



This choice will copy AXX 9840 INSITE configuration from a selected source installation into a target installation
Example:
Source: NMServer 1.8
Target: NMServer 1.9
User settings will be restored to factory settings at next server startup.
Audit configuration will be copied from source to target installation.

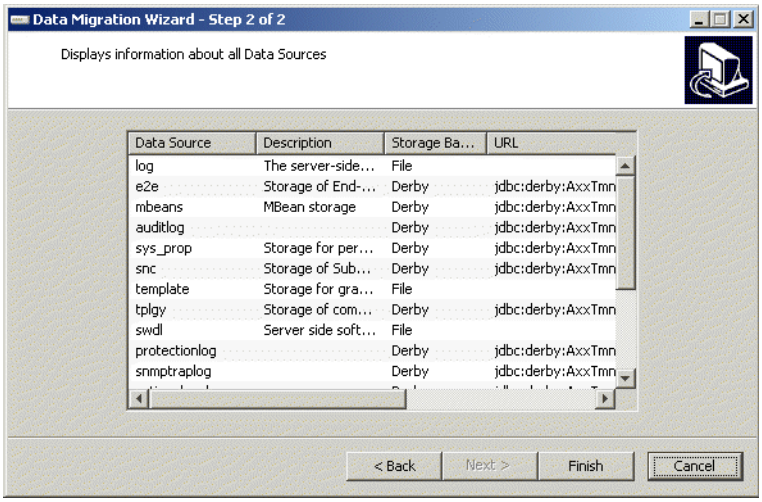
3. Press **Finish**.

4. **Start server** from target installation.

1.6.1.2 Display current Data Source information

Displays information about all Data Sources.

- Data Source
- Storage backend
- Description
- URL
- User



Press **Finish**.

1.6.1.3 Modify settings for database

If you want to view or modify the connection settings for the database, this option is very useful. Here, the current settings are displayed, and you can change the values if you have trouble because of type errors for instance.

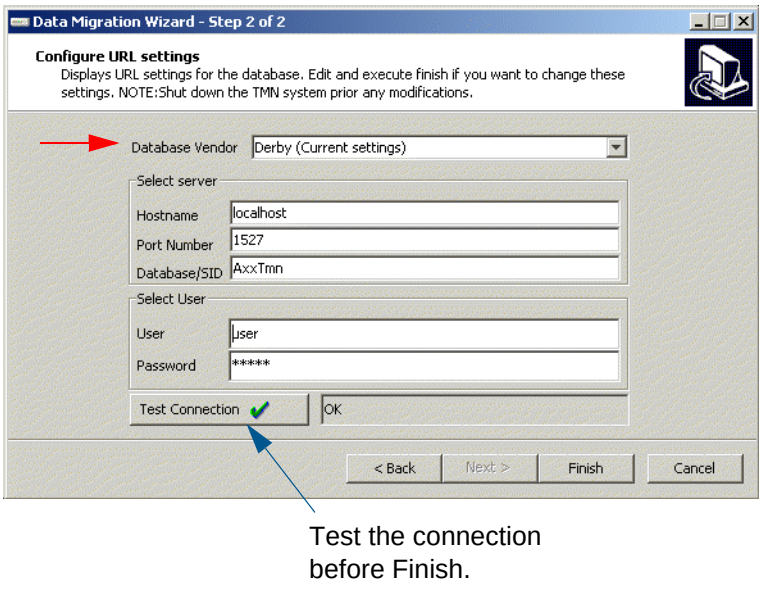


Figure 18 Modify database settings

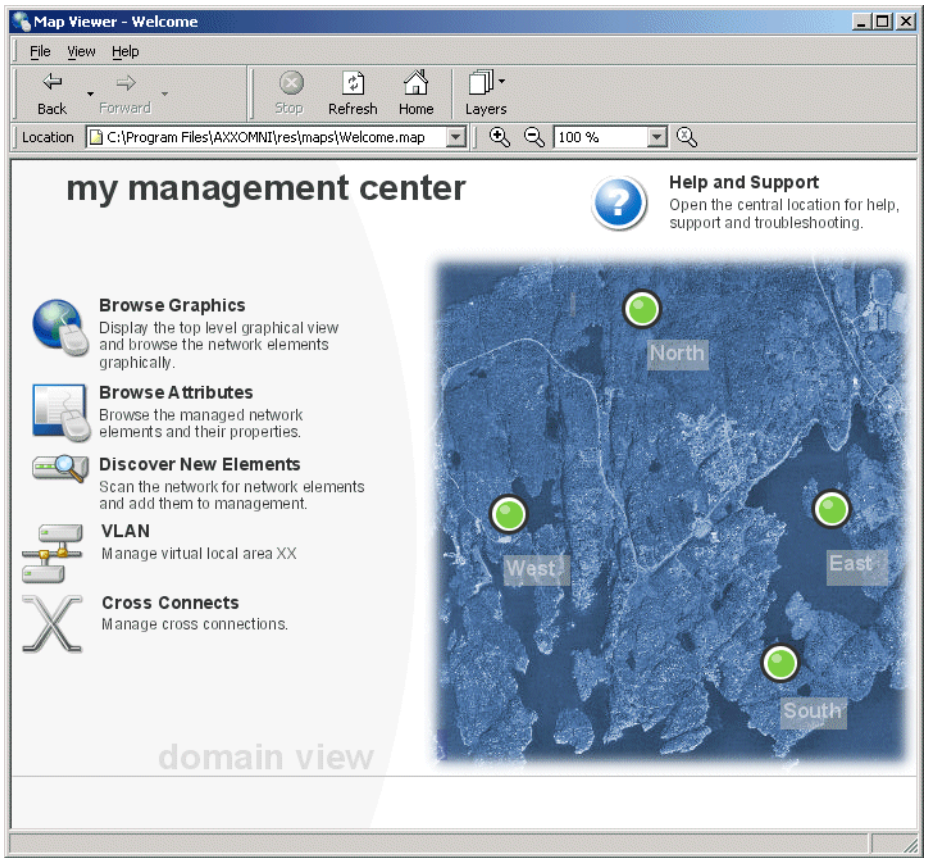
Progress indication

When Finish is pressed, a progress bar is shown along with information on each step involved in the selected task.The wizard operation is also logged in the **DataMigration.log**.

1.6.1.4 Maps

If you have created maps in the old installation, these can be manually copied from the old i **res/maps** folder to the new **/repository** folder.

NOTE! Make sure you don't overwrite any of the default maps, rename if necessary.
The maps can be copied to the server installation or the client installation, whatever is most convenient.



1.6.2 Start the Data Migration wizard on Solaris

1. **Enter** command line:

```
<install dir>/bin/./Data_Migration
```

The Data Migraton wizard is started, please follow guidelines given in “The Data Migration wizard” on page 29.

1.7 Uninstall AXX 9840 INSITE

No network element instances remain after you uninstall the system.

Warning!

Do not uninstall to upgrade SW. See “The Data Migration wizard” on page 29

1. Click Start>Program>AXX 9840 INSITE>**System Administration>**

2. Select **Uninstall** and follow the 3 step- instruction in dialog box.

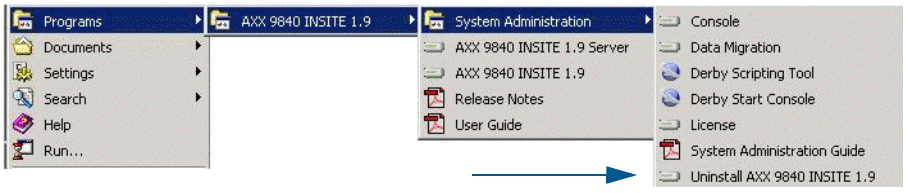


Figure 19 Uninstall program

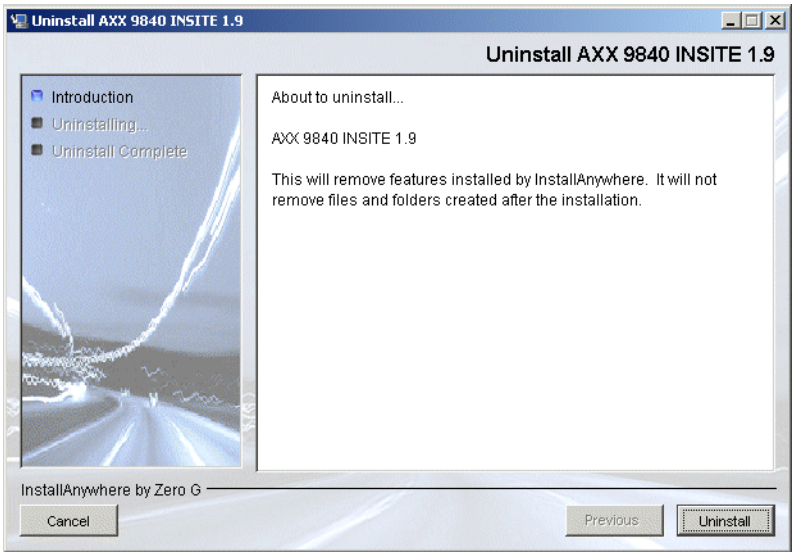


Figure 20 About Uninstall- example

3. Click **Uninstall.**

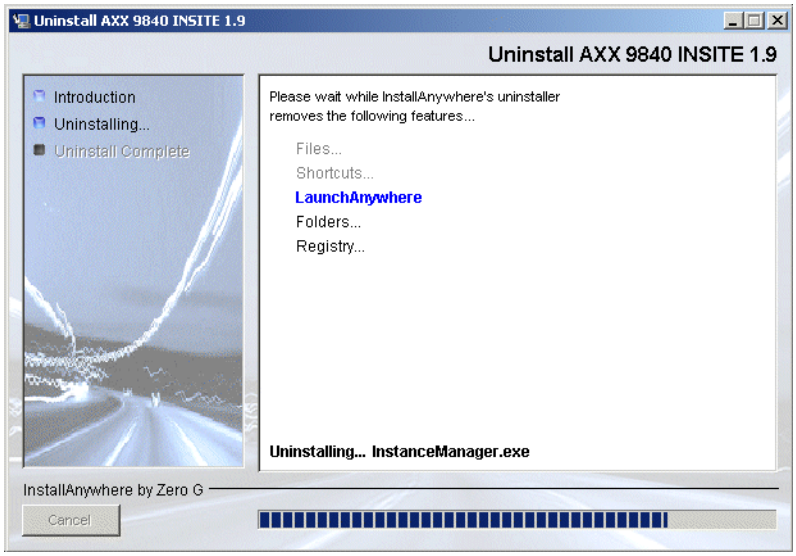


Figure 21 Uninstall progress

4. Select **restart option** and Press **Done** to complete uninstall**.**

NOTE! To avoid conflicts with processes pointing to folders in the uninstalled system, a Restart prior to new installation (in same program group) is recommended.

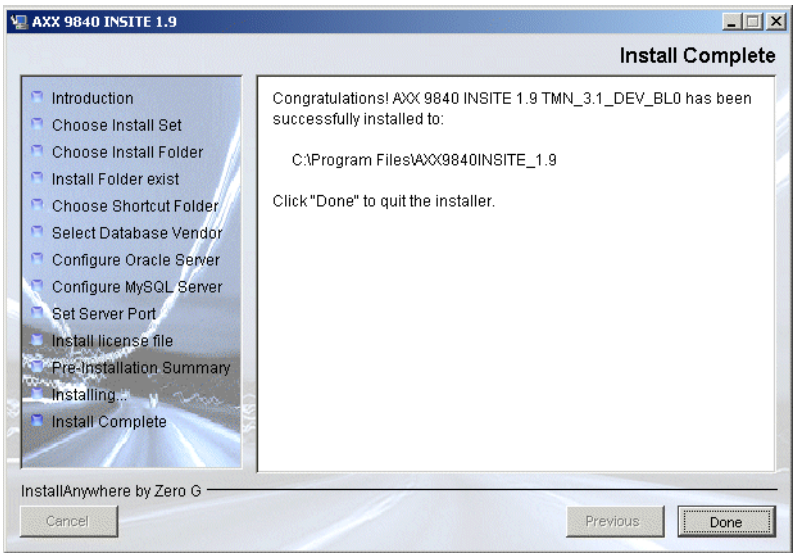


Figure 22 Uninstall completed

NOTE! Modified files after installation will not be removed (datastore, log). None- removable items must be removed manually from the folder.

2 License Management

2.1 Introduction

The purpose of this chapter is to describe how to manage the licenses related to the system.

A license can be installed and activated during installation of the system, or it can be updated later. Acquisition of licenses is not part of this description. See “Installation of AXX 9840 INSITE” on page 7.

2.1.1 License pool

Licenses for the AXX 9840 INSITE management system is delivered in a license file. Each licensed component, or feature, in the system defines a license point that has its own license key. It is encrypted and coded.

Licenses may be floating or node-locked, depending on the functionality the license point controls.

There are client licenses and server licenses. The file lists one or more licenses, depending on the product and the contract with AXXESSIT, and how many client licenses is part of the contract.

Server license

A server license can be a node locked license. All licenses are handled and stored on the server.

Client license

A client license is from a pool of floating licenses. When a client is started, it requests a license from the license pool on the server. If a license is not available the user will be notified and the operation is logged.

2.1.1.1 License features

AXX 9840 INSITE
The product contains the license features:

- NMServer_x.y
- NMClient_x.y
- NECount
- mapdesigner_x.y
- PM_x.y

2.1.2 License Activation

At the end of installation the installation wizard gives the choice to install the license now or later. Depending on the answer, the system displays the Install License Wizard or not. For detail on installation, see “Installation of AXX 9840 INSITE” on page 7.

The license(s) are installed and the management system can be started and applications are available according to the licenses.

2.1.3 Licensed Functionality

2.1.3.1 AXX 9840 INSITE

- Server selectable node-locked on IP address or un-licensed
- Number of concurrent users (floating)
- Number of network element instances under management (floating)
- Map Designer (floating - to be associated with the client licenses)
- Time/date limitations

2.1.4 License Identification

2.1.4.1 Customer attributes

- Customer Name
- Agreement number
- Purchasing date
- Product ID
- Contact data: Address, phone, e-mail, contact person (optional)

2.1.4.2 License specification

- Licensed feature ID
- Expiration date
- Type of license (node-locked / floating)
- Additional parameters required for the specific license type

Each of the license files contain one or several licenses. The files are encrypted.

2.2 Install License Wizard

The system administrator is supported by an Install License Wizard.

The system administrator can get an overview of currently used licenses in the system.

When the system administrator needs to install and activate a new license, the Install License Wizard guides the administrator through the steps of obtaining and installing a license.

2.2.1 License Parameters

The tool presents the licenses with the following parameters:

- Product
- Expiry date

- Type of license (floating or node locked)
- Contract

2.3 Install License(s) with License wizard

1. Start > Programs > **AXX<product>**.
2. Choose **License** from **System Administration** program menu.:

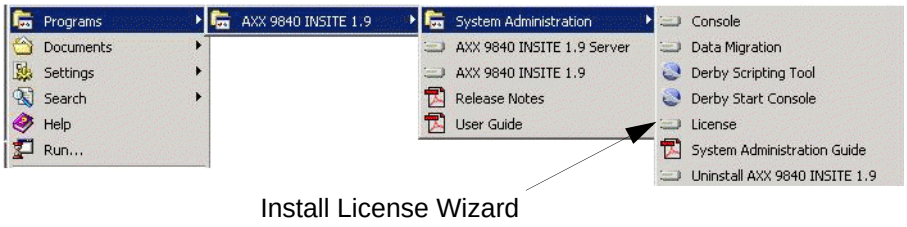


Figure 23 Start Install License Wizard

The Install License Manager wizard launches.

Figure 24 Install License Manager wizard - Welcome

3. Click **Next** to continue.

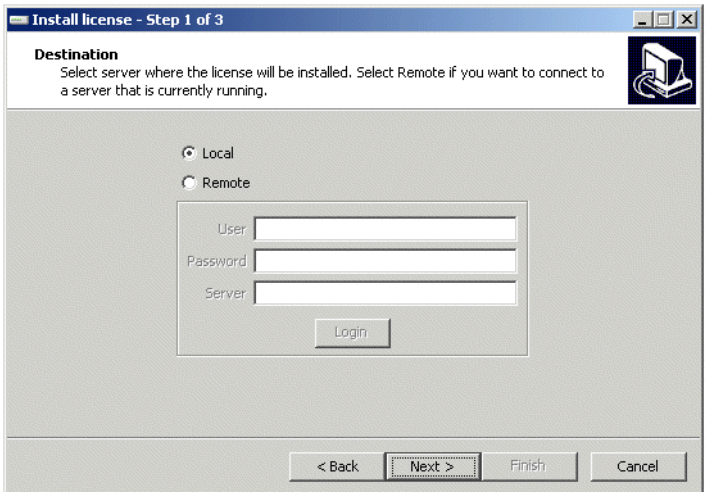


Figure 25 Install license

4. Select **Local** to install license to local computer or to view current license(s).

5. Press **Next**.

The Install License Manager wizard display possible current license(s).

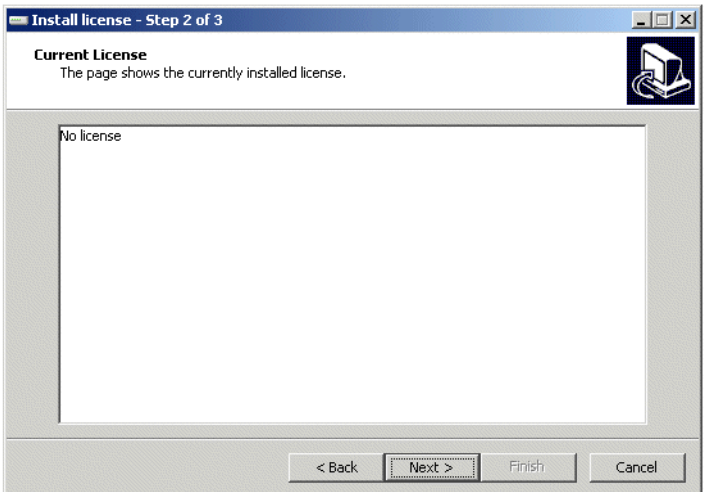


Figure 26 Install Current license

6. Press **Next** to install license(s).

7. Click **Select license** file.

8. **Browse** to license **folder.**

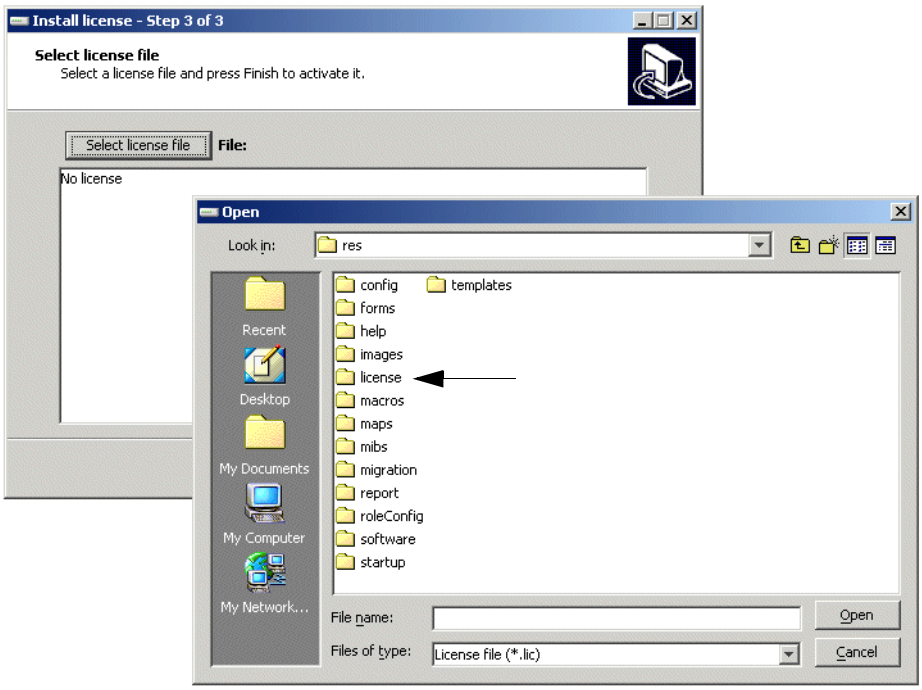


Figure 27 Install license - Select license file + file browser

9. Select **desired license file(s)** (*.lic).

10. Press **Open.**

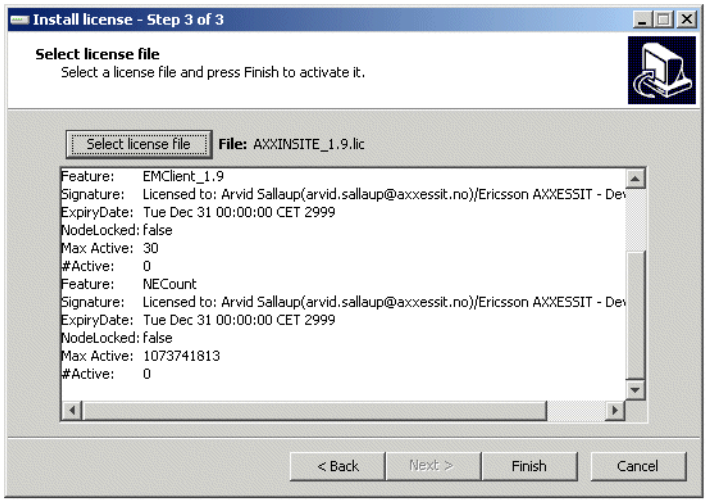


Figure 28 License Activation

11. Press **Finish** to activate selected license file(s).

2.3.1 Remote License Activation

1. Select **Remote**.
2. Enter **user id**, **password** and **server name** to install license from server to distributed client.
3. Press **Login**.

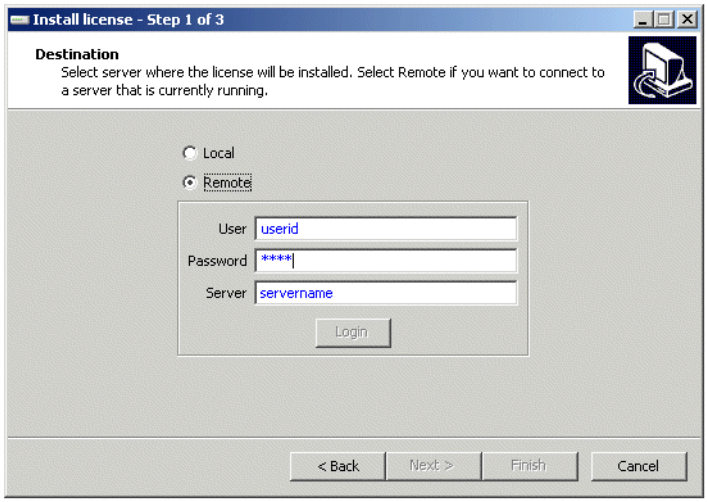


Figure 29 Install license - remote

4. Press **Next**.
5. Follow **guidelines 6- 11** on page 2-41.

3 User Administration

3.1 User Administration

This section describes how to administrate users of the AXX 9840 INSITE. User Administration is definition and management of users and their associated user profiles to:

- enable the users to do the tasks they are assigned to
- prohibit users from performing unauthorized tasks or seeing unauthorized information.

A user is defined with a user profile that defines his privileges and limitations within the system. Users can be created and removed, and their profiles are updated. Users can edit their own password. User administration is the responsibility of the system administrator. The system administrator is supported with a dedicated User Manager Tool.

3.2 Security and Access Control

User Administration implements the security in AXX 9840 INSITE. Each user is associated with a unique set of permission that controls the access to components and functionality in the system. All users are registered with user id, password, and contact information. Each defined user is associated with a role. Each role is granted a set of privileges. Multiple users can be associated with the same role.

3.2.1 User Authentication

Authentication is based on user id and password in AXX 9840 INSITE. The user identity is also used as a parameter during audit trial sessions. See “Audit Data Logging” on page 62.

3.2.1.1 Desktop

The desktop settings are associated with each user identity. They are not subject to explicit management, but are remembered by the system between each session.

The system administrator can edit the desktop settings in the User Manager. See “Update User Profile” on page 52 with reference to User Interface Configuration.

NOTE! A user keeps the desktop settings regardless of logon location.

3.2.2 User Authorization

User privileges are described in roles and user profiles:

- Roles have differentiated access to features on the Desktop.
- Users have differentiated access to network elements in the network.

See “Manage Roles” on page 54 and “User Profile” on page 48.

3.2.2.1 Domain

Logical grouping of managed objects to be accessed.

See “Manage Domains” on page 19.

3.3 Definitions

3.3.1 Role

The role represents a pre-defined set of privileges and permissions within the system, and may be viewed as a privilege and permission template. See Table 1 for more details. Each user can be assigned to one role.

System Administrator
This role has all privileges.

Network Administrator
This role has privileges such as network configuration, MCN set up and map design.

Operator
An Operator has privileges for provisioning, acknowledging and commenting notifications.

Guest
This role has only view privileges. Please see Table 1 on page 57 for details .

3.3.2 User

A user is a System Administrator, Operator , Network Administrator or Guest.

System Administrator
This user of the system is responsible for management of the management system itself and for setting up the environment for daily management of the managed network.

NOTE! There must always be at least one system administrator

Operator
Daily user of the system for management of the managed network that carries user traffic.

Network Administrator
This user is responsible to Manage Domains and Network Discovery.

Guest
Read permission only.

3.4 User Manager Tool

The System Administrator is guided with text and picture presentations in The User Manager Tool.

User Manager presents all currently registered users and sessions of the system.

Procedure: Open User manager

1. Open **User Manager** from Tools.

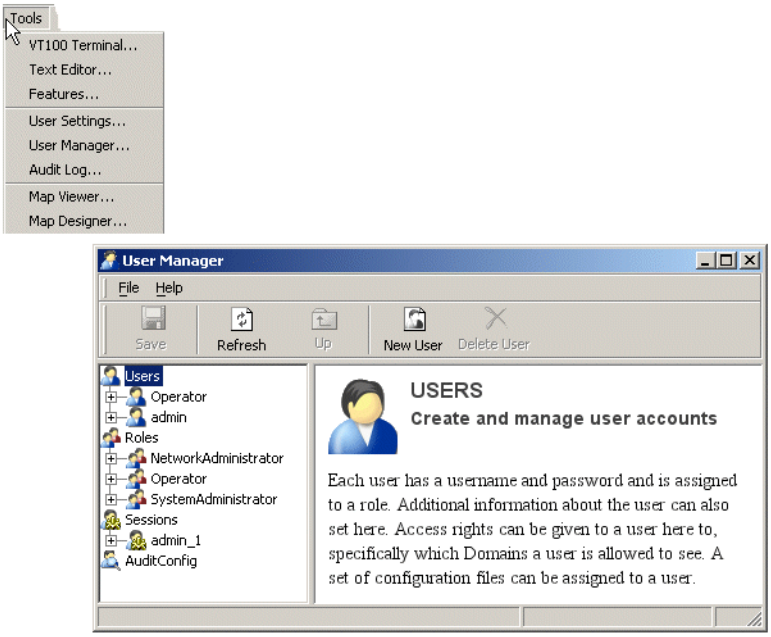


Figure 30 User Manager Tool

3.5 Manage Users

3.5.1 User Profile

A user profile is set of privileges and permissions uniquely assigned to a user. The profile is specific per instance of user, but may use templates and default values common to all users.

The configuration files defines which applications and functionality the users have access to. See “Audit Configuration” on page 63.

3.5.1.1 User profile parameters

User Identification:
A unique name to identify the user in the system

Password:
To be entered twice. The system warns if the spelling differs. Password is shown as "*" on the input field.

Role:
(default = operator)

Name:
First, Last name (OPTIONAL)

Telephone Number:
(OPTIONAL)

E-mail address:
(OPTIONAL)

Domain:
(OPTIONAL)

Procedure: Create New User

A new user of the system must be registered with an account and user profile.

1. Start **User Manager** from Tools.
2. Press icon **New User** from User Manager tool bar.

The system opens a dialog box.

Figure 31 Create a new user account

3. Enter **Userid.**
4. Enter **Password.** (Minimun six characters)

5. Re-enter **Password**

6. Click **OK.**

The system adds the new user account to User list and opens User settings.

7. Choose **Role** from pull- down menu in User information.

The system change role from Operator (default) to selected role, and add default settings to pre-defined role.

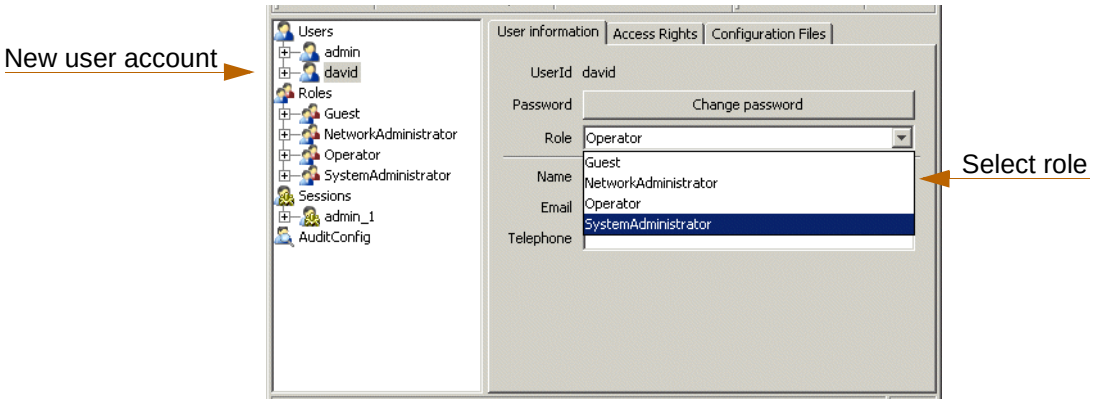


Figure 32 New user and role pull-down menu

8. Enter **Name and contact information** as User information.

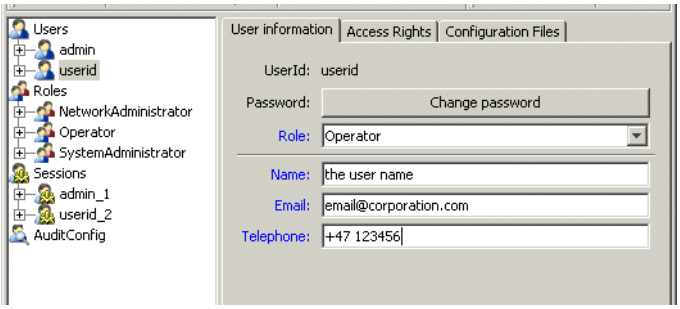


Figure 33 User information- example

9. Select **Access Rights** pane.

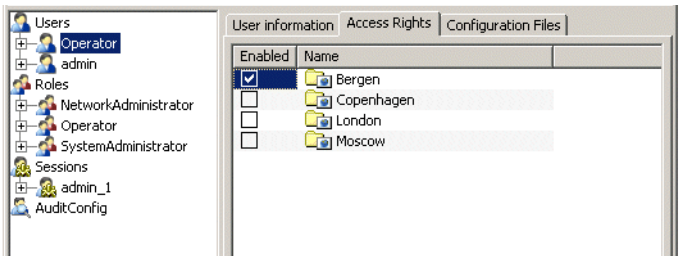


Figure 34 Access rights- example

10. Check desired **Domain** for network access (optional).

See “Manage Domains” on page 19.

11. Select and view **Configuration files** pane (optional).

The list is populated by the use of the system and show the User Interface Configuration.

12. Press **Save** from tool bar in User Manager.

Access rights not changed for currently logged-on users

TIP! If access rights for a currently logged-on user are modified, the new settings are not immediately in effect. User must log off system. The new privileges will become active next time the user logs on.

Procedure: Update User Profile

The system administrator can at any time update the user profile, except the User identification.

NOTE! The user ID cannot be edited. If there is a requirement to change the user ID, the user account can be deleted and a new user with the required user ID is re-created by the system administrator.

1. Select **user account** from User List.
2. View the **user profile.**
3. Update **user name, role, contact information** in User Information.
4. Edit **access rights.**
5. View **User Interface Configuration** files.

View shows configuration specific for that user account.

6. Add and/ or remove **configuration files** to modify user interface.

The system updates the configuration for that user account.

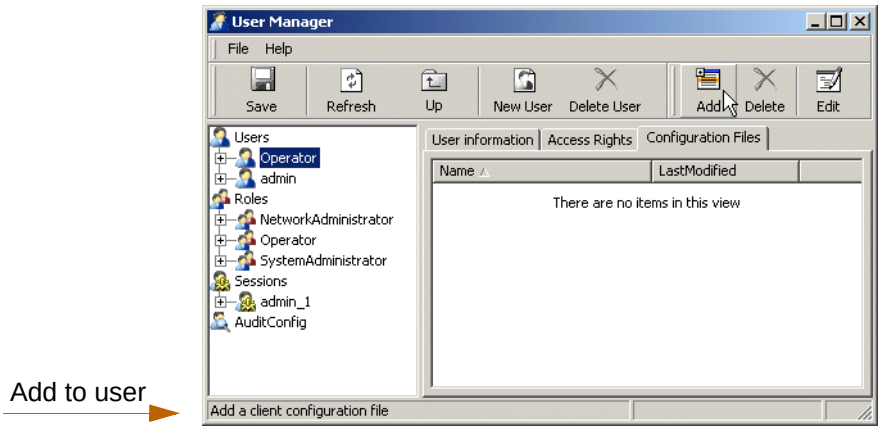


Figure 35 Configuration files

7. Click **Save.**

The system updates the user profile.

If the user is currently logged on the system, the new profile will take effect as soon as the changes is committed by the system administrator.

The user itself can also update a subset of the user profile parameters, see “Edit User settings” on page 5.

Procedure: Change Password

The system administrator may have to change a user password. The user may forget it. The system administrator do not have to know the old password in order to create a new password to user, see Figure 36

Figure 36 Password dialog box

The user itself is allowed to change the password, see “Edit User settings” on page 5.

Procedure: Delete User Account

The system administrator can at any time delete a user account from the list of registered user.

NOTE! The system will issue a warning message and cancel attempts to delete all system administrator users.

1. Select desired **User account** from User List.
2. Press **Delete** icon.

The system displays a warning.

Figure 37 User Manager Warning

3. Press **OK** to confirm operation.

The system removes user account from User list. The session will not terminate, but the next time the user tries to log on to the system, access will be denied. If the deleted user is currently logged on, all operations that need user identification and/or access rights will be denied by the system.

3.6 Manage Roles

The User Manager Tool lists all roles, see Figure 38 below.

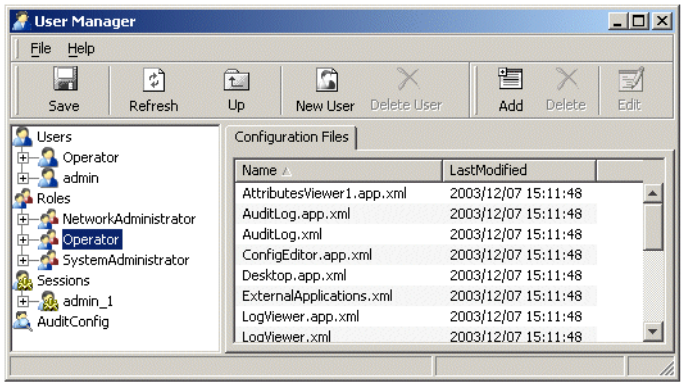


Figure 38 List role configurations- example

3.6.1 Feature Permission

The role associates users with permissions on identified system features in a feature-to-role matrix. The roles are predefined and can not be edited, see Table 1

If the feature under security management is associated with a system application, the permissions is used to control the execution of the application, the access to reading attributes and changing attributes.

If the feature under security management is associated with accessing network elements, the permissions is used to control the capability to monitor the network element, commission the network element or manage services provided by the network element.

If the feature under security management is associated with accessing the system information model, the permissions is used to control the creation, modification and deletion of objects within the model.

TIP! Filters are stored in the User Profile for each application (e.g. in AlarmHistory.app.xml). These are stored on the server. A default set for new users can be created by editing the Role configuration in the User Manager application. Consider creating a set of filters and then copy these from the current user into the configuration of the role.

3.6.1.1 Feature- to- Role Matrix

System Features				
	System Administrator	Network Administrator	Operator	Guest
Security Management				
Audit Trail report	x			
Assign Domain to users	x	x		
Manage Users	x	x (own user)	x (own user)	r
Manage Roles	x			
Define Community String	x	x		
System licence maintenance	x			
Network Management				
Define Community String	x	x		
Manage Domains	x	x	r	r
Manage Links	x	x	r	r
Network Element Management				
Module create/delete	x	x	r	r
MCN setup	x	x	r	r
NE restart	x	x	x	r
SW download (including FPGA firmware and NE feature licences)	x	x	r	r
Configuration backup/restore	x	x	x	r
Manage / un-manage network element	x	x	r	r
Port enable/disable	x	x	x	r
Bandwidth Management	x	x	x	r
VLAN Management	x	x	x	r
Notification Management	x	x	x	r
Cross Connect setting	x	x	x	r
Map Editing	x	x	r	r

System Features				
Set-up Performance Management data collection	x	x	x	
View collected PM data	x	x	x	r

Table 1 Feature- to- role matrix[1]

1. r- read permission only

x- read and write permissions

no indication - means that the application will not be visible as menu choices, or as associated attribute sets in the management tree.

3.6.2 Welcome/Overview map

The roles associates users with permissions on identified system features. The most commom tasks are available from the Welcome map.
The Welcome and Overview maps and can be customised for each role, according to the set of task/features defined.

Procedure: Customise User desktop Welcome/Overview map

1. In the User Manager **select** desired **Role**

Double-click MapDefaults.env.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<maps>
  <welcome value="../res/maps/Welcome.map"/>
  <default value="../res/maps/Error.map"/>
  <default value="../res/maps/Overview.map"/>
</maps>
```

2. **Edit location**/map **name** of Welcome/Overview map.

3. Press **Save**.

All users assigned to this role, will now see the Welcome/Overview map as configured in previous steps.
Please see MapDesigner User Guide for an example of how to customise the Welcome map.

Warning!

Rename Welcome map if editing the inital map.

TIP! Name the different Welcome maps with a role-suffix, e.g Welcome_Oper.map, Welcome_NetAdm.map.

3.7 Session

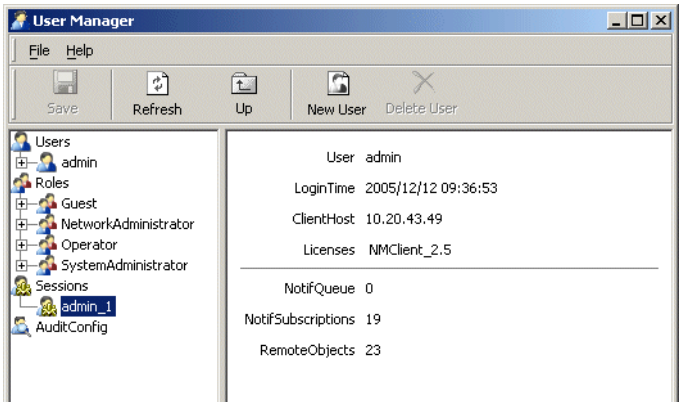
Procedure: List Users

User Manager list the currently running sessions on the server you are logged on to. See Figure 39

System parameters per user session:

- User id and associated information
- Time and date for client logon
- Client host address
- Associated licenses to client
- Possible Notification Queue (debug/ system information)
- Notification Subscriptions (debug/ system information)
- Remote Objects (debug/ system information)

See •to display parameters per user session.



- List and display parameters per user session

Procedure: End Session

Each session can be terminated from the User Manager.

NOTE! System administrator can not end its own session.

1. Start **User Manager** from desktop.
2. Select **Session** and locate user session.
3. Right- click **selected user session.**

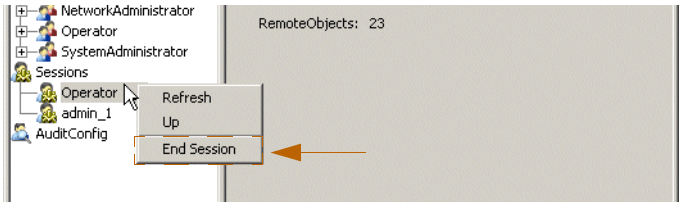


Figure 39 End session

4. Choose **End Session.**

The system will terminate the user session.

3.8 Audit Trail

The system is able to log system events and user actions such as notification handling and operations to a persistent storage on the server where the operation was performed.

The system logs a predefined set of user operations, see Table 2

The System administrator can toggle logging for some types of actions, while others are always logged, see “Audit Configuration” on page 63.

The logged information will serve analysis and reports, see “Audit Log Tool” and “Create Reports” on page 64.

For persistent storage, see “Datastore” on page 86.

3.8.1 Audit Data Logging

The following user actions are subject to audit trail:

Action/Event	**Default logging (On/Off)**	**Administrator controlled (Yes/No)**
Installation/activation of system, upgrades/updates	On	No
Installation/activation of new system components, licenses	On	No
Create/delete/modify users and roles	On	No
Audit trail analysis and reporting	On	No
Client log-on/-off	On	Yes
Security issues: attempts to violate user privileges	On	No
NE restart/reset	On	No
SW download and configuration read/write	On	No
Create/delete of NE objects	On	No
Create/delete of groups/domains	On	Yes
Create/delete of NE modules	On	No
Port enabling/disabling	On	Yes
Bandwidth setting	On	Yes
VLAN create/delete	On	Yes
Cross-connect settings	On	Yes

Table 2 Audit trail data

3.8.2 Audit Configuration

User Manager support Audit Configuration to choose which user operations and audit trail data to be logged in Audit Log.

Procedure: Log user operations & audit trail data - Enable

NOTE! Locked functions in gray are permanent logged data, and can not be edited.

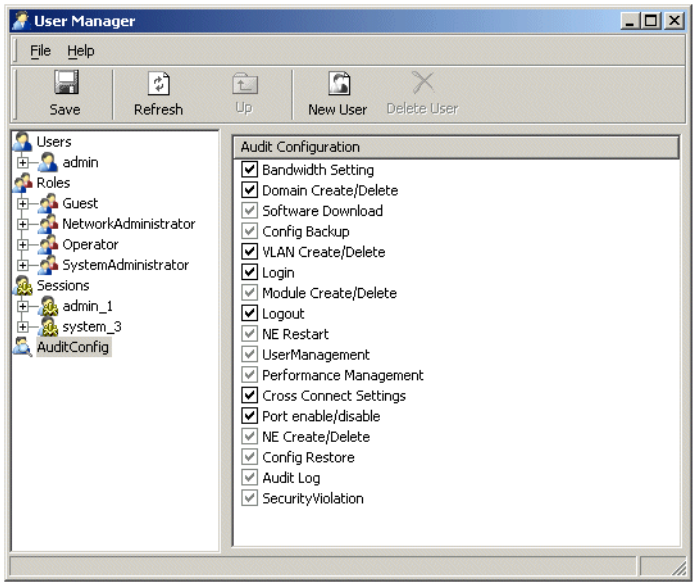


Figure 40 Audit Configuration List

1. Check desired **Function.**

Information to this function will be logged as audit data.

2. Click **Save.**

3.8.3 Audit Log Tool

Analysis of stored audit trail data is subject to user privilege control. It is possible to view log details and generate reports with Audit Log Tool.

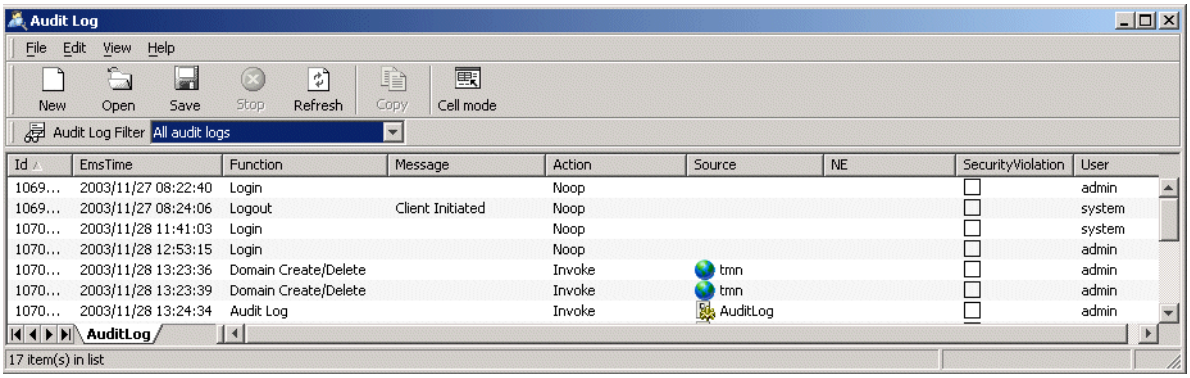


Figure 41 Audit Log Tool

Create Reports

You can generate views and reports from audit log. In Audit Log Filter you choose from a set of filter options. See page 3-64 to create a filtered list.

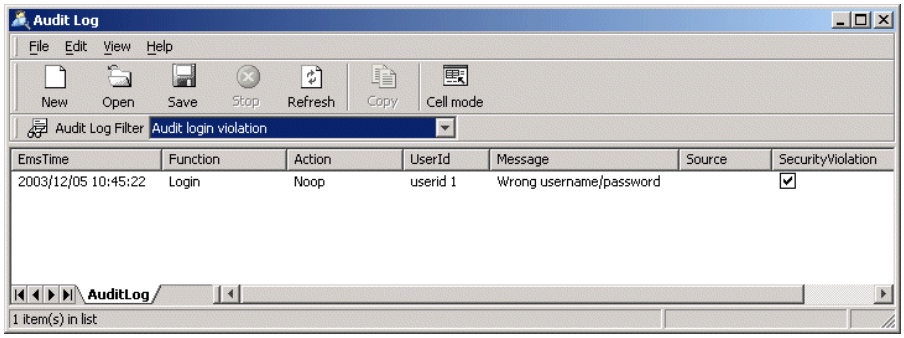


Figure 42 Audit Log filter- example

Use the Audit Log Filter to create reports, and save files to log or a local directory.

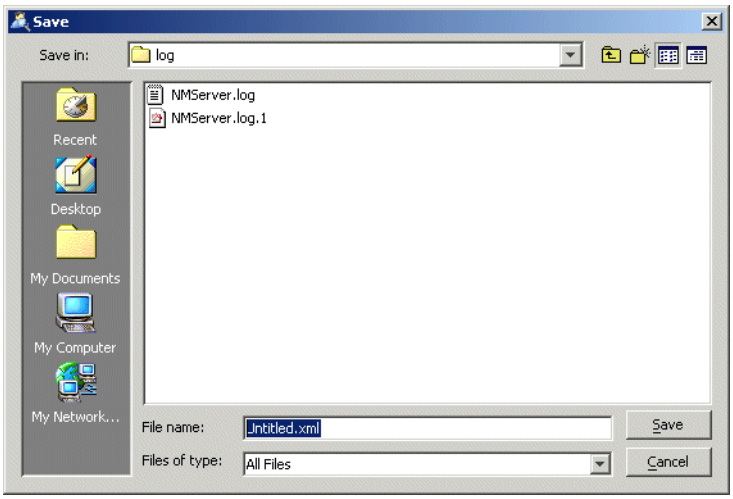


Figure 43 Audit log- save report

4 Server Administration

4.1 Server Administration

This chapter presents Server Administration and describes the tasks to start and stop the AXX 9840 INSITE server. The system administrator is supported with a dedicated Server Administration Tool. For details on client startup, see “Start program” on page 2.

4.1.1 Distributed Architecture

The system has a distributed architecture with a centralized server and distributed clients. The clients can run on one computer requesting services from the server applications on the server computer. The client and server applications can also run one the same physical computer. For more details, see “Set Up Environment” on page 2.

4.2 Server Administration Tool

Console is a server application with basic functions to support server administration. It is a command line tool (CLI). Console can be started in a non- interactive mode as well as in an interactive mode.

4.2.1 Basic Functions in Console

System administrator is supported with a selection of basic administrative functions that can be performed without the need of a client.

Functions are available from client also. See “Stop Server from Client” on page 71 and “End Session” on page 60.

1. List active sessions.

The command lists users currently logged on to the selected AXX 9840 INSITE server

To display clients, see “Session” on page 59.

2. Kill session <*session*>
The command removes the selected session and releases all associated resources and licenses.

3. Stop server.

The command performs a graceful shutdown of the AXX 9840 INSITE server. Configuration is saved and logged on clients are notified about the server shutdown.

4. Run JavaScript macro

The command runs a java script working towards java classes from the system

x. Exit

The command exit Console.

Procedure: Start Console on Windows

1. Start > Programs > AXX 9840 INSITE> **System Administration** from computer desktop.

2. Choose **Console** from program menu.

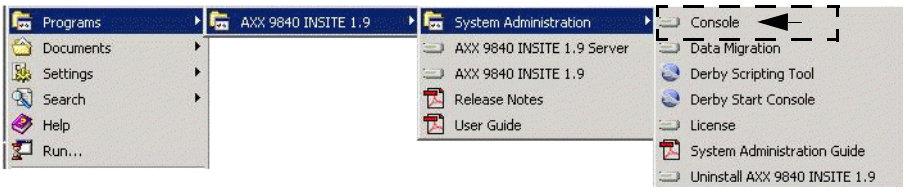


Figure 44 Start Console- Windows

3. Select AXX 9840 INSITE program and choose Console. The system presents the Console and list basic functions.

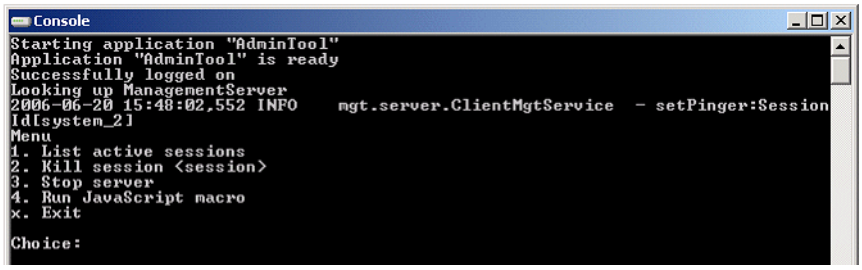


Figure 45 Console

4. Enter **function number** and press Enter. The system lists the selected parameters.

5. Enter **x** to log off the Console.

Procedure: Start the Console on Solaris

Figure 46 Start Console- Solaris

4.2.1.1 Interactive Mode

The interactive mode presents a menu- based CLI interface. When the server and Console are running on different machines, the Console will prompt the user for a user id, password and host address.

Figure 47 Interactive mode- different server- client

Procedure: Interactive Mode

1. Enter **user id**.
2. Enter **password** (default = none).
3. Enter **server** (default = local host).

The system is connected to desired server. You can now operate in the Console. See “Basic Functions in Console” on page 67.

4.2.1.2 Non- Interactive Mode

The Console is started in non-interactive mode by specifying commands on command line when starting the tool. In order to start the Console, the system will perform the authentication procedure.

- OS- user for the server startup matches the OS-user that starts the Console

NOTE! This approach requires the AXX 9840 INSITE server to be running on the same machine as the Console.

Procedure: List users and exit

This command will list current logged on users and exit Console:

```
d:\AXX9840INSITE\bin>Console -list
```

Procedure: List users and stop server

The commands may be combined as in example below:

```
d:\AXX9840INSITE\bin>Console -list -stop
```

This combination of commands will first list the active users and then issue a server shutdown.

4.3 Start and stop AXX 9840 INSITE Server

You can start server from Windows and Unix (Solaris).

The server is ready for operation when vital applications have started successfully. The system logs the result of the start up procedure. See “System logs” on page 73.

NOTE! This procedure will be the same for several of the AXX 9800 TMN product releases. You may find figures in this presentation that have another appearance then your AXX 9800 TMN product.

Procedure: Start Server on Windows

The system administrator can start the server from PC desktop.

1. Select Start > Programs > **<AXX9840INSITE>** from computer desktop.
2. Choose **Server** from the program menu.

Procedure: Start Server on Solaris (Unix)

The system administrator can start the server from a terminal.

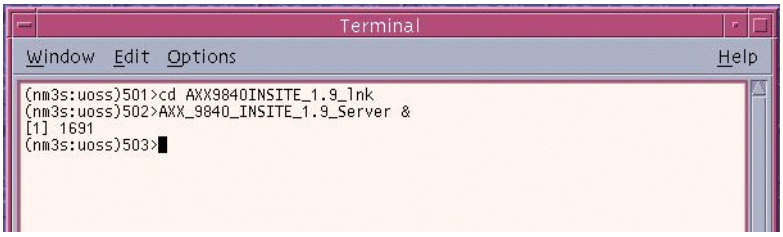


Figure 48 Start server from Solaris

Procedure: Stop Server from Client

This function is available from client as well as from Console. See “Basic Functions in Console” on page 67.

1. **Right- click** mouse from **System Management** in Management Tree.

2. Select **Shutdown** server.

The system shuts down server. Running sessions are not closed down, but applications running on client are none- operative.

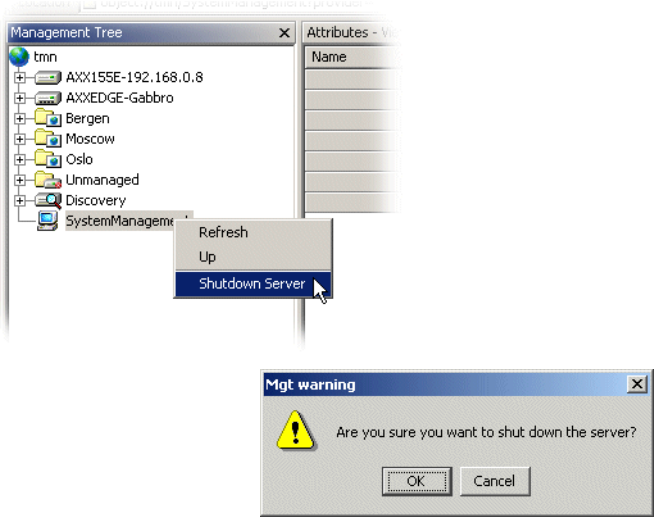


Figure 49 Stop server from Client

NOTE! The server will not wait for all clients to log off before shutting down.

TIP! Clients do not need to log off. However, the client application will need a restart to connect to server again.

4.4 System Management operation

The System administrator is offered a set of operations for system self management when logged on to the system.

4.4.1 AXX 9840 INSITE Self Management

Procedure: View Server statistics

- Read System State summary
- Read System Uptime since last restart
- Shut down server

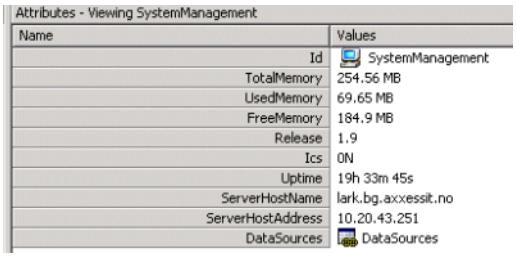


Figure 50 Server statistics view from client

4.5 Maintenance

4.5.1 Debugging

All components in the system have a debugging interface. The components can log different information decided by the debug level. See “Error log” on page 75 and “Set debug level” on page 76.

The debugging can be turned on and off in run time.

4.5.2 System logs

The system logs the result of the start up procedure, both successful and unsuccessful results. All system errors are logged.

Startup failures are part of a server log. The Error log contains error messages to operations detected in the system.

4.5.2.1 Server log

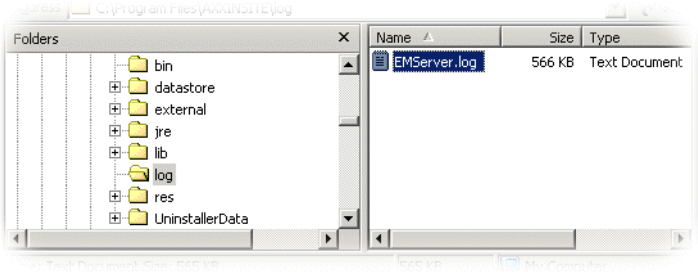


Figure 51 System log- example

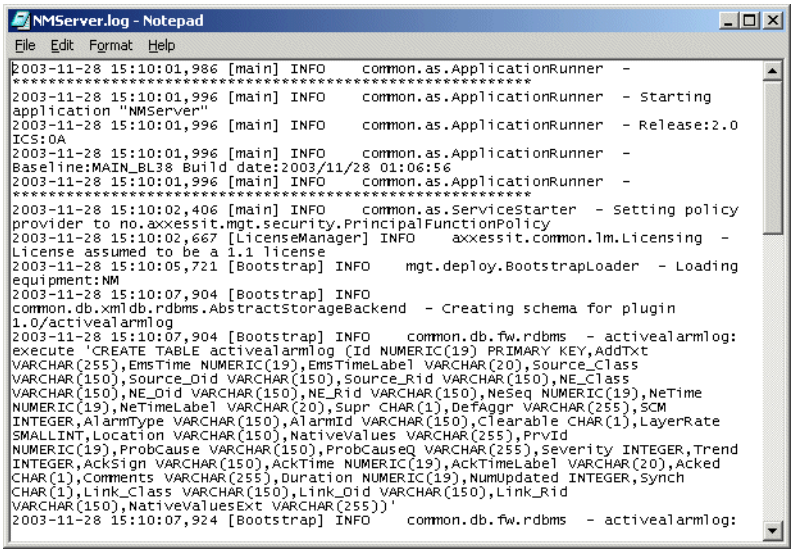


Figure 52 Server log- example

4.5.2.2 Error log

The Error log contains error messages. They are grouped into categories of Log Levels. The system error messages include:

- Severity
- Time stamp
- Text description of the message
- Category
- The module reporting the error condition
- Thread (to service (application))
- Exception (optional)
- Source (client/ server)
- User

See “Log Viewer” on page 9.

4.5.3 Set debug level

Error messages has a symbol to indicate the severity of the message. See Figure 53

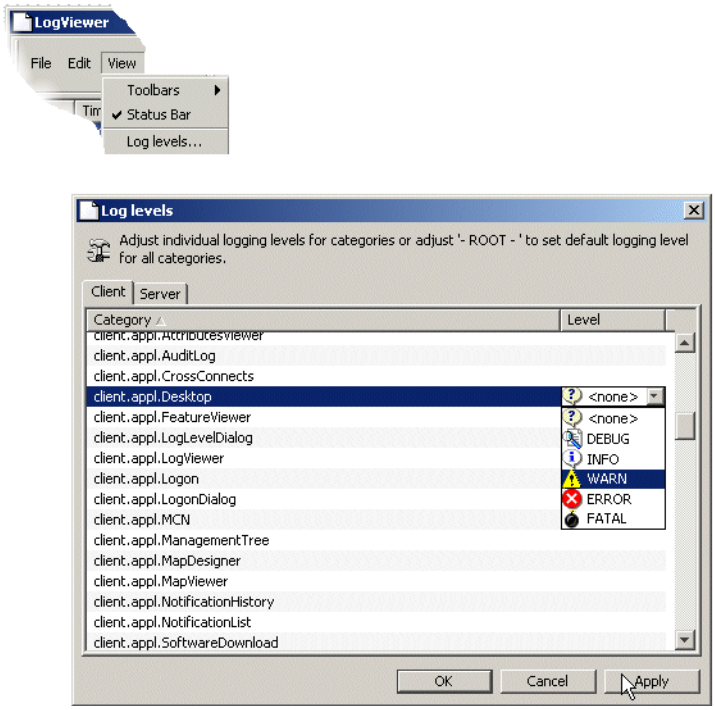


Figure 53 Log Levels

1. Select **Client** or **Server**.
2. Choose **Category of Error** message.
3. Choose desired **Level** from pull-down menu.
4. Click **Apply.**
5. **Repeat** operation.
6. Click **OK** to exit window.

The system will log message(s) according to log level(s).

Symbols are presented in severity- column in Log Viewer and in statusbar on Desktop.

4.5.4 Backup and Restore

See “Maintenance” on page 112 in Database Administration.

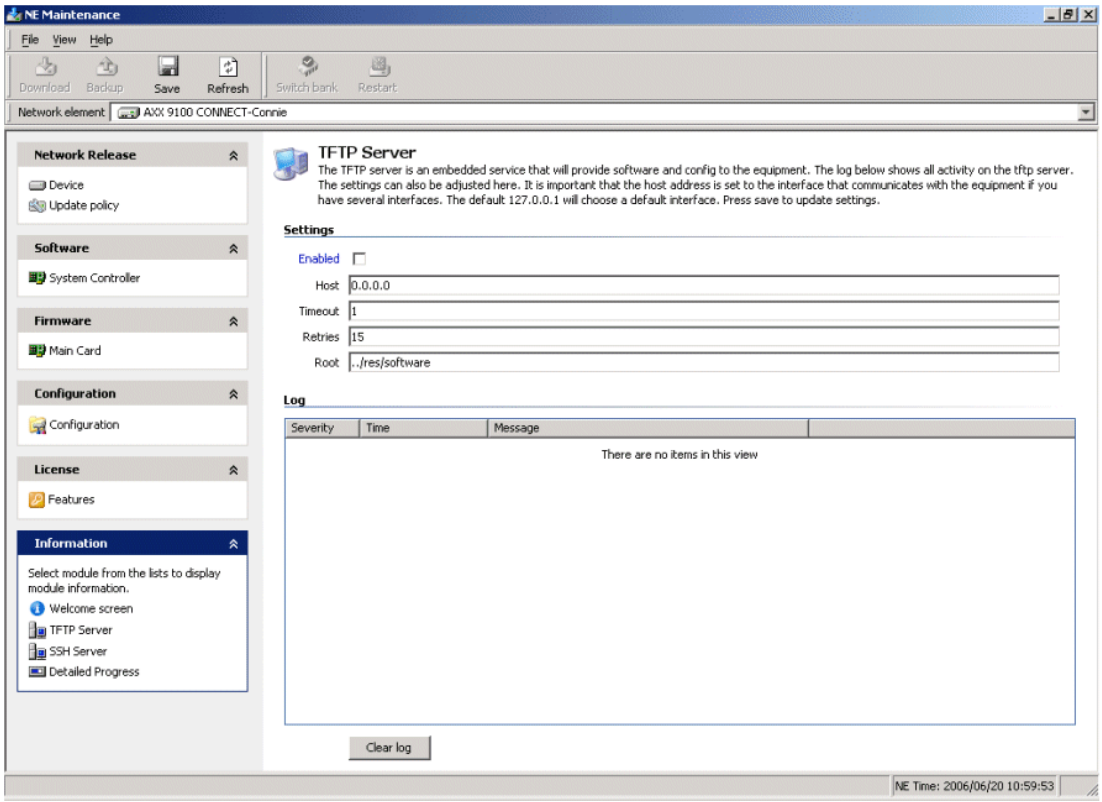
4.5.5 System Platform Security

System platform security (client and server platforms) may be implemented by standard IT security measures (firewalls, encrypted IP communication).

NOTE! Implementation of IT security measures is the responsibility of the TMN owner organization.

4.5.6 TFTP Server

TFTP Server settings are available from NE Maintenance under the Equipment menu:



host
The Ip address tftp-server host (only necessary in case of host-machines with two network addresses . The log will during Software Downolad application , display a message with info on which host-ip address that is active for the tftp-server.

retries

States how many re-transmit of packages, when timeout.

timeout

Number of seconds the tftp- server will wait for “acked” on sent package.

4.6 Troubleshooting

This section describes known incidents.

4.6.1 Unsuccessful startup

NOTE! View server log. See “System logs” on page 73. Program folders and files are located according to installation, <install>/log/*.log

4.6.1.1 The system is up and running

If applications that are not vital for normal operation were not able to start, the system reports the error yet the system is up an running.

4.6.1.2 The system is not up and running

If vital applications for normal operation did not start without failures, the system closes down and reports the error to arrearage. You must restart the system.

4.6.2 RMI Registry Port

This section describes a manually change of RMI port.

This apply to situations with several servers running on the same machine or if port is defined after installation of the system. Setting RMI port for single server is specified in installation wizard, see page -2

The TMN server requires that the TCP port **13170** is free (default). The port must be changed if this port is occupied by another process.

When the default RMI Registry port is unavailable, the server will terminate. View error message in the server log.

In the following examples, **13180** is used as the new TCP port number.

NOTE!

The dedicated line to be changed is found under "NMServer" for **AXX 9840 INSITE**

4.6.2.1 Possible error message in server log

```
ERROR   mgt.server.ClientMgtService  - Failed creating
RMI registry on port 13170
java.rmi.server.ExportException: Port already in use:
13170; nested exception is:
java.net.BindException: Address already in use: JVM_Bind
at sun.rmi.transport.tcp.TCPTransport.listen(Unknown
Source)
```

No clients, nor Console, will be able to connect to the server when this message appear in server log.

4.6.2.2 Change port to server

You can change the RMI Registry TCP port to another TCP port. Make sure that a server is not already started first.

1. **Locate** and **edit** the file from installation directory:

   ```
   ./bin/<product_version>_server.lax.
   ```

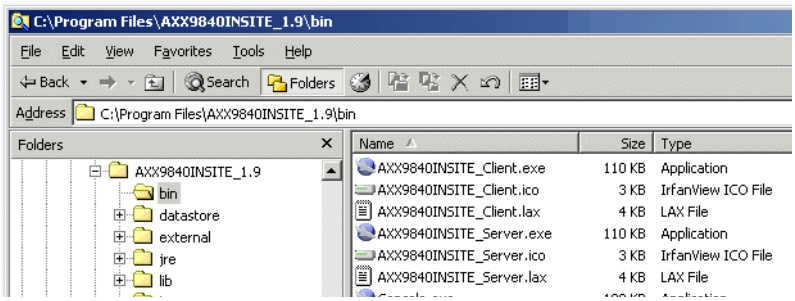


Figure 54 Directory- bin

2. **Locate** the following **lax.command.line**.args:

   ```
   _RMIRegistry.port "13170"
   ```

3. **Change** the **port number** to an available port:

   ```
   _RMIRegistry.port "13180"
   ```

NOTE! When the RMI registry port is changed to server, the clients must be notified to call that port in the login window, field ‘servername’. Example; to call server on host ‘myhost’ with RMI registry port 20000, the client must enter ‘myhost:20000’ as servername.

4.6.2.3 Change port to client

Possible logon failure

1. Start **program** and **logon client**.
2. Enter **name** of **server and port**: `<hostname>:13170`

When the logon fails due to wrong port, the error message on screen will reflect the registered port, see Figure 55

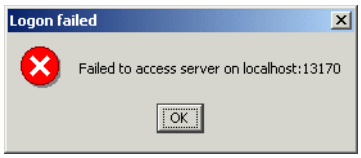


Figure 55 Client logon failure

See “Change port to client” below.

Change port to client

1. **Locate** and **edit** the file:

   `./bin/<product_version>_client.lax.`

2. **Locate** the following **lax.command.line**.args:

   `_RMIRegistry.port "13170"`

3. **Change** the **port number** to an available port:

   `_RMIRegistry.port "13170"`

The client installation is upgraded with new RMI port and access to server.

4.6.3 Allocate Memory to Server

It is possible to adjust the amount of memory available for the server application process.

1. **Locate** the **file** from directory:

   ./bin/AXX 9840 INSITE_Server.lax

2. Edit the **line** (default value indicated with blue):

   ```
   lax.nl.java.option.java.heap.size.max=536871872
   ```

4.6.4 SNMP Trap Port

Only one program can bind to the standard SNMP Trap Port, udp port 162, and on Unix, only root can bind to this port.

Please see Chapters of NE management in User Guide.

4.6.4.1 Possible error in server log

If the server program is unable to bind to the port, the following error will appear in the server log:

```
2003-12-01 15:43:45,250 [SnmpFactory] ERROR
il.cyberons.TrapReceiver  - Could not enable trap
receiving on 0.0.0.0:162
```

NOTE! The user will still see notifications in the current alarms list, but these are based on the elements notification history list, and are not dynamically updated.

4.6.4.2 Change SNMP Trap Port

1. **Before** changing the port, **investigate** which program holds the standard SNMP-trap port, and remove if appropriate.

NOTE! If you want to run the server as a normal user, not root, in Unix, or if you want to run more than one server on one machine, the SNMP-trap port that the server tries to bind to, can be changed.

2. **Edit** file:

```
res/config/services.xml
```

3. **Locate** the profile NMServer.

4. **Add** this line:

```
<property name="SNMP.trapPort"  value="1162"/>
```

The network elements are not able to send to other ports than the standard port. It is necessary to have a redirector program running that forwards packets from port 162 to other ports.

5 Administration of database

5.1 Database Support

AXX 9840 INSITE uses a Relation Database Management System (RDBMS) for persistent storage and quick and efficient data retrieval.
The system support tree databases: Oracle, MySQL and Derby (embedded solution).

5.1.0.1 AXX 9840 INSITE System requirements

The installation script gives the user the possibility to configure the system to enable the use of a database. AXX 9840 INSITE require to any database (except embedded derby database) that:

- RDBMS is created and running
- AXX 9840 INSITE database user with correct grants exists

The system is able to use Oracle , MySQL and Derby as back-end RDBMS database.

See “The Data Migration wizard” on page 29 for configuration of existing database.

5.1.1 Supported database versions

The supported database versions are:

- Oracle 8 Release 8.1.7
- Oracle 9i Release 9.2.0.7
- Oracle 10 g Release 10.2.0.2
- MySQL Database 4.0.23 or 4.1.20 with MySQL Connector 3.0.10 stable

The database system exists and is re-configured to the level required by the AXX 9840 INSITE installation script. The level of pre-configuration is dependent on the database system selected, see “Set up access to MySQL database” on page 97 and “Set Up Access to Oracle database” on page 104.

5.1.2 Datastore

This is the default location of datastorage in AXX 9840 INSITE:

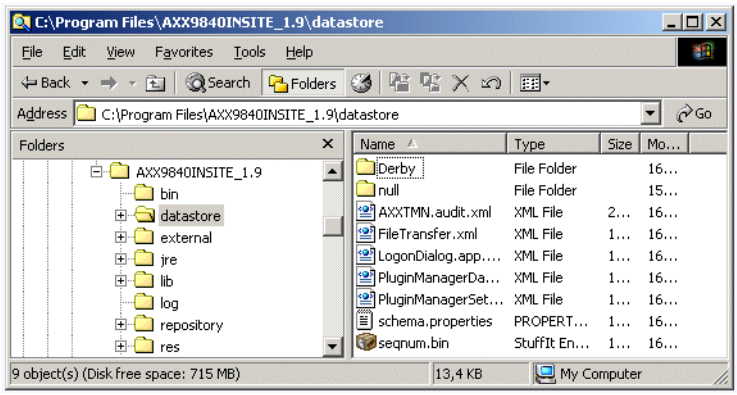


Figure 56 Datastore location- example

5.1.2.1 General Notification Management

Notifications are generated by the network elements and by services in the system itself. Notifications generated by the Network Element are stored on the network element, but the log has a limited size. All notifications from network elements and from the system itself are stored persistently in a database in the system. They are logged independent of whether a user is logged onto the system or not.

5.1.3 Log-size

Default log-size values can be changes in the file:

\res\config**persistence.xml**

When maximum log-size value is reached, the log is truncated to 90% of limit size, deleting oldest entries first.

Extract from the persistence.xml:

- <data-source name="**protectionlog**">

- <connection priority="1" uri="xmldb:axxrdbms://protectionlog" class="no.axxessit.common.db.xmldb.rdbms.DatabaseImpl">

<property name="plugin.protectionlog" value="no.axxessit.bs.log.ProtectionStorage" />

```
  <property name="plugin.protectionlog.MaxLog"
value="1000000000" />

  <property name="plugin.protectionlog.MaxRowsInResultset"
value="10000" />

...

<property name="plugin.activealarmlog.MaxLog" value="1000000"
/>

...

<property name="plugin.eventlog.MaxLog" value="1000000000" />
<property name="plugin.eventlog.MaxRowsInResultset"
value="10000" />

...

<property name="plugin.G826.MaxLog15min"
value="1000000000" />

<property name="plugin.G826.MaxLog24hour"
value="1000000000" />

<property name="plugin.G826.MaxRowsInResultset"
value="100000" />

...

<property name="plugin.pm_sample.MaxLog"
value="1000000000" />

<property name="plugin.pm_sample.MaxRowsInResultset"
value="100000" />
```

5.2 Embedded database - Derby

Derby is the embedded database in AXX 9840 INSITE. Derby is a relational database management system (RDBMS) that is based on Java™ and SQL. This database solution will support a small network of managed network elements and is also suited for demonstration and educational purposes.

Derby is a database engine written completely in Java; it will run in any certified Java Virtual Machine (JVM).

See “Copyrights And Licenses” on page 112.

5.2.0.1 Derby documentation

The following Derby documentation is available in <AXX 9840 INSITE product>**res\help** folder:

Tuning Derby , explains how to set properties to configure and tune systems,databases, specific tables and indexes, and queries. This guide also provides performance tuning tips and an in-depth study of query optimization and performance issues.

Derby tools describes how to use the tools and utilities. The tools and utilities covered in this book include: ij (scripting tool), import and export utilities, database class loading utilities, sysinfo and dblook

NOTE! The *ij* (scripting tool) is called Derby Scripting Tool in the Start menu

Derby Admin guide explains how to use Derby in a multiple-client environment. It also provides information that a server administrator might need to keep Derby running with a high level of performance and reliability in a server framework or in a multiple-client application server environment (When running in embedded mode, Derby databases typically do not need any administration).

Derby Reference Manual , provides reference information about Derby. It covers Derby's SQL language, the Derby implementation of JDBC, Derby system catalogs, Derby error messages, Derby properties, and SQL keywords.

5.2.1 Default Derby properties and log in AXX 9840 INSITE

There is a separate log for Derby under the **log** directory called **Derby.log** where Derby related information is logged.

Default properties for the Derby database are located in file /res/config/Derby/**Derby.properties**. Properties are further described in the Derby manual.

5.2.2 Derby Scripting Tool

The Derby scripting tool is Derby's interactive JDBC scripting tool. It is a simple utility for running scripts against a Derby database. You can also use it interactively to run ad hoc queries.

The tool provides several commands for ease in accessing a variety of JDBC features.

The Derby scripting tool accepts a number of different commands that let you execute SQL statements or run scripts. Each statement must end with a semicolon.

The Derby scripting tool is a JDBC-neutral scripting tool with a small command set. It can be used to access any JDBC driver and database accessible through that driver.

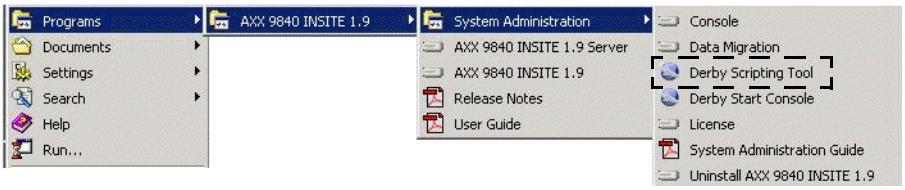
The main benefit of a tool such as Derby scripting tool is that it is easy to run scripts for creating a database schema and automating other repetitive database tasks. In addition, it accepts and processes SQL commands interactively for ad hoc database access.

Default properties for the Derby Scripting Tool are located in file /res/config/Derby/**DerbyScriptingTool.properties**. Properties are further described in the Derby manual.

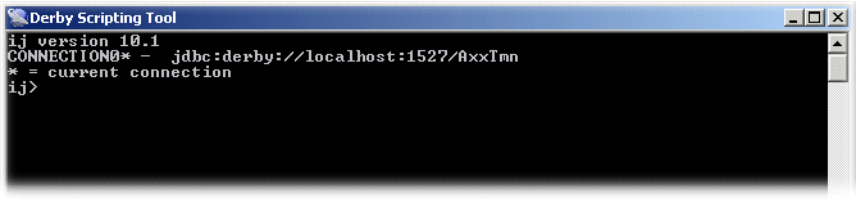
TIP! In addition to the command line tool 'Derby Scripting Tool' it is possible to use graphical JDBC enabled tools against Derby, e.g. the Squirrel SQL client, http://squirrel-sql.sourceforge.net/. This tool can also be used against MySql and Oracle.

Procedure: Launch Derby scripting tool

1. Select Programs>AXX 9840 INSITE product>System Administration>**Derby Scripting Tool**.



2. The Derby Scripting Tool console launches with the *ij*-prompt:



5.2.2.1 Commands

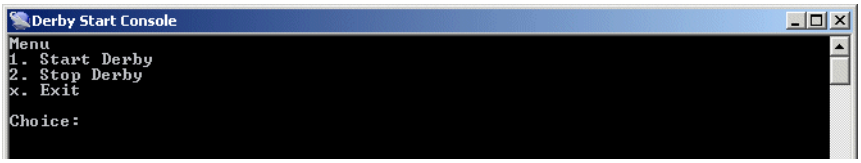
The Derby Scripting Tool accepts several commands to control its use of JDBC. It recognizes a semicolon as the end of an console or SQL command; it treats semicolons within SQL comments, strings, and delimited identifiers as part of those constructs, not as the end of the command. Semicolons are required at the end of an console or SQL statement. All console commands, identifiers, and keywords are case-insensitive. Commands can span multiple lines without any special escaping for the ends of lines. This means that if a string spans a line, the new lines will show up in the value in the string. Derby Scripting Tool treats any command that it does not recognize as an SQL command to be passed to the underlying connection, so syntactic errors in console commands will cause them to be handed to the SQL engine and will probably result in SQL parsing errors.

For commands available please see complete Derby documentation located in the res\help\Derby\tools-folder.

5.2.3 Derby Start Console

The Derby Start Console starts the Derby database in a standalone modus.The Derby Start Console is an administration tool for cases where the embedded database will not start due to errors in the TMN server.

This console tool has the following menu:



NOTE! Remember to stop the database before you start the server.

5.2.4 Backing up and restoring databases

Derby provides a way to back up a database while it is online. You can also restore a full backup from a specified location. While the backup is in progress, update operations are temporarily blocked, but read operations can still proceed.

5.2.4.1 Backing up a database

The topics in this section describe how to back up a database.

Procedure: Offline backups

To perform an offline backup of a database, use operating system commands to copy the database directory. You must shut down the database prior to performing an offline backup.

For example, on Windows systems, the following operating system command backs up a (closed) database that is named sample and that is located in d:\mydatabases by copying it to the directory c:\mybackups\2005-06-01:

```
xcopy d:\mydatabases\sample c:\mybackups\2005-06-01\sample /s
    /i
```

If you are not using Windows, substitute the appropriate operating system command for copying a directory and all contents to a new location.

NOTE! On Windows systems, do not attempt to update a database while it is beingbacked up in this way. Attempting to update a database during an offline backup will generate a java.io.IOException. Using online backups prevents this from occurring.

For large systems, shutting down the database might not be convenient. To back up a database without having to shut it down, you can use an online backup.

Procedure: Online backups

Use **Derby Scripting Tool** to back up a database while it is running. During the interval that the backup is running, the database can be read, but writes to the database are blocked. You can perform online backups by using the backup procedure or by using operating systems commands with the freeze and unfreeze system procedures. Using the backup procedure to perform an online backup:

The SYSCS_UTIL.SYSCS_BACKUP_DATABASE procedure locks the database so that any connection trying to write to the database will be frozen until the backup completes. Database reads can continue while the backup is running. The SYSCS_UTIL.SYSCS_BACKUP_DATABASE procedure takes a string argument that represents the location in which to back up the database. Typically, you provide the full path to the backup directory. (Relative paths are interpreted as relative to the current directory, not to the derby.system.home directory.)

For example, to specify a backup location of c:/mybackups/2005-06-01 for a database that is currently open, use the following statement (forward slashes are used as path separators in SQL commands):

```
CALL SYSCS_UTIL.SYSCS_BACKUP_DATABASE('c:/mybackups/2005-06-
      01')
```

The SYSCS_UTIL.SYSCS_BACKUP_DATABASE() procedure puts the database into a state in which it can be safely copied, then copies the entire original database directory (including data files, online transaction log files, and jar files) to the specified backup directory. Files that are not within the original database directory (for example, derby.properties) are not copied.

Uncommitted transactions do not appear in the backed-up database.

It is also possible to pass commands into Derby Scripting Tool from an input file from command line:

E.g. \bin>DerbyScriptingTool.exe backup.txt

NOTE! Do not back up different databases with the same name to the same backup directory. If a database of the same name already exists in the backup directory, it is assumed to be an older version and is overwritten.

5.2.5 Restoring a database from a backup copy

Procedure: Restore the database using a backup copy

To restore a database by using a full backup from a specified location, do as follow:

1. Shut down the AXX 9840 INSITE server.
2. Enable the restoreFrom property in the property file /res/config/Derby/**DerbyRestore.properties.** Delete the '#' character and specify the backup location and save the file. If no

database already exists, specify createFrom instead of restoreFrom.

3. Start the AXX 9840 INSITE server. The database will be restored during the start-up.

NOTE! If a database with the same name exists, the system will delete the database , copy it from the backup location, and then restart it.

5.3 MySQL

5.3.1 MySQL database requirements

The following steps are required to install and run database in the system:

Procedure: Download JDBC driver from Database Vendor

Ericsson AXXESSIT cannot distribute third party driver-files, thus you must visit the database vendor internet site**.**

1. Visit **Database vendor internet site**:

http://dev.mysql.com/downloads/connector/j/3.0.html

2. **Download JDBC driver** (pick a mirror)
3. **Unzip** mysql-connector-java-3.0.10-stable.zip
4. The jar file **mysql-connector-java-3.0.10-stable-bin.jar** is applicable for both Windows and Solaris.
5. **Place this file** on a directory **accessible** from the **AXX 9840 INSITE server.**

6. Make sure that this file is referenced in the **Select MySQL JDBC Driver** :

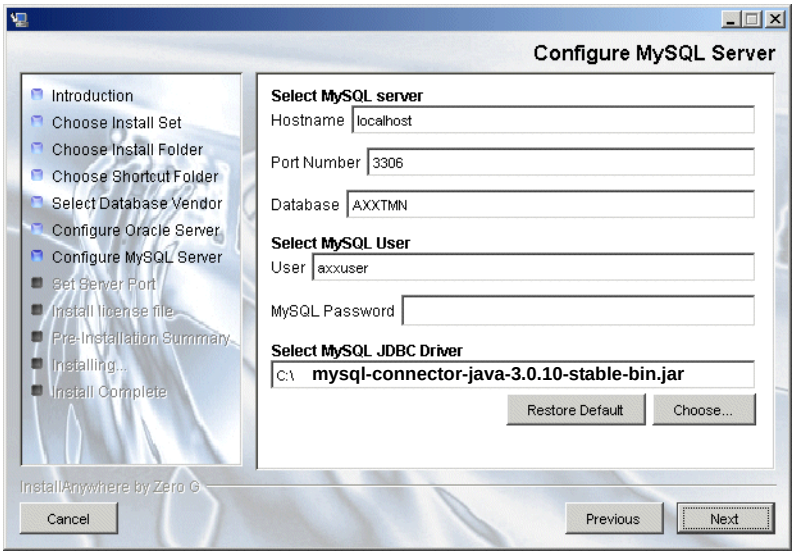


Figure 57 Example - Reference to location of downloaded JDBC driver

NOTE! The mysql-connector-java-3.0.10-stable-bin.jar is copied to <AXX 9840 INSITE_installdir>/lib/ext and given the name mysql-connector.jar

NOTE! If updating the JDBC Driver later, make sure that the filename and location is referenced correctly as shown in Figure 57

5.3.1.1 Create MySQL user and give grants to MySQL user,

Procedure: Example - User creation script

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\mysql\bin>mysql --user=root --password=mysql;
Welcome to the MySQL monitor.  Commands end with ; or
\g.
Your MySQL connection id is 3 to server version:
4.0.23-nt

Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.

mysql> CREATE DATABASE AXX 9840 INSITE;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON AXX 9840 INSITE.* TO
axxuser@localhost IDENTIFIED BY 'axxpw' WITH GRANT
OPTION;
Query OK, 0 rows affected (0.09 sec)

mysql> GRANT ALL PRIVILEGES ON AXX 9840 INSITE.* TO
axxuser@"%" IDENTIFIED BY 'axxpw' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
Bye

C:\mysql\bin>
```

Procedure: Set up access to MySQL database

The selected database system can be accessed by the installation wizard during execution of the installation script.
When AXX 9840 INSITE starts it will connect to the database and check if the necessary database tables exist. If not, the system will make tables.

1. Follow guidelines in installation.
2. Select **MySQL** as your Database Vendor.

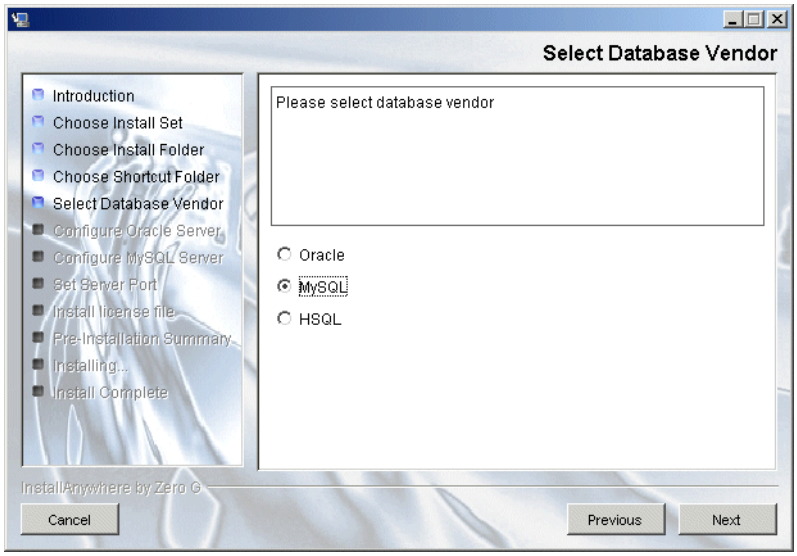


Figure 58 Select MySQL database

3. Click **Next**, see Figure 59 .
4. Select **MySQL server**.
5. Enter **Hostname**, **port** and **Database**.
6. Select **MySQL User**.
7. Enter **User** and **Password**.
8. Select **MySQL JDBC Driver**.

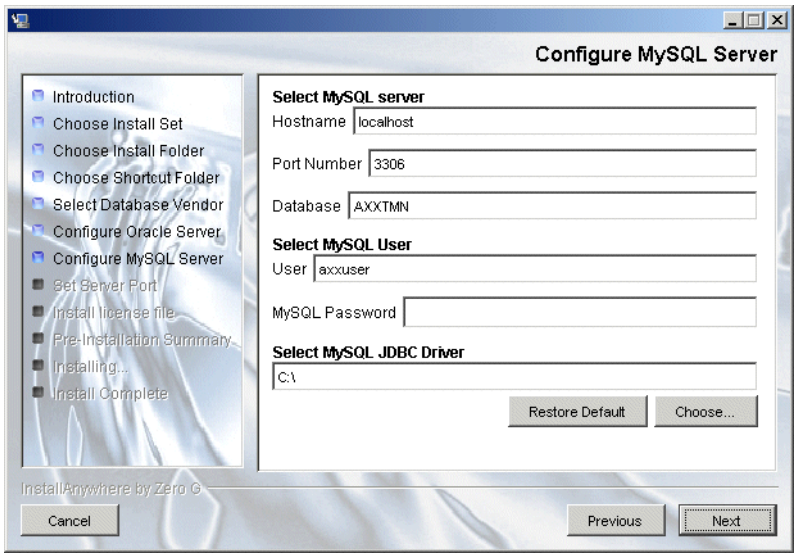


Figure 59 Configure MySQL- default view

9. Click **Next**.

10. Follow guidelines in installation.

5.3.2 MySQL Installation

5.3.2.1 Example

This example shows step by step, how to install **AXX 9840 INSITE** with an external database, MySQL.

Procedure: Download/install

1. From http:\\www.mysql.com, download **`mysql-4.0.23-win.zip`** and unzip the file.

2. Choose Destination Location -
   in this example; `c:\program files\mysql.`

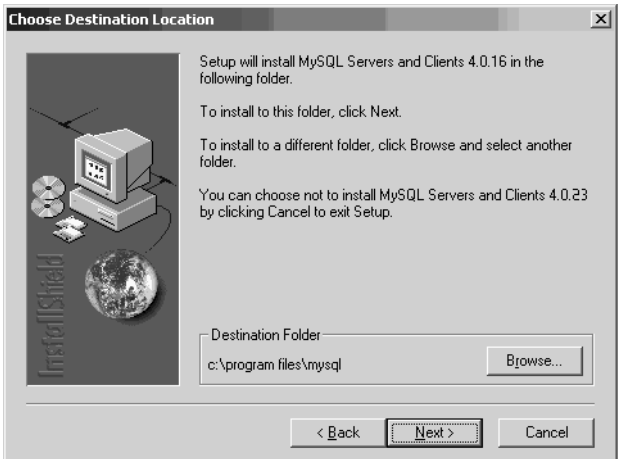


Figure 60 Mysql - Installation wizard - destination folder

3. Select "Typical" installation.

Procedure: Create a new database and user

1. After completed installation, start up the Windows command interface "cmd.exe".

Go to the **`mysql\bin`** directory - in this example: `c:\program files\mysql\bin`

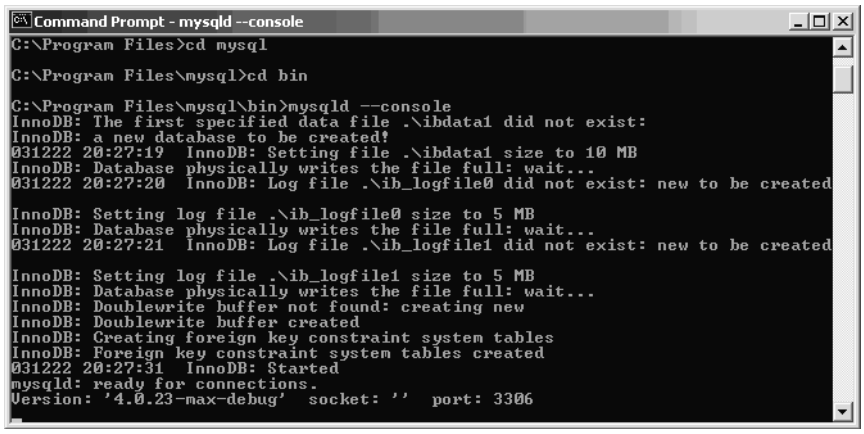


Figure 5-1. Windows CMD - change directory

2. Start up `c:\program Files\mysql\bin\`**`winmysqladmin.exe.`**

3. Check the existing databases using WinMySQL admin .

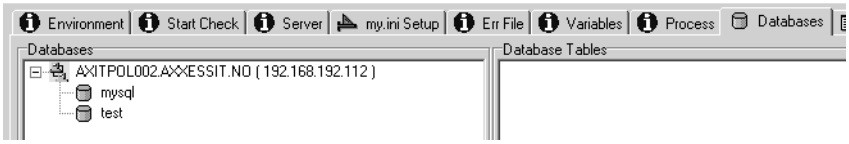


Figure 6 WinMySQL admin - existing databases

4. In mysql, **create a new database and user** for **AXX 9840 INSITE** (description how to do you can find in ""Example - User creation script" on page 96.

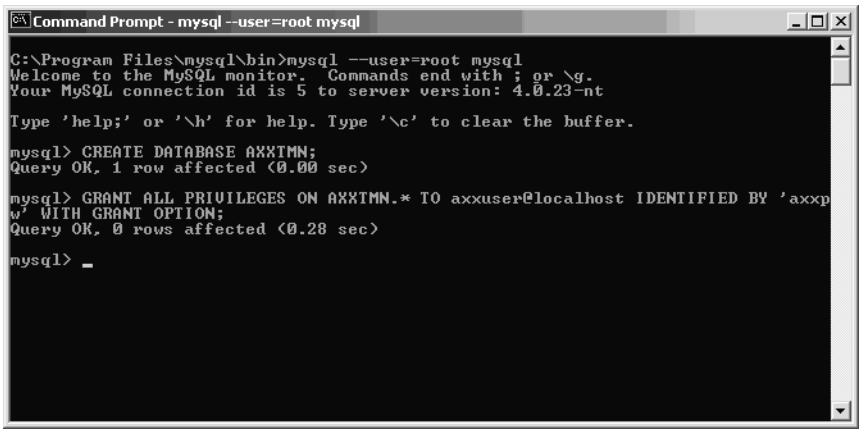


Figure 7 MySQL- create database

Procedure: Verify new db

1. You can check that database is existing in system by quick from this window - command: **\q**
   Then go again to mysql an enter the following command:

   mysql -D AXX 9840 INSITE (log on into AXX 9840 INSITE database).

2. You can also see new database in WinMySQL admin .If you don't have the same picture like below, you should "shut down" the "WinMySQLadmin" and launch again.

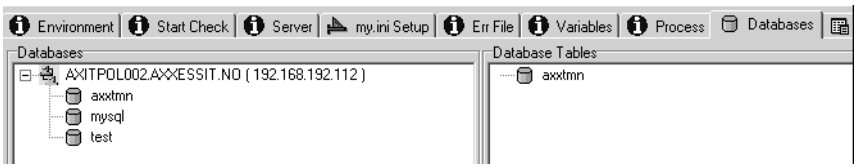


Figure 8 WinMySQL admin - verify new database

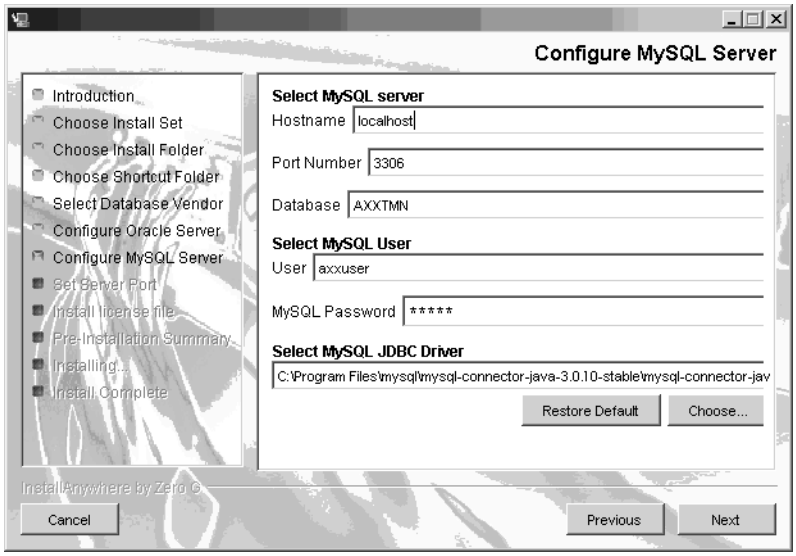
In "AXX 9840 INSITE" database you can't find any table - the tables will be created in the first time when you start up **AXX 9840 INSITE**.

Procedure: Configure MySQL Server

1. **Download** from http:\www.mysql.com; **`mysql-connector-java-3.0.10-stable.zip`**. Unzip the file to any directory (suggestion to mysql directory: **`c:\Program files\mysql`**)

2. **Install AXX 9840 INSITE** with MySQL database.

3. During the installation process in "Configure MySQL Server" **type** MySQL **password**; in this example axxpw.



4. **Launch** the "**AXX 9840 INSITE Server**".

Procedure: Verfy database tables

1. **Check** the database - now should exist in database tables. You can check by WinMySQLAdmin,

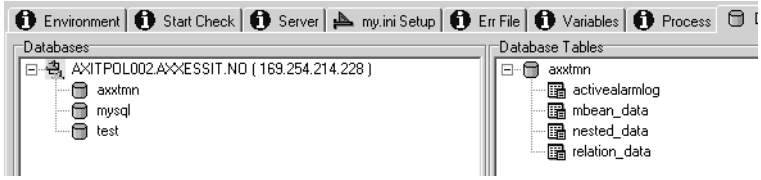


Figure 5-1. WinMySQL admin - verify new tables

or by SQL command:
SHOW TABLES;

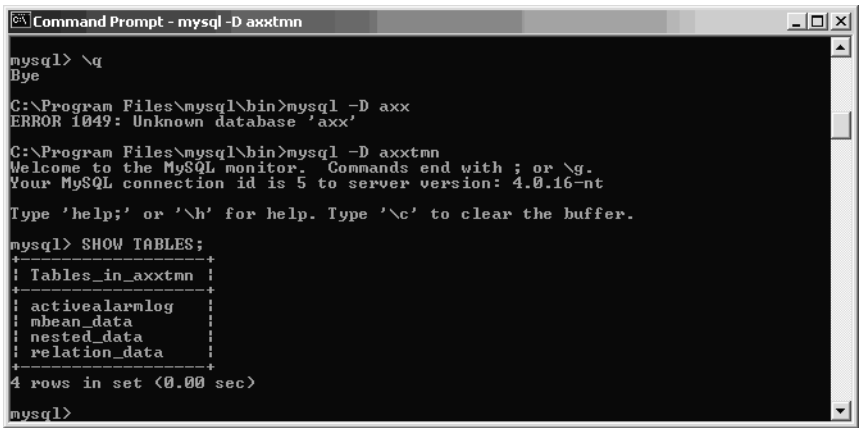


Figure 6 CMD - verify new tables

This is not all tables - the server will create more tables when needed - for example :
alarmlog (in this table will be all alarms)
eventlog

2. To see **records from table** you can send followed query to database:

```
SELECT * FROM alarmlog;
```

3. To see **how many records** you have in table, the query will be:

```
SELECT COUNT(*) FROM alarmlog;
```

5.4 Oracle

5.4.1 Create Oracle user and give grants to Oracle user

The following parameters are required to install and run database in the system:

Procedure: User creation script

```
oracle@axitblis010:~$ sqlplus /nolog

SQL> connect / as sysdba
Connected.
SQL> create user axxuser identified by axxpw;

User created.

SQL> grant resource, create session to axxuser;

Grant succeeded.

SQL> quit;
Disconnected from Oracle9i
oracle@axitblis010:~$
```

Procedure: Set Up Access to Oracle database

This example describes the tasks to set up the database together with the installation of AXX 9840 INSITE. It is a pre-requisite that an Oracle SID exists. Vendor specific tools (Database Configuration Assistant) or sql scripts can be used for this purpose.

The selected database system can be accessed by the installation wizard during execution of the installation script. When AXX 9840 INSITE starts it will connect to the database and check if the necessary database tables exist. If not, the system will make tables.

1. Follow the guidelines in the installation wizard.
2. Select **Oracle** as your database vendor.

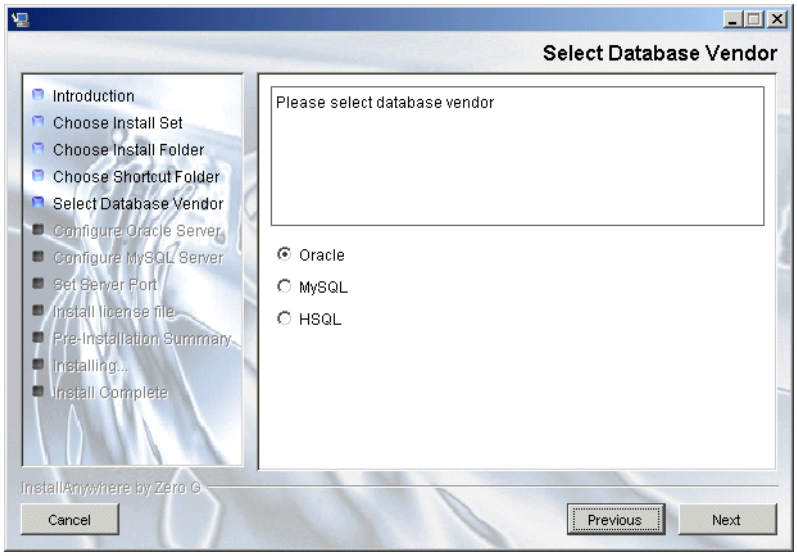


Figure 7 Select Oracle Database

3. Click **Next**.
4. Select **Oracle server**.
5. Enter **hostname, port** and **SID**.
6. Select **Oracle User**.
7. Enter **User** and **Password**.

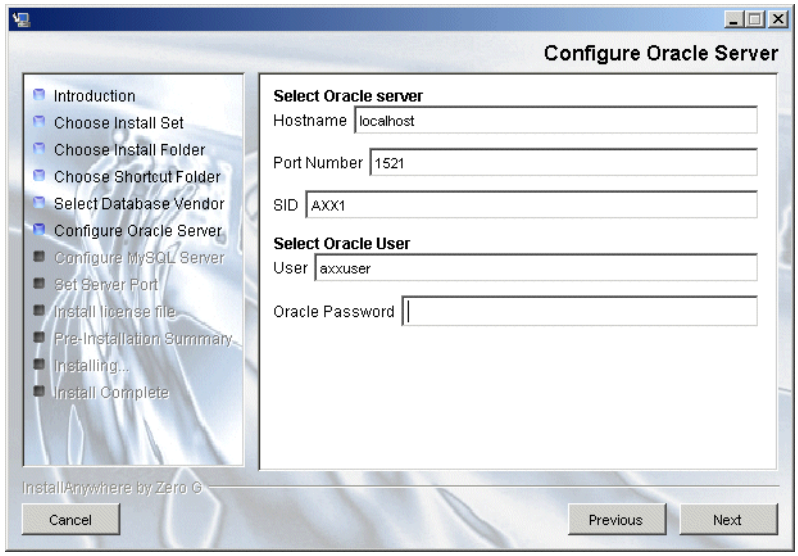


Figure 5-1. Configure Oracle Server- view default

8. Click **Next**.
9. Follow guidelines in installation wizard.

5.5 Database schemes and table descriptions

- SQL: Managed objects and relations between managed objects.
- TMN Logs: Audit, Event, Alarm.
- PM data for Performance Management (PM) Database interface.

5.5.1 Table RELATION_DATA (SQL)

This table holds information on MBean relations.

5.5.2 Tables MBEAN_DATA and NESTED_DATA (SQL)

The following tables hold information on attributes of MBeans. The field NESTED_DATA.REFID is a foreign key to MBEAN_DATA.VALUE for rows with MBEAN_DATA.TYPE equals to 'X' or 'M'.

5.5.3 Table protectionlog

Log of protection switching events.

5.5.4 Table sncs

Sub Network Connections

5.5.5 Table snc_routes

Sub Network Connection route definitions.

5.5.6 Table snmptraplog

Trap log for SNMP GW (IF installed/licensed)

5.5.7 Table sys_prop

TMN system wide properties.

5.5.8 Table EVENTLOG

This table holds all events.

5.5.9 Table ACTIVEALARMLOG

This table holds all active alarms in the system.

5.5.9.1 Automatically remove cleared alarms.

This feature is not activated in the default configuration. The feature is called "AutoAcknowledge", and will automatically simulate an operator "Acknowledge" operation whenever an alarm "Clear" notification is received. All cleared and acknowledged alarms are removed automatically from the Current Alarm View.

The “AutoAcknowledge” feature is active for all alarm instances where you expect to receive both "alarm" and "clear" notifications from the network element ("clearable alarms", as opposed to "non-clearables", which send only an "alarm" notificaiton and never "clear").

Procedure: Activate AutoAcknowledge feature

1. In the file:

   ```
   <installdirectory>\res\config\fm\AutoAcknoledgeAXX
   9840 INSITE.autoack.xml
   ```

2. Remove the left and right "**<!--**" comment statements on the line

   ```
   <!--
   <rule condition="Clearable = 'Y' AND Severity =
   'Cleared'" signature="AXX 9840 INSITE" />
     -->
   ```

5.5.10 Table ALARMLOG

This table holds all historical alarms in the system.

5.5.11 Table AUDITLOG

This table holds all audit information.

5.5.12 Tables PM_Q, PM_D and PM_SOURCE

These tables contain all the PM data collected from the network elements:

1. The PM_Q table holds all 15 minutes G.826 PM samples and,

2. The PM_D table holds all 24 hour ITU-T G.826 PM samples.

Each row in the table contains a sample from a given managed object.

The PM_SOURCE table contains managed object references to all Managed Objects that have been subject for PM collection. The SOURCE field is a foreign key in both PM_Q table and PM_D table to the ID in PM_SOURCE table. It is safe to truncate the PM_Q table or PM_D table.

Caution!

The PM_SOURCE table should not be truncated since both PM_Q table and PM_D table reference it and it is cached in memory during runtime.

The configuration file `../res/config/DEFAULT.database.xml` describes data-types for each supported database vendor.

5.5.13 PM_Q and PM_D table

Field	Description
Id	Unique row identifier
Source	Foreign key to a Managed Object identifier
EndTime	End time for the PM sample. Number of milliseconds from 1970. The EndTimeDate field contains the same information.
EndTimeDate	End time for the PM sample.
StatusFarEnd	Character that indicates whether the sample is invalid or valid for far end samples. Value 'I' for invalid samples and 'V' for valid samples.
BBEFarEnd	Background Block Error for far end samples.
ESFarEnd	Error Seconds for far end samples.
SESFarEnd	Severely Error Seconds for far end samples.
UASFarEnd	Unavailable seconds for far end samples.
StatusNearEnd	Character that indicates whether the sample is invalid or valid for near end samples. Value 'I' for invalid samples and 'V' for valid samples.
BBENearEnd	Background Block Error for near end samples.
ESNearEnd	Error Seconds for near end samples.
SESNearEnd	Severely Error Seconds for near end samples.
UASNearEnd	Unavailable seconds for near end samples.
Cbklm	SDH CBKLM value

5.5.14 PM_SOURCE

Field	Description
Id	Unique row identifier
Source_Class	Managed Object class.
Source_Oid	Managed Object identifier.
Source_Rid	Managed Object resource id, e.g. ip address.
Slot	The slot where the Managed Object is located.
Port	The port where the Managed Object is located.
NE_Class	Network element class that contains the Managed Object.
Source_Label	Name on source object (displayed)
NE_Label	Name on NE object (displayed)

5.5.15 PM_SAMPLE

Field	Description
Id	Unique row identifier
Source	Foreign key ot a Managed Object Identifier
Type	Sample type. Currently only Bandwidth Utilization samples.
EmsTime	End time for the PM sample. Number of milliseconds from 1970. The SampleDate field contains the same information.
SampleDate	End time for the PM sample.
Value	Sample value
SampleInterval	Sample interval

5.6 Maintenance

5.6.1 Notification Repository Maintenance

The system offers facilities to export and import notifications stored in a persistent notification repository. This facility is provided to prevent the notification repository to overflow the available memory and disk space available in the system platform.

5.6.2 Backup

Please see Database Vendor for backup- procedures

TIP! A standby server can be used to ensure database redundancy. The clients must do a new logon.

5.7 Copyrights And Licenses

5.7.1 Derby

Apache Derby / Apache License, Version 2.0

The AXX 9840 INSITE includes software developed by:

The Apache Software Foundation (http://www.apache.org/).

Portions of this software were developed at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign.

This software contains code derived from the RSA Data Security Inc. MD5 Message-Digest Algorithm, including various modifications by Spyglass Inc., Carnegie Mellon University, an Bell Communications Research, Inc (Bellcore).

Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England. The original software is available from ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/

5.7.2 Hypersonic SQL

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes Hypersonic SQL."

- Products derived from this software may not be called "Hypersonic SQL" nor may "Hypersonic SQL" appear in their names without prior written permission of the Hypersonic SQL Group.

- Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes Hypersonic SQL." This software is provided "as is" and any expressed or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Hypersonic SQL Group or its contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption). However caused any on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

This software consists of voluntary contributions made by many individuals on behalf of the Hypersonic SQL Group.

5.7.3 Trademark Credits

Apache Jakarta and XML: DB API

This product contains Apache Jakarta and XML components: commons, log4j, POI, Xalan, Xerces:

Apache Jakarta: http://www.apache.org/foundation/licence-FAQ.html)

XML:DB API: http://www.xmldb.org.

Install Anywhere

This product uses InstallAnywhere version 6.1 Enterprise

Zero G, Zero G Software, ZeroG.com, the Zero G logo, InstallAnywhere, the InstallAnywhere logo, PowerUpdate, PowerUpdate.com.

The PowerUpdate logo, LaunchAnywhere, and SpeedFolder are trademarks or registered trademarks of Zero G Software, Inc.

Oracle: Oracle 8*i, 9i, 10g* - Oracle Corporation

MySQL: MySQL version 4.0.23/4.1.20

Windows: Windows NT/ XP - Microsoft Corporation

Solaris, Java and Sun Microsystems

6 Firewall

6.1 Using AXX 9840 INSITE with Firewalls

AXX 9840 INSITE is compatible with filtering router firewalls, NAT (network address translation) firewalls, and SSH tunnels. This chapter is inteded for System Administrators and Network Administrators, and describes the configuration of clients , servers and network.

6.1.1 Components

The components that needs to be separated and have well defined

properties, are these:

- Client
- Server
- Database
- Managed Network Elements

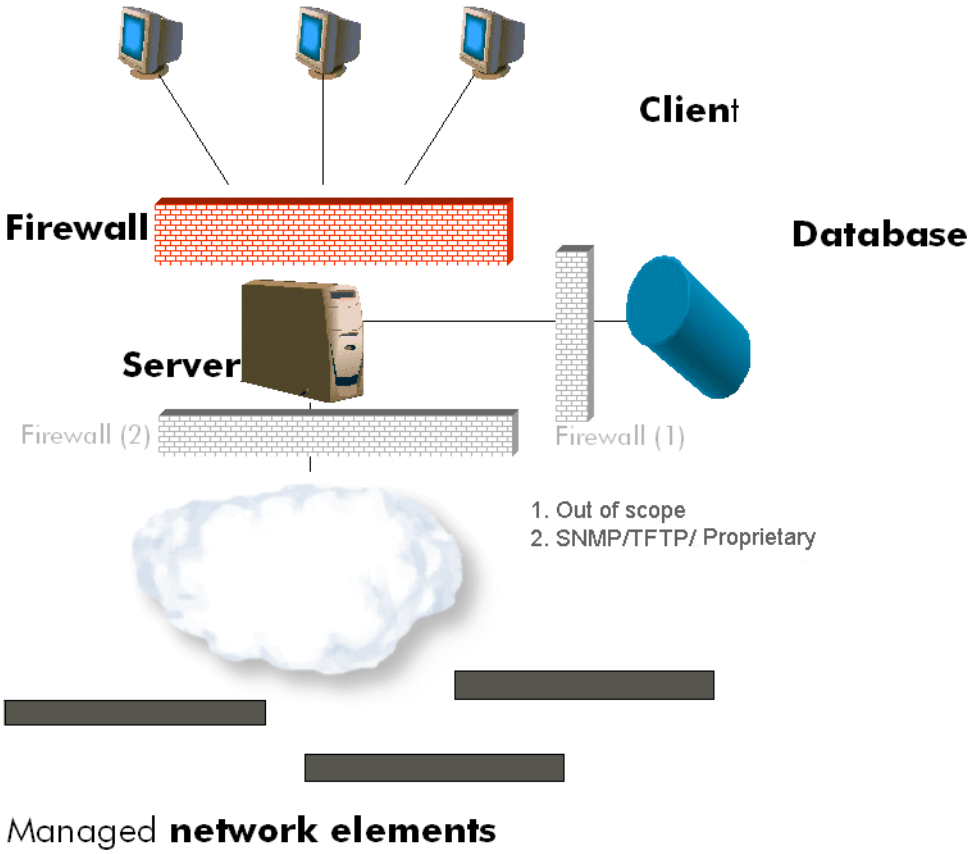


Figure 6 Overview AXX 9840 INSITE Firewall - components

6.1.2 Java RMI and callbacks

Java RMI makes use of communication in a way that natively has little structure; traffic is initiated from both the client and the server, and the connections are primarily non-persistent and short lived. These two properties are normally not compatible with most firewalls. In the AXX 9840 INSITE solution this has been modified to require only one way tcp connection initiations.

In Java RMI, each communicating program is partly a server, partly a client. Therefore, use of the words server and client alone can be confusing. To clarify, we use the phrases RMI server and RMI client when we refer to the java RMI nomenclature. We use the name AXX 9840 INSITE Server and AXX 9840 INSITE Client for the SW provided by Ericsson AXXESSIT.

6.1.3 Technical overview - AXX 9840 INSITE use of Java RMI

The AXX 9840 INSITE client connects to the AXX 9840 INSITE server using 3 tcp ports. All connections are initiated from the client side. The ports are configured at installation by selecting the first port to use. The other ports used are the two following port numbers.

Example: At installation, TCP port 13170, which is also the default, is selected. The next two ports, 13171 and 13172, are also used.

The three ports are: RMI registry port, proprietary service port and the RMI listening port.

Source	Destination	Service	Comments
Clients	Server	RMI Registry (baseport)	Port used for the RMI registry
Clients	Server	Proprietary service (baseport+1)	
Clients	Server	RMI (baseport+2)	Object to Object

Baseport is default 13170 and may be configured during AXX 9840 INSITE installation.

The RMI registry is running as a part of the AXX 9840 INSITE server. The RMI servers residing in the AXX 9840 INSITE server and the AXX 9840 INSITE client will report their existence with their IP address or dns name and listening port to the RMI registry. When a connection to a RMI is required, the RMI registry will be consulted, and a conncton from the RMI client to the RMI server will be established. In the case of an RMI client in the AXX 9840 INSITE client, a tcp connection will be made to the RMI listening port as normal. At some time an RMI client residing in the AXX 9840 INSITE server may want to connect to an RMI server residing in the AXX 9840 INSITE client (a callback). In this case, the call is intercepted by specially crafted socket Factories. A signal is sent through the out-of-band channel to the AXX 9840 INSITE client, which makes a connection to the AXX 9840 INSITE server's RMI listening port , and this connection is presented to the RMI layer.

This chapter focus on the client-server connectivity. Care has been taken during development of the AXX 9840 INSITE client to make sure that there is only need of connections to be established in the

direction from the client to the server. No outbound connections needs to be initiated.

6.2 Configuration

6.2.1 Network scenario

The network scenario that is covered here is:

- An AXX 9840 INSITE manangement server residing in a private network . The managed network elements are on the same network.
- The private network is connected to the Internet using a filtering firewall or a NAT firewall.
- The client is connected either directly to the Internet, or is on a private network behind a filtering router firewall or a NAT firewall.

6.2.2 Overview of the configuration tasks

The two configuration tasks have to be completed:

- A path must be opened through the firewalls for the client to connect to the server via tcp.
- The AXX 9840 INSITE server must be configured with the address where it can be seen from the other party, using the java.rmi.server.hostname property.

Each task will be described in detail in this section.

6.2.3 TMN Server is behind a filtering router firewall

In this case, the server's IP address is a globally routable address. The firewall must be configured to allow incoming calls to the machine running the AXX 9840 INSITE server. That is, all inbound tcp packets to the correct IP and ports must be allowed to pass. In the outbound direction, all tcp packets belonging to established connections must be allowed to pass. A configuration example for netfilter/iptables is provided at the end of this chapter. In this

configuration, no special attention to the java.rmi.server.hostname property is needed.

NOTE! This method is insecure, in that no encryption is provided for the connection. As a minimal security measure, only connections from certain hosts should be allowed.

6.2.4 TMN Server is behind a NAT firewall

In this configuration, the client connects to the firewall IP address. In the firewall, the tcp destination adress is changed to that of the AXX 9840 INSITE server and forwarded to the inside network. In the outbound packets, the source address is changed to that of the firewall. This allows the use of private IP adresses to be used in the inside network. The firewall must be configured to allow incoming connections to three ports, and these connections must be forwarded to the AXX 9840 INSITE server. A configuration example for the firewall is provided at the end of this chapter.

Due to the adress translation, the AXX 9840 INSITE server's **`java.rmi.server.hostname`** property must be set.

To set the property, edit the file **AXX 9840 INSITE_Server.lax file** found in the installation's bin directory, adding a line like this:

```
java.rmi.server.hostname=<ip or dns name>
```

The purpose of this variable is to tell the RMI registry where the RMI server objects are to be found. .

We use a dns-name , this allows to resolve to one address when queried from inside the private network, and to another address when queried from the Internet.

Example:

The AXX 9840 INSITE server is on IP address 10.1.1.10, the firewall's inside interface is 10.1.1.1, the firewall's outside interface is 64.28.25.143.

Add this line to the server's lax file:

```
java.rmi.server.hostname=AXX 9840 INSITE.company.com
```

On the Internet, this name will resolve to 64.28.25.143 in our example, them firewall's outside interface. When this name is queried from the private network, it will resolve to 10.1.1.10, the AXX 9840 INSITE server address.

To accomplish this with DNS (domain name system), use an internal DNS server that takes control over this name, and forwards all other name queries to a regular DNS server. The name may also be added to each workstation's hosts file, pointing directly to the server on internal workstations, and pointing to the NAT firewall's outside interface for workstations on the Internet.

6.2.5 Client is on the internet

When the AXX 9840 INSITE client is running on a workstation directly connected to the internet, no special care has to be taken.

6.2.6 TMN Client is behind a NAT firewall

When the AXX 9840 INSITE client is running on a workstation behind a NAT firewall, no special care has to be taken. Make sure that the dns name of the server resolves to the correct address, which is the firewalls outside interface. An unumeric IP address can be used in the “server” field when connecting , but the server still has to resolve to that IP address.

If a dns-server is not available, a hostfile can be used.

6.2.7 Using SSH tunnels

SSH (Secure Shell) is a protocol for establishing cryptographically strong, secure connections using public key cryptograhy or passwords. The secure channel that is established, can carry the tcp connections necessary for AXX 9840 INSITE client/server communication.

To use SSH, the machine running AXX 9840 INSITE server must also run a SSH server. The AXX 9840 INSITE client workstations must run a SSH client. The firewall must allow incoming calls to the SSH server. In the AXX 9840 INSITE server, the java.rmi.server.hostname must be configured, preferrably using a name in DNS as described in the NAT firewall example above. In the private network, this name mus resolve to the real IP of the server.

This satisfies AXX 9840 INSITE clients running on the inside, not using SSH. On the workstation using SSH, this name must resolve to 127.0.0.1. This is accomplished by inserting the server name in the workstations hosts file, pointing to 127.0.0.1.

Before the AXX 9840 INSITE client is started, the tunnel must be set up. This is an example using the standard ports, assuming "myuser" is a valid username for establishing the ssh connection:

```
ssh -L 13170:127.0.0.1:13170 -L 13171:127.0.0.1:13171 -L
13172:127.0.0.1:13172 myuser@AXX 9840 INSITE.company.net
```

This command will establish a SSH connection to the host running AXX 9840 INSITE server, and forward three local ports to the corresponding ports on the server host. After this, AXX 9840 INSITE can be started, specifying localhost for the server.

NOTE! This method of communication is secure, provided that non guessable passwords are used, or private/public key pairs are used. See SSH documentation for details.

6.2.8 Reference implementation for iptables

The follow lines should be sufficient for a very basic connectivity using iptables on supported platforms:

6.2.8.1 Assign base port for AXX 9840 INSITE

```
RMIREGISTRY=13170
REMOTE=`expr $BASE +1`
CALLBACK=`expr $BASE + 2`
```

6.2.8.2 Set Policies, by default allow nothing.

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

6.2.8.3 Create new inbound TCP-chain

```
iptables -N tcp_inbound
iptables -A INPUT -p ALL -m state --state ESTABLISHED,RELATED -j
ACCEPT
```

6.2.8.4 Add to chain, access for AXX 9840 INSITE-connectivity

```
iptables -A tcp_inbound -p TCP -s 0/0 --source-port $BASE -j ACCEPT
iptables -A tcp_inbound -p TCP -s 0/0 --source-port $REMOTE -j
ACCEPT
iptables -A tcp_inbound -p TCP -s 0/0 --source-port $CALLBACK -j
ACCEPT
```

6.2.8.5 Create outbound connections for manage networks

```
iptables -N tcp_outbound
iptables -A tcp_outbound -p TCP -d 0/0 --destination-port 161 -j
ACCEPT
```

6.2.8.6 Connect using SSH and remote admin of server

```
iptables -A tcp_inbound -p TCP -s 0/0 --destination-port 22 -j
ACCEPT
```

6.2.8.7 Bind the tcp_inbound chain to the INPUT rules

```
iptables -A INPUT -p TCP -j tcp_inbound
iptables -A INPUT -p ALL -d 255.255.255.255 -j DROP
```

6.2.8.8 Bind the tcp_outbound chain to the OUTPUT rules

```
iptables -A OUTPUT -p tcp -j tcp_outbound
iptables -A OUTPUT -p ALL -d 255.255.255.255 -j DROP
```

This configuration can be made far more extensive to support state-inspection, etc. Latitude also need to be given to the servers other needs, such as access to DNS, Mail, inbound SNMP traps, etc.

6.2.9 Verify the configuration

To verify the configuration, output from the command "**`netstat -na`**" should show similar results to this after starting the services:

```
tesstuser@server(/home/testuser)$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address         State
tcp        0      0 0.0.0.0:13170 0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:13171 0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:13172 0.0.0.0:*               LISTEN
```

The lines above represent the various interfaces to which the services are bound. The special IP address 0.0.0.0 indicates the service is listening to all available interfaces that is configured with IP addresses.

6.2.10 Firewall considerations for AXX 9840 INSITE server - NEs

6.2.10.1 Configuration management

- SNMP SET/GET - UDP 161
- Telnet (AXXCLI) - TCP 23
- Ericsson AXXESSIT proprietary protocol Bulk Transfer - default TCP range 4500 - 4510 (editable range and configurable to either "true" or "false". If "false" normal SNMP GET operations will be used.)

File : res/config/**Bulktransfer.env.xml**.

```
<bulktransfer>
<enabled  value="true"/>
  <compress value="true"/>
  <lowerlimit value="4500"/>
  <upperlimit value="4510"/>
</bulktransfer>
```

6.2.10.2 Fault management

- SNMP traps - UDP 162

6.2.10.3 Maintenance

- TFTP (software download and config upload)- UDP 69
- Time protocol (RFC868) - UDP 37

6.2.10.4 Port summary

Open ports for traffic towards the NE:
UDP 161

TCP 23

Open ports for traffic towards the AXX 9840 INSITE Server from the NE:
UDP 162

UDP 69

TCP 4500 - 4510 (can be edited)

AXX 9300 METRO - SSH server

SSH server standard default TCP port 22.

AXX 9840 INSITE SSH server default TCP port 13179 (configureable)

7 AXX 9880 MEDIATOR Notification GW

7.1 AXX 9880 MEDIATOR Notification Gateway

7.1.1 The generic alarm and event Notification Gateway

This interface implements user configurable real-time forwarding of notification events (alarm or event notifications) to external

applications. It provides a standardized data interface and a customizable gateway protocol, as shown in Figure 7

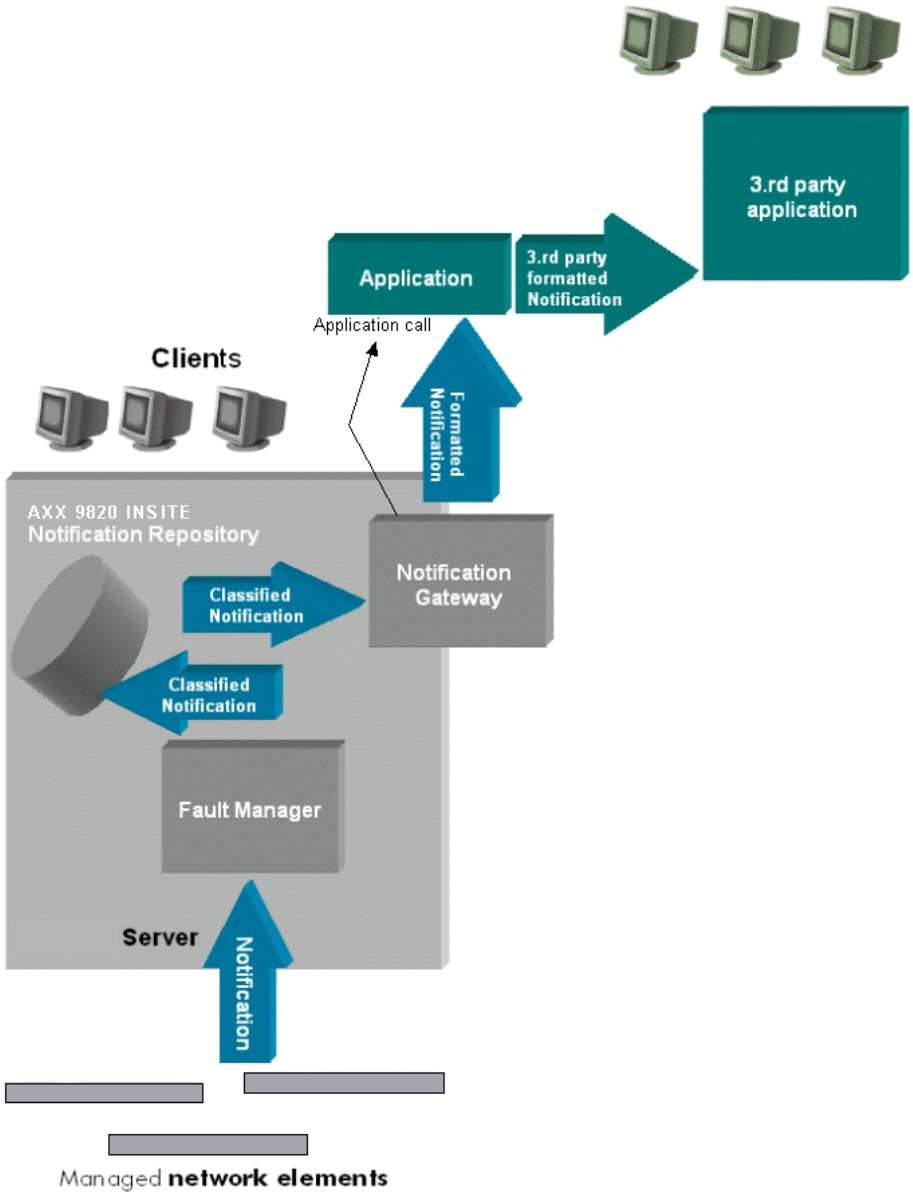


Figure 7 Notifaction Gateway - Overview

7.2 Output formats

The Notification Gateway supports two output formats:

- Comma Separated Value (CSV)
- Extensible Markup Language (XML)

These formats are described in more details in the following sections. Values are formatted according to Notification List if nothing is mentioned explicitly. Table 3 - Table 4 list all fields in correct order for alarms and events.

7.2.1 Alarm notification fields

Field	Description
NotificationType	[alarm]. Fixed constant.
AddTxt	See Notification List.
AlarmType	See Notification List.
AlarmId	See Notification List.
Clearable	[yes|no]See Notification List.
EmsTime	See Notification List.
LayerRate	See Notification List.
Location	See Notification List.
Source	See Notification List.
NeTime	See Notification List.
NE	See Notification List.
PrvId	Previous Notification Identifier. Identifier on the notification that updated current Notification. See Notification List.
ProbCause	See Notification List.
ProbCauseQ	See Notification List.
SCM	State Change Mode. 0 = new, 1=updated, 2= Return To Normal. An alarm point is identified by NE, Source and AlarmId.
Id	Unique Identifier for a Notification.
Severity	See Notification List.
Trend	See Notification List.
AckSign	See Notification List.
AckTime	See Notification List.
AckSign	See Notification List.
Acked	[yes|no]. See Notification List.
Comment	See Notification List.
Duration	See Notification List.
Link	Affected link. See Notification List.

Table 3 Alarm notification fields

7.2.2 Event notification fields

Field	Description
NotificationType	[event]. Fixed constant.
AddTxt	See Notification List.
EventType	See Notification List.
EventId	See Notification List.
Source	See Notification List.
NeTime	See Notification List.
NE	See Notification List.
SCM	State Change Mode. 0 = new, 1=updated, 2= Return To Normal. An alarm point is identified by NE, Source and AlarmId.
Id	Unique Identifier for a Notification.

Table 4 Event notification fields

7.2.3 Comma Separated Value (CSV) format

The following conventions are used:

- Each record is one line
- Fields are separated with commas
- Fields with embedded commas is delimited with double-quote characters

7.2.4 Extensible Markup Language (XML) format

The following XML structures are used, shown by examples:

```
<?xml version="1.0" encoding="UTF-8"?>
<alarm>
<AddTxt val=""/>
<AlarmType val="equip"/>
<AlarmId val="1.2.4"/>
<Clearable val="no"/>
```

```
<EmsTime val="2004/11/10 15:39:29"/>
<LayerRate val="Not_Applicable"/>
<Location val="--"/>
<Source val="AXX 9200 EDGE-Basalt"/>
<NeTime val="2004/11/10 15:39:29"/>
<NE val="AXX 9200 EDGE-Basalt"/>
<PrvId val="1100092422372"/>
<ProbCause val="abortTftp"/>
<ProbCauseQ val="TFTP session aborted -"/>
<SCM val="1"/><Id val="1100092422384"/>
<Severity val="Warning"/>
<Trend val="No Change"/>
<AckSign val=""/>
<AckTime val="0"/>
<Acked val="no"/>
<Comments val=""/>
<Duration val="0h 00m 00s"/>
<Link val=""/>
</alarm>

<?xml version="1.0" encoding="UTF-8"?>
<event>
<AddTxt val=""/>
<EventType val="equip"/>
<EventId val="2.2.2"/>
<EmsTime val="2004/11/10 15:39:24"/>
<Source val="AXX 9200 EDGE-Basalt"/>
<NeTime val="2004/11/10 15:39:24"/>
<NE val="AXX 9200 EDGE-Basalt"/>
<SCM val="0"/>
<Id val="1100092422382"/>
<Description/>
</event>
```

7.3 Configuration

Configuration of the Notification Gateway is located under the **AXX 9880 MEDIATOR> Notification Gateway** object in Management Tree.

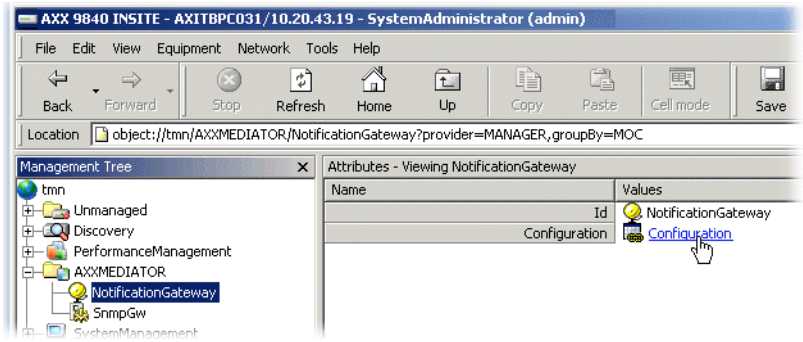


Figure 8 Loaction of Notification Gateway in the Management Tree

The configuration is a table where each row represents an executable command that may be executed when an alarm or event is received based on filter criteria. The command may launch a shell script, executable etc., with the notification as parameter or the notification is indirectly passed to the application through a file.

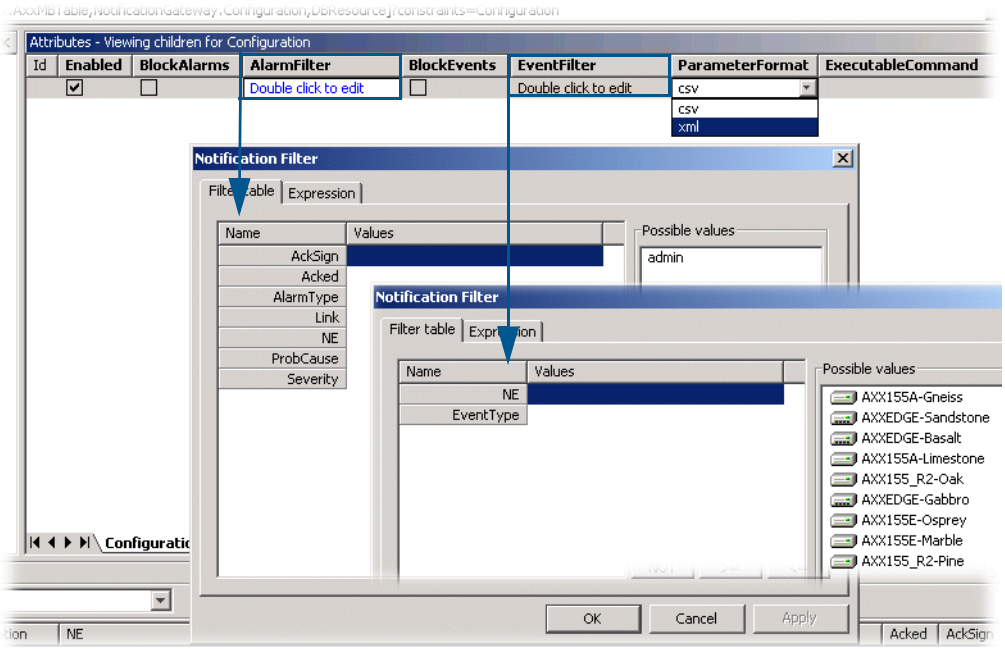


Figure 9 Configuration of Notification Gateway

A description of the attributes if found in Table 5

TIP! Please see the User Guide for Filter set-up.

Attribute	Description
Enabled	Enables or disables the execution of the command.
BlockAlarms	Will block all alarms independently of the alarm filer.
EventFilter	A fine-grained event filter.
AlarmFilter	A fine-grained alarm filter.
BlockEvents	Will block all events independently of the event filer.
Parameterformat	[csv|xml]
ExecutableCommand	The command(s) that is executed when a notification is accepted. There are defined three parameter constants: %VALUE%, %FILE%, %FILE:<path>%. **%VALUE%**, a notification is passed to an application as a parameter (xml or csv format). Note: This may cause interpretation error in some shells due to escape characters. **%FILE%**, a temporary file is created and the notification is written to this file. The filename is passed to the application as a parameter. NOTE: It is the responsibility of the target application to delete the temporary file. However, the file is deleted on a successfully server shutdown. The temporary file follows name pattern: ngw<unique generated pattern>.tmp **%FILE:<file path>%**, notifications are appended to a given file. The file path is passed to the application as a parameter. NOTE: It is the responsibility of the target application to manage the file.It is possible to use the constants several times in a command, e.g. "./sendmail %FILE:./file1.txt% %FILE:./file2.txt%" will write every notification to file1.txt and file2.txt.
WorkingDirectory	The working directory of the new sub-process. If empty, the sub-process inherits the current working directory of the current process, i.e. <installation>/bin. The temporary file is written to working directory.

Table 5 Notification Gateway - Attributes

7.3.0.1 Ex. 1 (Solaris, FILE-parameter)

1. Add a line in NotificationGateway parameters with Executable command:

```
/usr/local/bin/notificationgw2 %FILE%
```

Specify a working directory, f.ex /tmp

2. Create an executable file /usr/local/bin/notificationgw2 with content:

```
#!/usr/bin/bash
cat >> /tmp/notificationgw2.out <$1
rm $1
```

This will append the notification temporary file to/**tmp/notificationgw2.out** and delete the tempfile.

7.3.0.2 Ex. 2 (Win VALUE-parameter, cygwin is installed)

1. Add a line in NotificationGateway parameters with Executable command:

```
d:\cygwin\bin\bash.exe d:\work\notificationgw4.sh %VALUE%
```

Script file d:\work\notificationgw4.sh contains:

```
echo "$@" >> d:\\work\\notificationgw4bash.out
```

This will append all parameters (constituting the notification) to the file d:\\work**notificationgw4bash.out**

8 AXX 9880 MEDIATOR SNMP Gateway

8.1 Introduction

8.1.1 Purpose

Ericsson AXXESSIT is pleased to present a solution that addresses the requirements for integration with overall SNMP-based fault management systems, typically industry standard management systems such as HP OpenView, Micromuse Netcool or Castle Rock Computing SNMPc.

This chapter presents AXX 9880 MEDIATOR SNMP Gateway, the SNMP based alarm-forwarding solution for the AXX 9840 INSITE network manager.

8.1.2 Definitions, Acronyms and Abbreviations

O-FM
Overall Fault Management System

SNMP GW
AXX 9880 MEDIATOR SNMP Gateway

SNMP
Simple Network Management Protocol (SNMP) is the protocol governing network management and the monitoring of network devices and their functions. It is not necessarily limited to TCP/IP networks. SNMP is described formally in the Internet Engineering Task Force (IETF) Request for Comment (RFC) 1157 and in a number of other related RFCs

MIB
A management information base (MIB) is a formal description of a set of network objects that can be managed using the Simple Network Management Protocol (SNMP). The format of the MIB is defined as part of the SNMP.

8.2 Overview

Ericsson AXXESSIT offers several alternative solutions for integration between the Ericsson AXXESSIT network products and management systems from other vendors. These solutions are utilizing the open interfaces of the Ericsson AXXESSIT NEs and management software to provide the desired level of integration and functionality. They also allow the operator to choose between integration products provided by Ericsson AXXESSIT or by third party vendors.

AXX 9880 MEDIATOR is the plug-in framework for Element and Network Management-level management integration. Based on the AXX 9800 TMN framework, the AXX 9880 MEDIATOR is a multi-technology and multi-vendor integration platform for the AXX 9840 INSITE Network Management system. AXX 9880 MEDIATOR can be delivered with standard or proprietary interfaces. Proprietary interfaces can be developed as part of a system integration project.

The AXX 9880 MEDIATOR interfaces are delivered as plug-in gateways. The SNMP GW is one of the AXX 9880 MEDIATOR integration interfaces.

The SNMP GW plugs into AXX 9840 INSITE. The SNMP GW forwards alarms (traps) collected from the managed network to any SNMP-based fault management system.

8.2.1 Customer Benefits

Many customers installing Ericsson AXXESSIT network equipment and management products already have installed some overall fault management system, covering multiple vendors. Very often, this fault management system is SNMP based. When the requirement is only to collect alarms, it is often not feasible to do a direct integration with the network elements, but rather collect the alarm information from each vendor’s element manager (or even network manager). This has a number of benefits:

- Reduced number of connections and traffic in the management network. Overall fault management system does not connect to each network element in parallel with element- and network management system.
- Easier integration due to a significantly smaller MIB for the SNMP GW than the network element MIBs.

- Lower maintenance cost of integration (“integration tax”) as the integration is not directly exposed to upgrades in the network elements.
- Alarm flooding of overall fault management system is avoided through filtering mechanisms in the SNMP GW.
- Alarms generated by the element- or network management system (e.g. internal alarms) are made available to the overall fault manager.
- Avoid the scalability problems of having each O-FM user registered in every NE’s community table
- Support for alarm synchronisation.

To accommodate integration with the most common SNMP based fault management systems such as HP Open View (Network Node Manager) and Micromuse Netcool the trap formats are configurable.

8.2.2 Features

The SNMP gateway has the following overall functionality:

- Maps AlarmNotifications and EventNotifications to SNMP traps and send them to a specified set of receivers.
- Filter alarms and events based on attributes of the alarms/events.
- Stores all alarms in a log that is accessible to the higher level management system as an ordinary SnmpTable using SNMP calls to the gateway.
- Resending of alarms and resynchronisation on restart.
- Cold restart. • Sends the traps to the receivers using either of three trap data formats.

8.3 Configuration

This chapter provides necessary procedures needed for configuration and operation of the AXX 9880 MEDIATOR SNMP Gateway.

8.3.1 Starting the SNMP GW

Administrating the SNMP GW is handled only by the system administrator role. When the system administrator launches AXX 9840 INSITE the SNMP GW object is presented in the Management Tree.

Selecting this object, the attributes for filtering and adding receivers (O-FMs) are displayed.

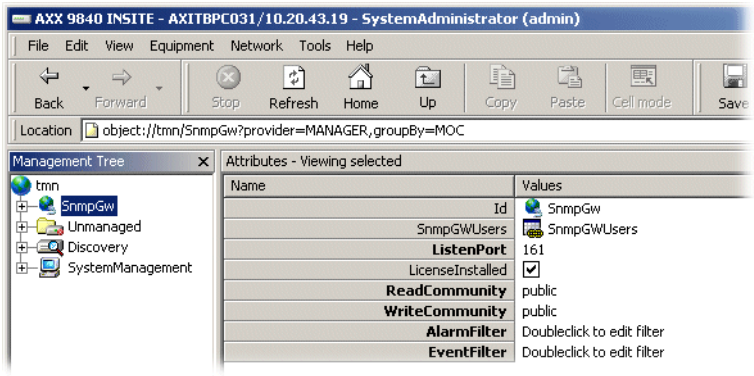


Figure 10 SnmpGW in Management Tree

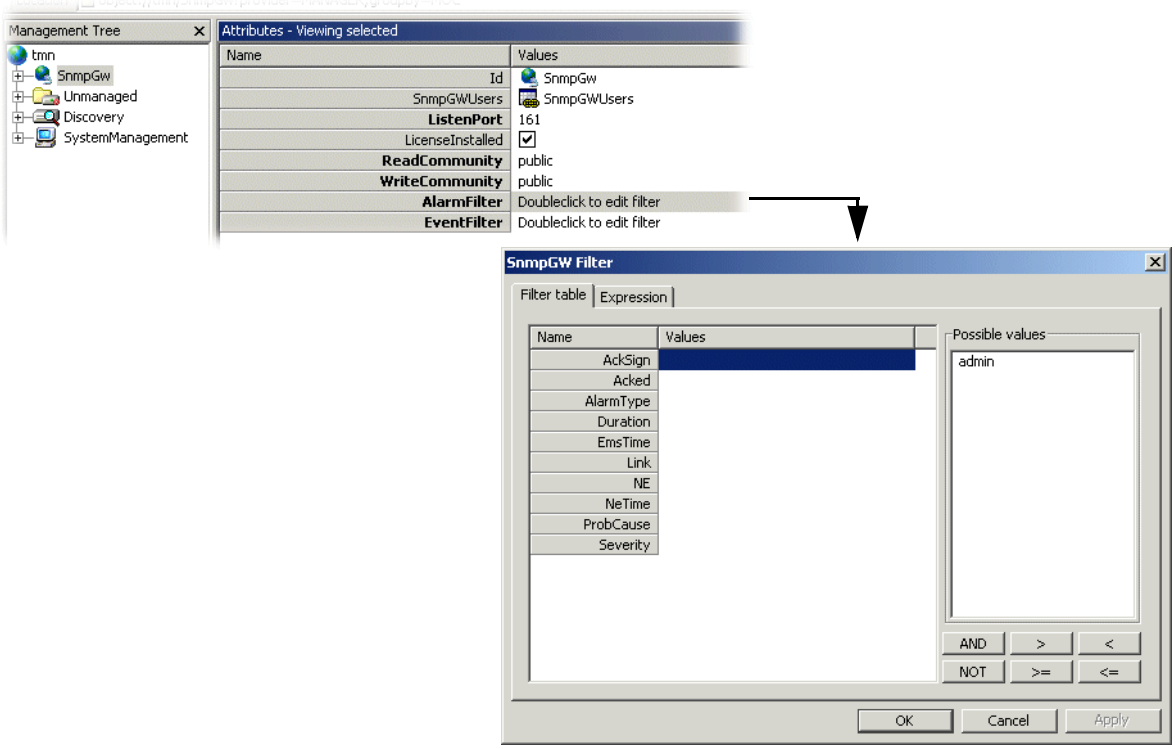
8.3.2 SNMP GW object attributes

The SNMP GW object contains several attributes:

Procedure: Specify AlarmFilter and EventFilter

The AlarmFilter and EventFilter are dialogs where the user may specify the alarm and event filters. Each filter is valid for all receivers. Individual filters for each receiver is not allowed in this version.

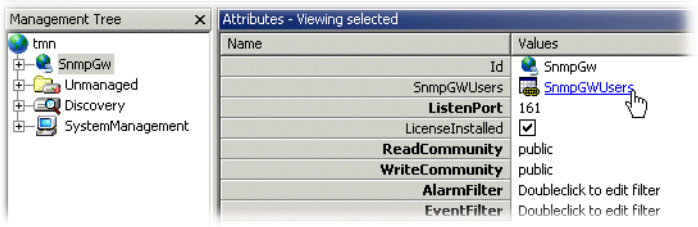
1. Doubleclick to edit filters.



Procedure: Add SnmpGWUsers

There is a table used to specify the receivers of the SNMP traps.

1. Click **SnmpGWUsers**.

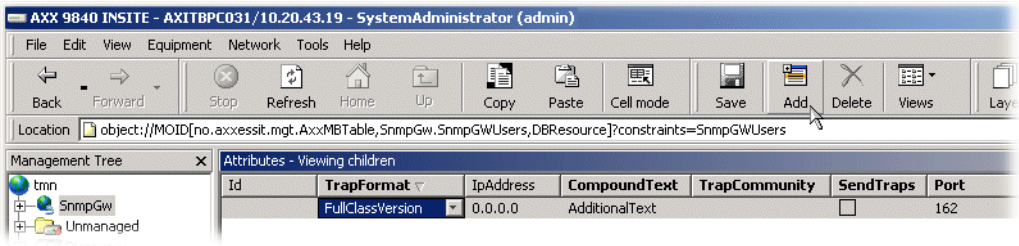


2. Click **Add** in the toolbar.

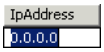
Each receiver represents one northbound manager and is defined by the following attributes:

- Ip Address
- Compound Text

- Trap Community
- Send Traps
- Port



3. **Enter** the IP address used for the manager.

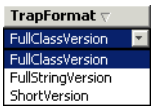


Compound Text; an optional user defined string composed of text and any combination of the fields shown in “Alarm formats (VarBinds)” on page 151.

4. **Select** desired combination from pulldown menu.

5. Enter **Community string** sent with the trap.

6. Select the **format of the trap** the O-FM expects.



7. Set the **port** the O-FM uses to receive traps from the SNMP GW. (and possibly other sources.)

8. Click checkbox to indicates that a user is going to do get and set operations, and to receive traps as well.

ReadCommunity

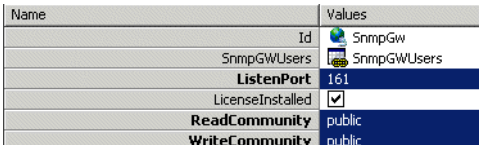
Common for all users.

WriteCommunity

Common for all users.

ListenPort

The SNMP port that is used for get and set operations.



8.3.3 Network Element - Management IP address

1. In AXX 9840 INSITE, select the NE managed object **to view current** Management IP address.

Default management IP address is the network element IP address.

2. **If this differ** from the address expected by the SNMP Manager , select the ManagementIP managed object and to **edit** the IP address to desired value.

3. Press **Save** in the Toolbar.

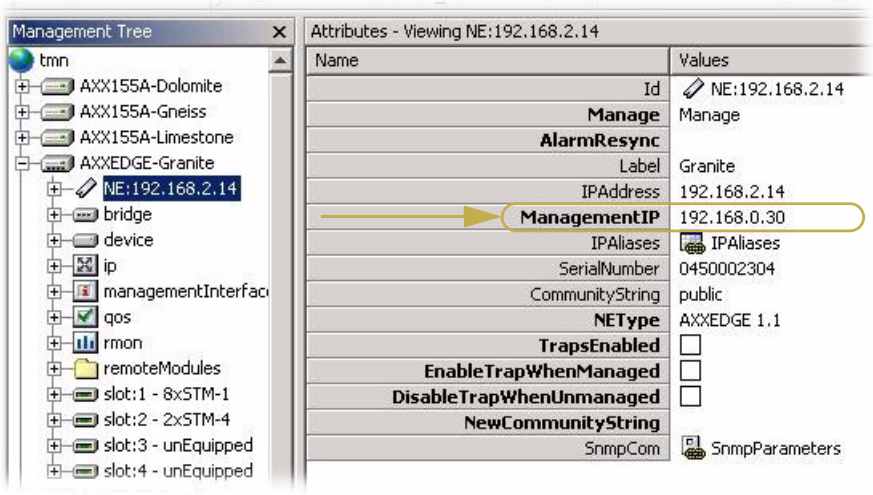


Figure 8-1. Management Tree - NE Management IP address

8.4 Scenarios

This sections presents four possible scenarios for the AXX 9880 MEDIATOR SNMP Gateway. Two scenarios focus on the gateway and the overall fault management system (O-FM) installed on

different server. The two others, shows the gateway and the O-FM installed on same server.

NOTE! For details on configuration needed on the AXX 9880 MEDIATOR SNMP Gateway in the different scenarios, please see “Add SnmpGWUsers” on page 141 (“Port”) and “ListenPort” on page 142.

8.4.1 Scenario 1

In this scenario, the AXX 9880 MEDIATOR SNMP Gateway and the O-FM are running on different servers. The port the O-FM uses to receive traps from the SNMP GW is set to # 162 and the Listen port is set (default) to # 161.

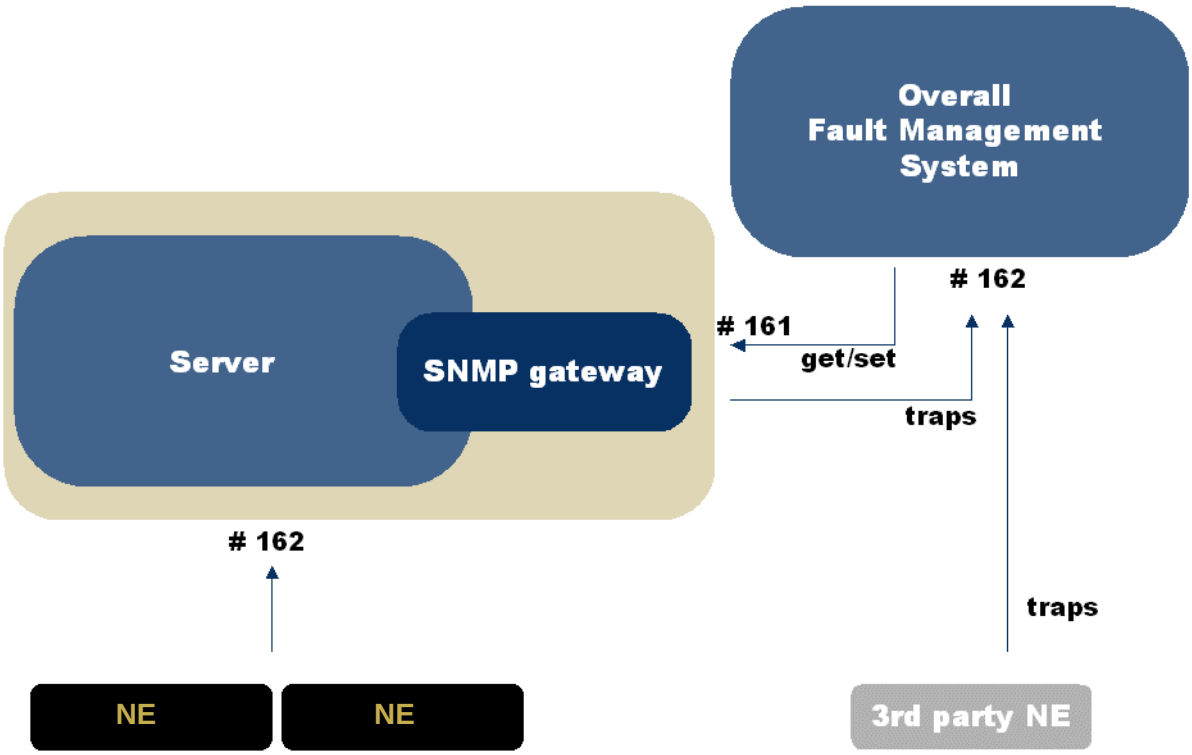


Figure 9 SNMP gateway and O-FM - running on different servers

8.4.2 Scenario 2

The second secenario introduces an 3rd party SNMP agent. AXX 9880 MEDIATOR SNMP Gateway and the O-FM are running on different servers. The port the O-FM uses to receive traps from the SNMP GW is set to # 162 The Listen port for SNMP Gateway (default # 161) must be configured and set to a port differnt from the Listen port chosen for the 3rd party SNMP agent .

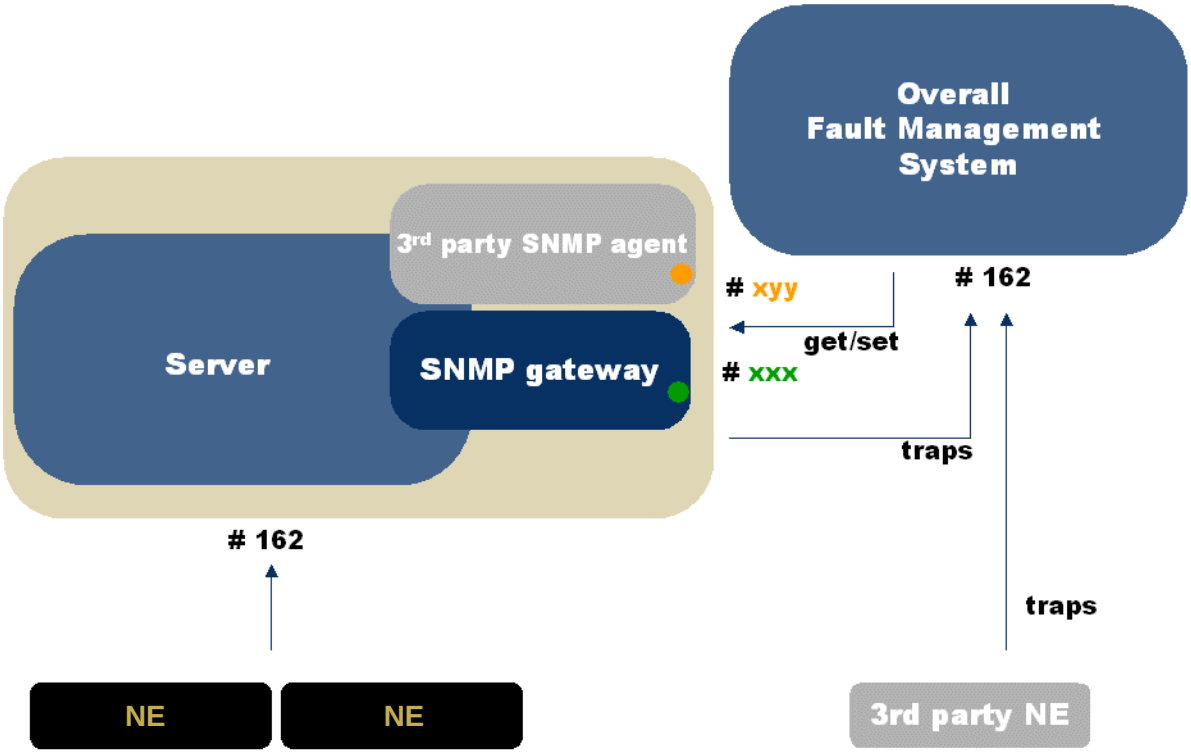


Figure 10 GW and O-FM on different servers - 3rd party SNMP agent

8.4.3 Scenario 3

In scenario 3, the AXX 9880 MEDIATOR SNMP Gateway and the O-FM are running on same server. The Trap port on the O-FM must be configured and set to a port different than # 162. The SNMP Gateway must be configured to send traps to the chosen O-FM Trap port.

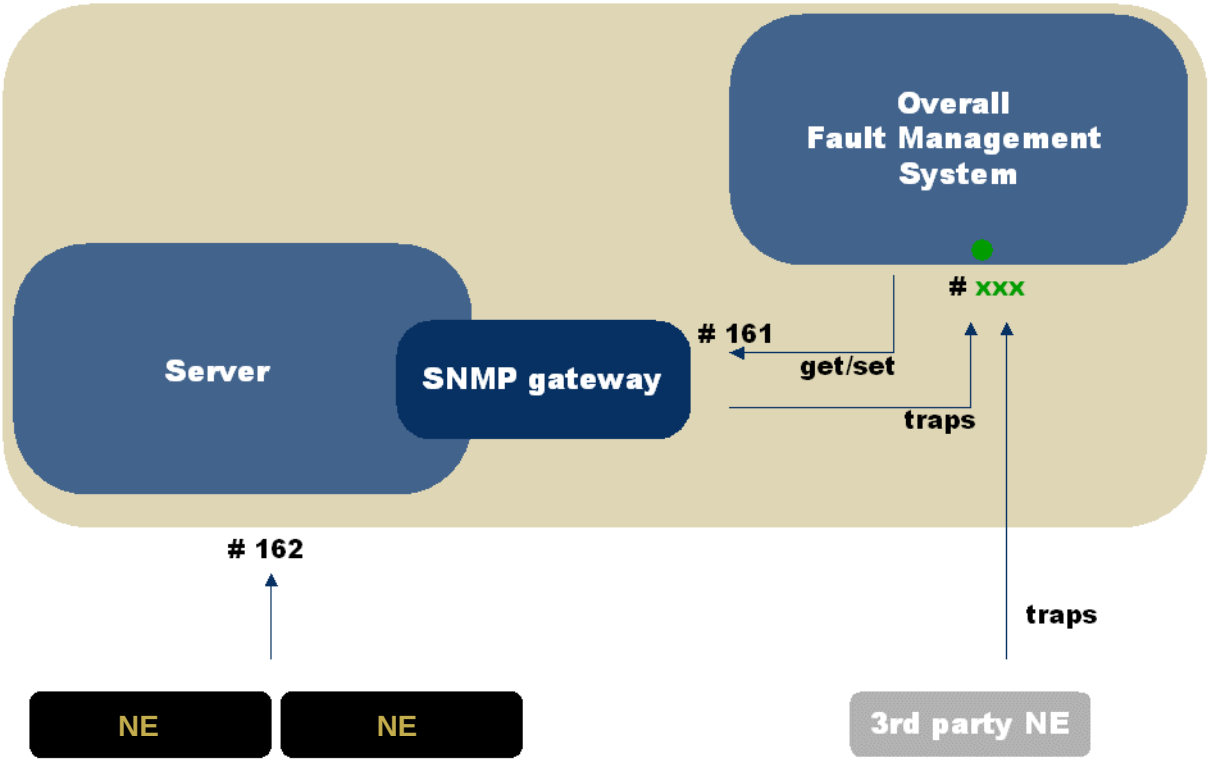


Figure 11 SNMP Gateway and O-FM running on same server

8.4.4 Scenario 4

If the O-FM must receive traps from other NEs from on port # 162 and the AXX 9880 MEDIATOR SNMP Gateway and the O-FM are running on the same server, a trap distributor is needed. The trap distributor listen on port # 162 and distributes trap to the Trap ports of O-FM and the SNMP Gateway. This requires that the SNMP GW Trap port is configured and set to a port different from the port chosen for the O-FM Trap port. Please contact your Ericsson AXXESSIT contact for further details.

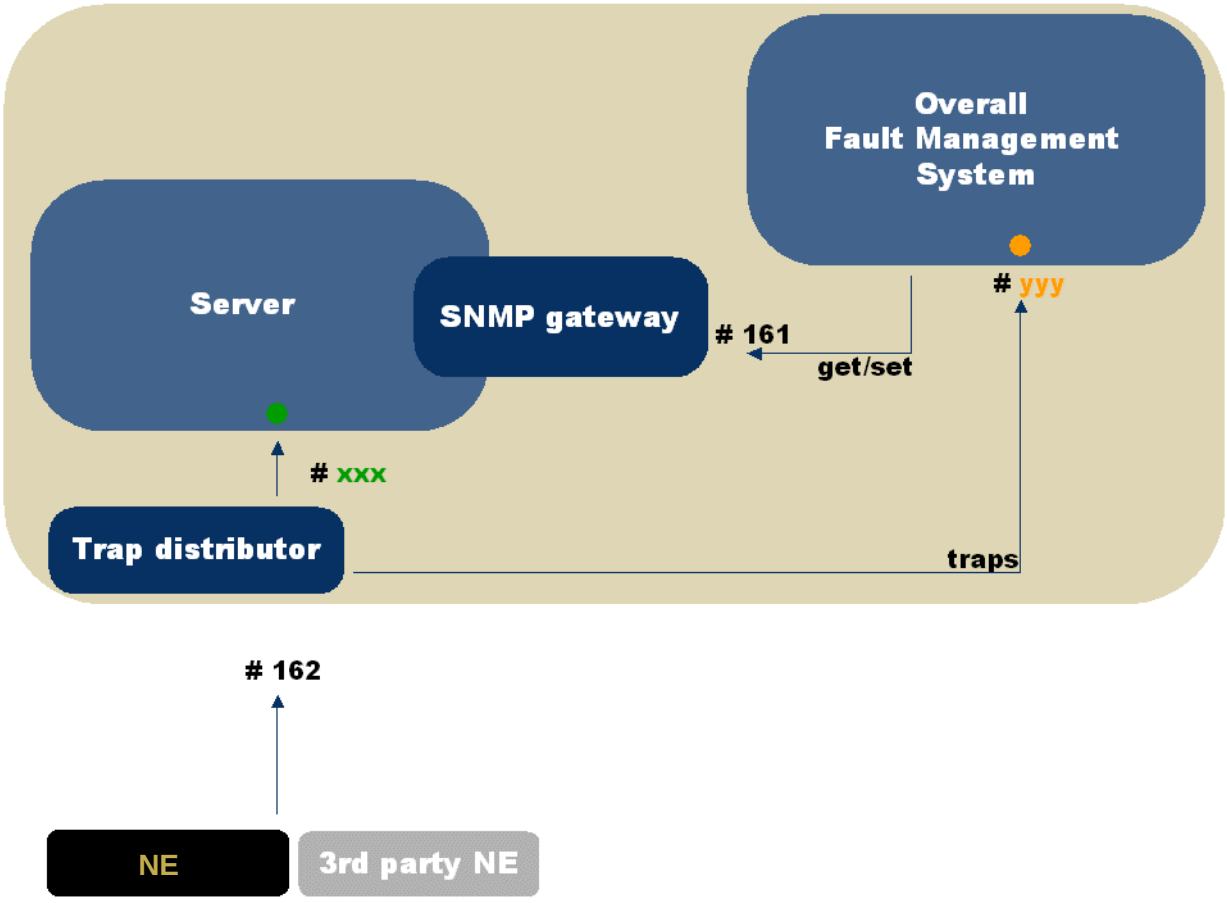


Figure 12 Trap distributor - SNMP Gateway/O-FM

8.5 SNMP Agent

8.5.1 Protocol

The agent supports SNMPv1. 4.2

8.5.2 Multiple SNMP Agents

The SNMP GW agent use of UDP ports is user configurable, in order to enable the SNMP GW to co-exist with the server operating system SNMP Agent.

8.5.3 Management Information (MIB)

The SNMP GW MIB supports AXX 9840 INSITE alarm information in terms of:

- Traps, i.e. spontaneous reporting of alarms and events.
- Active Alarms Table, i.e. all active alarms registered in the AXX 9840 INSITE database.
- Additional parameters.

8.5.4 SNMP trap formats

The SNMP TRAP sent from the SNMP GW contains the following fields:

Field	Value
Enterprise	1.3.6.1.4.1.7546.10
AgentAddress	The agent-address from the PDU. This means the address of the AXX 9840 INSITE Server
SourceAddress	The address of the NE that generated the trap
GenericTrapType	COLDSTART(0)
SpecificTrapType	ALARMRAISED(1) ALARMCLEARED(2) ALARMEVENT(3)
ErrorStatus	NOERROR(0)
ErrorIndex	0
PduType	V1TRAP(-92)
TimeStamp	SysUpTime
VarBinds	This is dependent upon the format of the trap. The format again depends on what the northbound manager expects to receive. Three different formats are specified: Format**#1** is a typical application-readable format with strings and enums. Format**#2** is a typical “conversion to human reading” format only containing strings Format**#3** is similar to Format#2 but the string to be presented in the O-FM is pre-formatted by the AXX 9840 INSITE server.

Table 6 SNMP trap structure

8.5.5 VarBinds

Field	Description	Format#1	Format#2	Format#3	Applicable to
Id	Unique identifier of the alarm	long	long	long	Alarm + Event
Type	Alarm or Event	Boolean	String	N/A	Alarm + Event
Acked	True if acknowledged	Boolean	String	N/A	Alarm
Ack signature	The users signature	String	String	N/A	Alarm
Act time	The acknowledgement time stamp	Date	String	N/A	Alarm
Severity		Enum	String	Enum	Alarm
NE IP address	Source IP address	IP address	String	IP address	Alarm + Event
NE	Name of the NE	String	String	N/A	Alarm + Event
Source	The managed object	String	String	N/A	Alarm + Event
NeTime	The network element timestamp	Date	String	N/A	Alarm + Event
EmsTime	The management system timestamp	Date	String	N/A	Alarm + Event
ProbCause	Probable cause	Enum	String	N/A	Alarm
ProbCause Q	Probable cause qualifier	String	String	N/A	Alarm
Clearable	Indicate that an raised alarm may be followed by a cleared alarm	Boolean	String	N/A	Alarm
Alarm/EventType	The alarm/event type	Enum	String	N/A	Alarm + Event
Description	The description of the event	String	String	N/A	Alarm + Event
Location	Slot/Port/cbklm value	String	String	N/A	Alarm
Comments	A user comment	String	String	N/A	Alarm
Alarm/Event Id	Identification of the alarm. (same as probable cause)	String	String	N/A	Alarm + Event
Trend	more/less severe	Enum	String	N/A	Alarm

Field	Description	Format#1	Format#2	Format#3	Applicable to
Link	The identification of the affected link	String	String	N/A	Alarm
Layer rate		Enum	String	N/A	Alarm
AddTxt	Additional Text	String	String	N/A	Alarm + Event
Compound Text	An optional user defined string composed of text and any combination of the above fields.	N/A	N/A	String	Alarm + Event
Max Alarm Severity	Telling the highest severity level currently active for the network element subject to the trap	Enum	String	Enum	Alarm

Table 7 Alarm formats (VarBinds)

8.6 Management information base - MIB

8.6.1 Introduction

The following MIBs are delivered with the AXX 9880 MEDIATOR SNMP Gateway and must be compiled in the same order as listed below:

- SNMPv2-MIB
- If-MIB
- AXXESSIT-ROOT-MIB
- GFMI-MIB

8.6.2 AXXESSIT ROOT MIB

```
AXXESSIT-ROOT-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        MODULE-IDENTITY,
        enterprises         FROM SNMPv2-SMI;

    axxessIt OBJECT IDENTIFIER ::= { enterprises 7546 }
```

```
    axxRootMIB MODULE-IDENTITY
        LAST-UPDATED "0110051521Z"
        ORGANIZATION "AxxessIT"
       CONTACT-INFO
            "Arvid Sallaup
             Postal: AXXESSIT ASA,
                     P.O.Box 6120, Postterminalen
                     N-5892 Bergen
                     Norway
             Tel:    +
             E-Mail: arvid.sallaup@axxessit.no"
DESCRIPTION
            "This is the root MIB of AxxessIT"
        REVISION      "0009291638Z"
        DESCRIPTION
            "Version 0.1:   The initial version of this MIB module
                            with fake enterprise OID for AxxessIT."
        REVISION      "0011091528Z"
        DESCRIPTION
            "Version 0.2:   axx155 => axx to make AXX-MIB generic to all
                            AXX devices."
        REVISION      "0011291254Z"
        DESCRIPTION
            "Version 0.3:   Updated with real Axxess IT enterprise OID."
        REVISION      "0110051521Z"
        DESCRIPTION
            "Version 0.4:   Added specific OID for AXX155 release 2,
                            AxxLink and AXX 9200 EDGE."
        ::= { axxessIt 1 }

    -- AxessIT specific OIDs.
    axx      OBJECT IDENTIFIER ::= { axxRootMIB 1 }
    axx155   OBJECT IDENTIFIER ::= { axxRootMIB 2 }
    axxLink  OBJECT IDENTIFIER ::= { axxRootMIB 3 }
    AXX 9200 EDGE  OBJECT IDENTIFIER ::= { axxRootMIB 4 }
AXX 9800 TMN   OBJECT IDENTIFIER ::= { axxRootMIB 30 }

END
```

8.6.3 GFMI MIB

```
GFMI-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        MODULE-IDENTITY,
        OBJECT-TYPE,
        Unsigned32,
        IpAddress
          FROM SNMPv2-SMI

        DisplayString,
        DateAndTime        FROM SNMPv2-TC

        TRAP-TYPE          FROM RFC-1215
        AXX 9800 TMN             FROM AXXESSIT-ROOT-MIB;
```

```
--------------------------------------------------------------------------------
---
--  Module identity part.
--------------------------------------------------------------------------------
---
    gfmiMIB MODULE-IDENTITY
        LAST-UPDATED "0502181130Z"
        ORGANIZATION "AXXESSIT"
        CONTACT-INFO
            "Arvid Sallaup
             Postal: AXXESSIT ASA,
                     P.O.Box 6120, Postterminalen
                     N-5892 Bergen
                     Norway
             Tel:    +47 977 63 516
             E-Mail: arvid.sallaup@axxessit.no"
        DESCRIPTION
            "The MIB module defines the interface to the AXX 9800 TMN
             Generic Fault Manager Interface (GFMI)."
        REVISION      "0402061500Z"
        DESCRIPTION
            "Version 0.1:   The initial version of this MIB module."
REVISION      "0502181130Z"
        DESCRIPTION
           "Version 0.2: Added gfmiMaxAlarmSeverity gfmiMaxAlarmSeverityStr."

        ::= { AXX 9800 TMN 1 }

--------------------------------------------------------------------------------
---
--  GFMI MIB main groups.
--------------------------------------------------------------------------------
---
    gfmi        OBJECT IDENTIFIER ::= { gfmiMIB 1 }
    gfmiObject  OBJECT IDENTIFIER ::= { gfmi 1 }

--------------------------------------------------------------------------------
---
--  Local data types (in alphabetical order).
--------------------------------------------------------------------------------
---
    Acked            ::= INTEGER
    {
        acked    (1),
        notAcked (2)
    }

    AckedStr         ::= DisplayString (SIZE (9))

    AckSignature     ::= DisplayString (SIZE (0..150))

    AckTimeStr       ::= DisplayString (SIZE (19))

    AddTxt           ::= DisplayString (SIZE (0..250))

    AlarmId          ::= DisplayString (SIZE (0..10))

    AlarmType        ::= INTEGER
    {
      communications   (1),
```

```
      qualityOfService (2),
      processingError  (3),
      equipment        (4),
      environmental    (5)
   }

    AlarmTypeStr     ::= DisplayString (SIZE (0..255))

    Clearable        ::= INTEGER
    {
        clearable (1),
        notClearable (2)
    }

    --  Clearable/Not Clearable
    ClearableStr     ::= DisplayString (SIZE (9..13))

    Comments         ::= DisplayString (SIZE (0..150))

    CompoundTxt      ::= DisplayString (SIZE (0..2048))

    Description      ::= DisplayString (SIZE (0..250))

    -- dd.mm.yyyy hh.mm.ss
    EmsTimeStr       ::= DisplayString (SIZE (19))

    EventId          ::= DisplayString (SIZE (0..10))

    EventType ::= AlarmType

    EventTypeStr     ::= DisplayString (SIZE (0..255))

    LayerRate        ::= INTEGER
    {
       lineRateNotApplicable                  (1),
       lineRateT1AndDS115M                    (2),
       lineRateT2AndDS26M                     (3),
       lineRateT3AndDS345M                    (4),
       lineRateE12m                           (5),
       lineRateE28m                           (6),
       lineRateE334m                          (7),
       lineRateE4140m                         (8),
       lineRateE5565m                         (9),
       lineRateVT15AndTU11VC11               (10),
       lineRateVT2AndTU12VC12                (11),
       lineRateVT6AndTU2VC2                  (12),
       lineRateLowOrderTu3Vc3                (13),
       lineRateSTS1AndAU3HighOrderVC3        (14),
       lineRateSTS3cAndAU4VC4                (15),
       lineRateSTS12cAndVC44c                (16),
       lineRateSTS48cAndVC416c               (17),
       lineRateSTS192cAndVC464c              (18),
       lineRateSectionOC1STS1AndRSSTM0       (19),
       lineRateSectionOC3STS3AndRSSTM1       (20),
       lineRateSectionOC12STS12AndRSSTM4     (21),
       lineRateSectionOC48STS48AndRSSTM16    (22),
       lineRateSectionOC192STS192AndRSSTM64  (23),
       lineRateLineOC1STS1AndMSSTM0          (24),
       lineRateLineOC3STS3AndMSSTM1          (25),
       lineRateLineOC12STS12AndMSSTM4        (26),
```

```
    lineRateLineOC48STS48AndMSSTM16        (27),
    lineRateLineOC192STS192AndMSSTM64      (28),
    lineRateOpticalChannel                 (40),
    lineRateOpticalMultiplexSection        (41),
    lineRateOpticalTransmissionSection     (42),
    lineRateAtmNi                          (43),
    lineRateAtmVp                          (44),
    lineRateAtmVc                          (45),
    lineRatePhysicalElectrical             (46),
    lineRatePhysicalOptical                (47),
    lineRatePhysicalMedialess              (48),
    lineRateOpticalSection                 (49),
    lineRateDigitalSignalRate              (50),
    lineRateAsyncFots150m                  (51),
    lineRateAsyncFots417m                  (52),
    lineRateAsyncFots560m                  (53),
    lineRateAsyncFots565m                  (54),
    lineRateAsyncFots1130m                 (55),
    lineRateAsyncFots1g7                   (56),
    lineRateAsyncFots1g8                   (57),
    lineRateD1Video                        (58),
    lineRateEscon                          (59),
    lineRateEtr                            (60),
    lineRateFastEthernet                   (61),
    lineRateFc12133m                       (62),
    lineRateFc25266m                       (63),
    lineRateFc50531m                       (64),
    lineRateFc1001063m                     (65),
    lineRateFddi                           (66),
    lineRateFicon                          (67),
    lineRateGigabitEthernet                (68),
    lineRateDs064k                         (69),
    lineRateIsdnBri                        (70),
    lineRatePots                           (71),
    lineRateDsrOc1Stm0                     (72),
    lineRateDsrOc3Stm1                     (73),
    lineRateDsrOc12Stm4                    (74),
    lineRateDsrOc24Stm8                    (75),
    lineRateDSROC48AndSTM16                (76),
    lineRateDSROC192AndSTM64               (77),
    lineRateDSROC768AndSTM256              (78),
    lineRateDsr15m                         (79),
    lineRateDsr2m                          (80),
    lineRateDsr6m                          (81),
    lineRateDsr8m                          (82),
    lineRateDsr34m                         (83),
    lineRateDsr45m                         (84),
    lineRateDsr140m                        (85),
    lineRateDsr565m                        (86),
    lineRateDsrGigabitEthernet             (87),
    lineRateSectionOC24STS24AndRSSTM8      (88),
    lineRateLineOC24STS24AndMSSTM8         (89),
    lineRateSectionOC768STS768AndRSSTM256  (90),
    lineRateLineOC768STS768AndMSSTM256     (91)
}

LayerRateStr     ::= DisplayString (SIZE (0..63))

Link             ::= DisplayString (SIZE (0..63))
```

```
Location         ::= DisplayString (SIZE (0..30))

Ne               ::= DisplayString (SIZE (0..63))

NeIpAddress      ::= IpAddress

NeIpAddressStr   ::= DisplayString (SIZE (7..15))

-- dd.mm.yyyy hh.mm.ss
NeTimeStr        ::= DisplayString (SIZE (19))

ProbableCause    ::= INTEGER
{
   unknown                              (1),
   ais                                  (2),
   aisRx                                (3),
   aisTx                                (4),
   lof                                  (5),
   lofRx                                (6),
   lofTx                                (7),
   lop                                  (8),
   los                                  (9),
   losSy                               (10),
   deg                                 (11),
   lom                                 (12),
   rdi                                 (13),
   tim                                 (14),
   plm                                 (15),
   uneq                                (16),
   exc                                 (17),
   csf                                 (18),
   ssf                                 (19),
   alarmInp                            (20),
   uFail                               (21),
   temp                                (22),
   fan                                 (23),
   pwrInA                              (24),
   pwrOut                              (25),
   modMis                              (26),
   modOut                              (27),
   modFail                             (28),
   msp                                 (29),
   pwrFail                             (30),
   wanDelay                            (31),
   seqFail                             (32),
   t0HoldOver                          (33),
   t0Defect                            (34),
   t4Squelch                           (35),
   inventoryFail                       (36),
   diagFail                            (37),
   inletFail                           (38),
   inletBitError                       (39),
   cardIsolated                        (40),
   cardAnomaly                         (41),
   hotSwapFailure                      (42),
   pwrInB                              (43),
   td                                  (44),
   tf                                  (45),
   unass                               (46),
   oof                                 (47),
```

```
   pwrUIL                              (48),
   pwrUIH                              (49),
   moduleFail                          (50),
   moduleMis                           (51),
   moduleLost                          (52),
   moduleShutdown                      (53),
   moduleRestart                       (54),
   demFail                             (55),
   freqFail                            (56),
   modulatorFail                       (57),
   rxFail                              (58),
   txFail                              (59),
   comm                                (60),
   pwrSwitch                           (61),
   modemSwitch                         (62),
   wanCapReduced                       (63),
   wanDown                             (64),
   epj                                 (65),
   orxo                                (66),
   switchWork                          (67),
   switchProt                          (68),
   rxOverflowHWFault                   (69),
   txOverflowHWFault                   (70),
   resetRequired                       (71),
   abortTftp                           (72),
   faultBackUp                         (73),
   forwardingTabOverflow               (74),
   framRelaySwitchConnectionDown       (75),
   framRelaySwitchConnectionUp         (76),
   errorsDuringInit                    (77),
   rsSDclientsTableOverflow            (78),
   rsSDinactiveServer                  (79),
   rsSnmpSetRequestInSpecialCfgState   (80),
   rsPingCompletion                    (81),
   rsWSDRedundancySwitch               (82),
   rsDhcpAllocationFailure             (83),
   rlIgmpTableOverflow                 (84),
   rlPimTableOverflow                  (85),
   routeTableOverflow                  (86),
   ipxRipTblOverflow                   (87),
   ipxSapTblOverflow                   (88),
   facsAccessVoilation                 (89),
   rsIpZhrConnectionsTableOverflow     (90),
   rsIpZhrReqStaticConnNotAccepted     (91),
   rsIpZhrVirtualIpAsSource            (92),
   rsIpZhrNotAllocVirtualIp            (93),
   pppSecurityViolation                (94),
   frDLCIStatudChange                  (95),
   chapFailedCommunication             (96),
   rlIpFftStnOverflow                  (97),
   rlIpFftSubOverflow                  (98),
   rlIpxFftStnOverflow                 (99),
   rlIpxFftSubOverflow                (100),
   rlIpmFftOverflow                   (101),
   papFailedCommunication             (102)
}

ProbableCauseQual ::= DisplayString (SIZE (0..255))

ProbableCauseStr  ::= DisplayString (SIZE (63))
```

```
    SequenceNumber   ::= INTEGER (1..4294967295)

    Severity         ::= INTEGER
    {
        indeterminate (0),
        critical      (1),
        major         (2),
        minor         (3),
        warning       (4),
        cleared       (5)
    }

    SeverityStr      ::= DisplayString (SIZE (5..13))

    Source           ::= DisplayString (SIZE (0..63))

    Trend            ::= INTEGER
    {
        lessSevere (0),
unChanged  (1),
        moreSevere (2)
    }

    TrendStr         ::= DisplayString (SIZE (0..11))

    TmnId            ::= DisplayString (SIZE (1..20))

-------------------------------------------------------------------------
---
-- Abbreviations:
-- GFMI     Generic Fault Manager Interface
-- HLM      Higher Leverl Manager
-- AXX 9800 TMN   AXXESSIT line of management products
-------------------------------------------------------------------------
---

-------------------------------------------------------------------------
---
-- GFMI objects groups
-------------------------------------------------------------------------
---
    gfmiCommand  OBJECT IDENTIFIER ::= { gfmiObject 1 }
    gfmiAlarm    OBJECT IDENTIFIER ::= { gfmiObject 2 }
    gfmiTraps     OBJECT IDENTIFIER ::= { gfmiObject 3 }


-------------------------------------------------------------------------
---
-- Command Group
-------------------------------------------------------------------------
---
    gfmiCommandSendActiveAlarms OBJECT-TYPE
        SYNTAX      SequenceNumber
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "If set different to 0, the agent will send all active alarms to
the
```

```
             requesting manager. One-shot (stateless) alarms and events are
not sent.
             The requester must be one of the registered managers. Otherwise
            the command is ignored. The traps are only sent to the requesting
            manager, not the other registered managers. Requests are queued,
             i.e. a manager may write to this variable while another request
             from that or another manager is in progress. The value is reset
to
             0 when all requests have been processed."
         ::= { gfmiCommand 1 }

    gfmiCommandRetransmitOne OBJECT-TYPE
        SYNTAX      SequenceNumber
        MAX-ACCESS  write-only
        STATUS      current
        DESCRIPTION
           "If set different to 0, the agent will re-transmit the Alarm/Event
             having this HLM-ID (sequence number). If not present in the AXX
9800 TMN
             database, nothing is sent.
             The requester must be one of the registered managers. Otherwise
            the command is ignored. The traps are only sent to the requesting
            manager, not the other registered managers. Requests are queued,
             i.e. a manager may write to this variable while another request
             from that or another manager is in progress. The value is reset
to
             0 when all requests have been processed."
         ::= { gfmiCommand 2 }

    gfmiCommandRetransmitMultiple OBJECT-TYPE
        SYNTAX      SequenceNumber
        MAX-ACCESS  write-only
        STATUS      current
        DESCRIPTION
            "If set different to 0, the agent will re-transmit all
Alarms/Events
             having this HLM-ID (sequence number) and higher. If not present
in
             the AXX 9800 TMN database, all newer Alarm/events are.
             The requester must be one of the registered managers. Otherwise
            the command is ignored. The traps are only sent to the requesting
            manager, not the other registered managers. Requests are queued,
             i.e. a manager may write to this variable while another request
             from that or another manager is in progress. The value is reset
to
             0 when all requests have been processed.
             This mechanism is protected with respect to wrap-around
situations."
         ::= { gfmiCommand 3 }

-------------------------------------------------------------------------
---
-- Alarm Groups
-------------------------------------------------------------------------
---
    gfmiAlarmObjects    OBJECT IDENTIFIER ::= { gfmiAlarm 1 }
    gfmiAlarmTables     OBJECT IDENTIFIER ::= { gfmiAlarm 2 }

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

```
-- Alarm Objects
-------------------------------------------------------------------------------
---
        gfmiAlarmLastHlmId OBJECT-TYPE
        SYNTAX     SequenceNumber
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "This object contains the HLM-ID (sequence number) of the last
            trap transmitted."
        ::= { gfmiAlarmObjects 1 }

-------------------------------------------------------------------------------
---
-- Alarm Table. This table contains all active alarms. Note that the AXX 9800
TMN has
-- both stateless (one-shot) and stateful (on-off) alarms. The alarm table
-- contains stateful alarms only.
-------------------------------------------------------------------------------
---
    gfmiAlarmTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF GfmiAlarmEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table contains all active alarms"
        ::= { gfmiAlarmTables 1 }

    gfmiAlarmEntry OBJECT-TYPE
        SYNTAX     GfmiAlarmEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "One entry per active alarm,"
        INDEX { gfmiAlarmHlmId }
        ::= { gfmiAlarmTable 1 }

    GfmiAlarmEntry ::= SEQUENCE
    {
        gfmiAlarmHlmId             SequenceNumber,
        gfmiAlarmAXX 9800 TMNId          TmnId,
        gfmiAlarmAcked             Acked,
        gfmiAlarmAckedStr          AckedStr,
        gfmiAlarmAckSignature      AckSignature,
        gfmiAlarmAckTime           DateAndTime,
        gfmiAlarmAckTimeStr        AckTimeStr,
        gfmiAlarmSeverity          Severity,
        gfmiAlarmSeverityStr       SeverityStr,
        gfmiAlarmNeIpAddress       NeIpAddress,
        gfmiAlarmNeIpAddressStr    NeIpAddressStr,
        gfmiAlarmNe                Ne,
        gfmiAlarmSource            Source,
        gfmiAlarmNeTime            DateAndTime,
        gfmiAlarmNeTimeStr         NeTimeStr,
        gfmiAlarmEmsTime           DateAndTime,
        gfmiAlarmEmsTimeStr        EmsTimeStr,
        gfmiAlarmProbableCause     ProbableCause,
        gfmiAlarmProbableCauseStr  ProbableCauseStr,
        gfmiAlarmProbableCauseQual ProbableCauseQual,
        gfmiAlarmClearable         Clearable,
```

```
        gfmiAlarmClearableStr      ClearableStr,
        gfmiAlarmAlarmType         AlarmType,
        gfmiAlarmAlarmTypeStr      AlarmTypeStr,
        gfmiAlarmLocation          Location,
        gfmiAlarmComments          Comments,
        gfmiAlarmAlarmId           AlarmId,
        gfmiAlarmTrend             Trend,
        gfmiAlarmTrendStr          TrendStr,
        gfmiAlarmLink              Link,
        gfmiAlarmLayerRate         LayerRate,
        gfmiAlarmLayerRateStr      LayerRateStr,
        gfmiAlarmAddTxt            AddTxt,
        gfmiAlarmCompoundTxt       CompoundTxt,
gfmiMaxAlarmSeverity       Severity,
gfmiMaxAlarmSeverityStr    SeverityStr
    }

    gfmiAlarmHlmId OBJECT-TYPE
        SYNTAX     SequenceNumber
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
           "The HLM (Higher-Level Manager) alarm sequence number. This number
            should be used by the HLM to monitor trap loss. When a trap has
been
            lost, this value may be used to initiate a re-transmission.
            The difference between the AXX 9800 TMN and HLM sequence number
is that
            the AXX 9800 TMN sequence number counts all alarms/events while
the HLM
            sequence number counts the alarms/events actually sent to the
HLM(s)
            after filtering (if defined)."
        ::= { gfmiAlarmEntry 1 }

    gfmiAlarmAXX 9800 TMNId OBJECT-TYPE
        SYNTAX     TmnId
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The AXX 9800 TMN internal alarm/event sequence number. Shall be
            used in alarm scoped launching of AXX 9800 TMN applications."
        ::= { gfmiAlarmEntry 2 }

    gfmiAlarmAcked OBJECT-TYPE
        SYNTAX     Acked
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Says whether the alarm has been acknowledged or not (enum
variant)."
        ::= { gfmiAlarmEntry 3 }

    gfmiAlarmAckedStr OBJECT-TYPE
        SYNTAX     AckedStr
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Says whether the alarm has been acknowledged or not (string
variant)."
```

```
        ::= { gfmiAlarmEntry 4 }

    gfmiAlarmAckSignature OBJECT-TYPE
        SYNTAX     AckSignature
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Says who did acknowledge the alarm."
        ::= { gfmiAlarmEntry 5 }

    gfmiAlarmAckTime OBJECT-TYPE
        SYNTAX     DateAndTime
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Says when the alarm was acknowledged (standard SNMP
representation)."
        ::= { gfmiAlarmEntry 6 }

    gfmiAlarmAckTimeStr OBJECT-TYPE
        SYNTAX     AckTimeStr
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Says when the alarm was acknowledged (string variant)."
        ::= { gfmiAlarmEntry 7 }

    gfmiAlarmSeverity OBJECT-TYPE
        SYNTAX     Severity
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Alarm severity (enum variant)."
        ::= { gfmiAlarmEntry 8 }

    gfmiAlarmSeverityStr OBJECT-TYPE
        SYNTAX     SeverityStr
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Alarm severity (string variant)."
        ::= { gfmiAlarmEntry 9 }

    gfmiAlarmNeIpAddress OBJECT-TYPE
        SYNTAX     NeIpAddress
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "NE IP Address (standard SNMP format)."
        ::= { gfmiAlarmEntry 10 }

    gfmiAlarmNeIpAddressStr OBJECT-TYPE
        SYNTAX     NeIpAddressStr
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "NE IP Address (string variant)."
        ::= { gfmiAlarmEntry 11 }

    gfmiAlarmNe OBJECT-TYPE
```

```
        SYNTAX     Ne
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "NE name."
      ::= { gfmiAlarmEntry 12 }

    gfmiAlarmSource OBJECT-TYPE
        SYNTAX     Source
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The managed object subject to alarm."
      ::= { gfmiAlarmEntry 13 }

    gfmiAlarmNeTime OBJECT-TYPE
        SYNTAX     DateAndTime
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Date/Time the alarm was registered on the NE (standard format)."
      ::= { gfmiAlarmEntry 14 }

    gfmiAlarmNeTimeStr OBJECT-TYPE
        SYNTAX     NeTimeStr
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Date/Time the alarm was registered on the NE (string variant)."
      ::= { gfmiAlarmEntry 15 }

    gfmiAlarmEmsTime OBJECT-TYPE
        SYNTAX     DateAndTime
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Date/Time the alarm was registered on the AXX 9800 TMN (standard
format)."
      ::= { gfmiAlarmEntry 16 }

    gfmiAlarmEmsTimeStr OBJECT-TYPE
        SYNTAX     EmsTimeStr
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Date/Time the alarm was registered on the AXX 9800 TMN (string
variant)."
      ::= { gfmiAlarmEntry 17 }

    gfmiAlarmProbableCause OBJECT-TYPE
        SYNTAX     ProbableCause
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Probable cause alarm parameter (enum variant)."
    ::= { gfmiAlarmEntry 18 }

    gfmiAlarmProbableCauseStr OBJECT-TYPE
        SYNTAX     ProbableCauseStr
        MAX-ACCESS read-only
```

```
        STATUS     current
        DESCRIPTION
            "Probable cause alarm parameter (string variant)."
        ::= { gfmiAlarmEntry 19 }

    gfmiAlarmProbableCauseQual OBJECT-TYPE
        SYNTAX     ProbableCauseQual
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Alarm description text."
        ::= { gfmiAlarmEntry 20 }

    gfmiAlarmClearable OBJECT-TYPE
        SYNTAX     Clearable
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Says whether the alarm is clearable or not, i.e. stateful
            or stateless (enum variant). All entries in this table is
            stateful, but the variable is included here because the alarm
            entry definition also serves for the trap definitions."
      ::= { gfmiAlarmEntry 21 }

    gfmiAlarmClearableStr OBJECT-TYPE
        SYNTAX     ClearableStr
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Says whether the alarm is clearable or not (string variant)."
      ::= { gfmiAlarmEntry 22 }

    gfmiAlarmAlarmType OBJECT-TYPE
        SYNTAX     AlarmType
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Alarm Type (enum variant)."
        ::= { gfmiAlarmEntry 23 }

    gfmiAlarmAlarmTypeStr OBJECT-TYPE
        SYNTAX     AlarmTypeStr
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Alarm Type (string variant)."
        ::= { gfmiAlarmEntry 24 }

    gfmiAlarmLocation OBJECT-TYPE
        SYNTAX     Location
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Slot/Port/CBKLM value."
        ::= { gfmiAlarmEntry 25 }

    gfmiAlarmComments OBJECT-TYPE
        SYNTAX     Comments
        MAX-ACCESS read-only
        STATUS     current
```

```
    DESCRIPTION
        "A user comment on the alarm."
    ::= { gfmiAlarmEntry 26 }

gfmiAlarmAlarmId OBJECT-TYPE
    SYNTAX     AlarmId
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "This field contains a unique ID of this alarm,
        the format is typically 2.1.5 (string)."
    ::= { gfmiAlarmEntry 27 }

gfmiAlarmTrend OBJECT-TYPE
    SYNTAX     Trend
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "More/less severe (enum variant)."
    ::= { gfmiAlarmEntry 28 }

gfmiAlarmTrendStr OBJECT-TYPE
    SYNTAX     TrendStr
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "More/less severe (string variant)."
    ::= { gfmiAlarmEntry 29 }

gfmiAlarmLink OBJECT-TYPE
    SYNTAX     Link
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "Identification of the affected link."
    ::= { gfmiAlarmEntry 30 }

gfmiAlarmLayerRate OBJECT-TYPE
    SYNTAX     LayerRate
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "Data rate of the affected interface (enum variant)."
    ::= { gfmiAlarmEntry 31 }

gfmiAlarmLayerRateStr OBJECT-TYPE
    SYNTAX     LayerRateStr
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "Data rate of the affected interface (enum variant)."
    ::= { gfmiAlarmEntry 32 }

gfmiAlarmAddTxt OBJECT-TYPE
    SYNTAX     AddTxt
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "Additional text."
    ::= { gfmiAlarmEntry 33 }
```

```
    gfmiAlarmCompoundTxt OBJECT-TYPE
        SYNTAX     CompoundTxt
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "User defined text, optionally containing a numer of the above
            alarm parameters."
        ::= { gfmiAlarmEntry 34 }

gfmiMaxAlarmSeverity OBJECT-TYPE
        SYNTAX     Severity
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Max Alarm severity for this network element(enum variant)."
        ::= { gfmiAlarmEntry 35 }

    gfmiMaxAlarmSeverityStr OBJECT-TYPE
        SYNTAX     SeverityStr
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Max Alarm severity for this network element (string variant)."
        ::= { gfmiAlarmEntry 36 }


-------------------------------------------------------------------------
---
-- Alarm and Event Traps
-------------------------------------------------------------------------
---
    gfmiTrapsFmt1    OBJECT IDENTIFIER ::= { gfmiTraps 1 }
    gfmiTrapsFmt2    OBJECT IDENTIFIER ::= { gfmiTraps 2 }
    gfmiTrapsFmt3    OBJECT IDENTIFIER ::= { gfmiTraps 3 }
    gfmiTrapObjects  OBJECT IDENTIFIER ::= { gfmiTraps 4 }

-------------------------------------------------------------------------
---
-- Format #1 uses enums where possible. For programmatic access. Here,
strings
-- are used only when necessary.
-------------------------------------------------------------------------
---
--    gfmiTrapAlarmRaisedFmt1 NOTIFICATION-TYPE
--        OBJECTS
--        {
--            gfmiAlarmAXX 9800 TMNId,
--            gfmiAlarmHlmId,
--            gfmiAlarmAcked,
--            gfmiAlarmAckSignature,
--            gfmiAlarmAckTime,
--            gfmiAlarmSeverity,
--            gfmiAlarmNeIpAddress,
--            gfmiAlarmNe,
--            gfmiAlarmSource,
--            gfmiAlarmNeTime,
--            gfmiAlarmEmsTime,
--            gfmiAlarmProbableCause,
--            gfmiAlarmProbableCauseStr,
```

```
--            gfmiAlarmProbableCauseQual,
--            gfmiAlarmClearable,
--            gfmiAlarmAlarmType,
--            gfmiAlarmLocation,
--            gfmiAlarmComments,
--            gfmiAlarmAlarmId,
--            gfmiAlarmTrend,
--            gfmiAlarmLink,
--            gfmiAlarmLayerRate,
--            gfmiAlarmAddTxt
--        }
--        STATUS  current
--        DESCRIPTION
--            "Alarm raised trap."
--        ::= { gfmiTrapsFmt1 1 }

--    gfmiTrapAlarmClearedFmt1 NOTIFICATION-TYPE
--        OBJECTS
--        {
--            gfmiAlarmAXX 9800 TMNId,
--            gfmiAlarmHlmId,
--            gfmiAlarmAcked,
--            gfmiAlarmAckSignature,
--            gfmiAlarmAckTime,
--            gfmiAlarmSeverity,
--            gfmiAlarmNeIpAddress,
--            gfmiAlarmNe,
--            gfmiAlarmSource,
--            gfmiAlarmNeTime,
--            gfmiAlarmEmsTime,
--            gfmiAlarmProbableCause,
--            gfmiAlarmProbableCauseStr,
--            gfmiAlarmProbableCauseQual,
--            gfmiAlarmClearable,
--            gfmiAlarmAlarmType,
--            gfmiAlarmLocation,
--            gfmiAlarmComments,
--            gfmiAlarmAlarmId,
--            gfmiAlarmTrend,
--            gfmiAlarmLink,
--            gfmiAlarmLayerRate,
--            gfmiAlarmAddTxt
--        }
--        STATUS  current
--        DESCRIPTION
--            "Alarm cleared trap."
--        ::= { gfmiTrapsFmt1 2 }

--    gfmiTrapEventFmt1 NOTIFICATION-TYPE
--        OBJECTS
--        {
--            gfmiAlarmAXX 9800 TMNId,
--            gfmiAlarmHlmId,
--            gfmiAlarmNeIpAddress,
--            gfmiAlarmNe,
--            gfmiAlarmSource,
--            gfmiAlarmNeTime,
--            gfmiAlarmEmsTime,
--            gfmiEventType,
--            gfmiEventDescription,
```

```
--            gfmiEventId
--        }
--        STATUS  current
--        DESCRIPTION
--            "Event trap."
--        ::= { gfmiTrapsFmt1 3 }


-------------------------------------------------------------------------
---
-- Format #2 is equal to Format#1 but has supplementary strings strings for
all
-- enums. For use with e.g. OpenView.
-------------------------------------------------------------------------
---

--    gfmiTrapAlarmRaisedFmt2 NOTIFICATION-TYPE
--        OBJECTS
--        {
--            gfmiAlarmAXX 9800 TMNId,
--            gfmiAlarmHlmId,
--            gfmiAlarmAcked,
--            gfmiAlarmAckedStr,
--            gfmiAlarmAckSignature,
--            gfmiAlarmAckTime,
--            gfmiAlarmAckTimeStr,
--            gfmiAlarmSeverity,
--            gfmiAlarmSeverityStr,
--            gfmiAlarmNeIpAddress,
--            gfmiAlarmNeIpAddressStr,
--            gfmiAlarmNe,
--            gfmiAlarmSource,
--            gfmiAlarmNeTime,
--            gfmiAlarmNeTimeStr,
--            gfmiAlarmEmsTime,
--            gfmiAlarmEmsTimeStr,
--            gfmiAlarmProbableCause,
--            gfmiAlarmProbableCauseStr,
--            gfmiAlarmProbableCauseQual,
--            gfmiAlarmClearable,
--            gfmiAlarmClearableStr,
--            gfmiAlarmAlarmType,
--            gfmiAlarmAlarmTypeStr,
--            gfmiAlarmLocation,
--            gfmiAlarmComments,
--            gfmiAlarmAlarmId,
--            gfmiAlarmTrend,
--            gfmiAlarmTrendStr,
--            gfmiAlarmLink,
--            gfmiAlarmLayerRate,
--            gfmiAlarmLayerRateStr,
--            gfmiAlarmAddTxt
--        }
--        STATUS  current
--        DESCRIPTION
--            "Alarm raised trap."
--        ::= { gfmiTrapsFmt2 1 }

--   gfmiTrapAlarmClearedFmt2 NOTIFICATION-TYPE
--        OBJECTS
```

```
--        {
--            gfmiAlarmAXX 9800 TMNId,
--            gfmiAlarmHlmId,
--            gfmiAlarmAcked,
--            gfmiAlarmAckedStr,
--            gfmiAlarmAckSignature,
--            gfmiAlarmAckTime,
--            gfmiAlarmAckTimeStr,
--            gfmiAlarmSeverity,
--            gfmiAlarmSeverityStr,
--            gfmiAlarmNeIpAddress,
--            gfmiAlarmNeIpAddressStr,
--            gfmiAlarmNe,
--            gfmiAlarmSource,
--            gfmiAlarmNeTime,
--            gfmiAlarmNeTimeStr,
--            gfmiAlarmEmsTime,
--            gfmiAlarmEmsTimeStr,
--            gfmiAlarmProbableCause,
--            gfmiAlarmProbableCauseStr,
--            gfmiAlarmProbableCauseQual,
--            gfmiAlarmClearable,
--            gfmiAlarmClearableStr,
--            gfmiAlarmAlarmType,
--            gfmiAlarmAlarmTypeStr,
--            gfmiAlarmLocation,
--            gfmiAlarmComments,
--            gfmiAlarmAlarmId,
--            gfmiAlarmTrend,
--            gfmiAlarmTrendStr,
--            gfmiAlarmLink,
--            gfmiAlarmLayerRate,
--            gfmiAlarmLayerRateStr,
--            gfmiAlarmAddTxt
--        }
--        STATUS  current
--        DESCRIPTION
--            "Alarm cleared trap."
--        ::= { gfmiTrapsFmt2 2 }

--   gfmiTrapEventFmt2 NOTIFICATION-TYPE
--        OBJECTS
--        {
--            gfmiAlarmAXX 9800 TMNId,
--            gfmiAlarmHlmId,
--            gfmiAlarmNeIpAddress,
--            gfmiAlarmNeIpAddressStr,
--            gfmiAlarmNe,
--            gfmiAlarmSource,
--            gfmiAlarmNeTime,
--            gfmiAlarmNeTimeStr,
--            gfmiAlarmEmsTime,
--            gfmiAlarmEmsTimeStr,
--            gfmiEventType,
--            gfmiEventTypeStr,
--            gfmiEventDescription,
--            gfmiEventId
--        }
--        STATUS  current
--        DESCRIPTION
```

```
--            "Event trap."
--        ::= { gfmiTrapsFmt2 3 }

-------------------------------------------------------------------------
---
-- Format #3 uses a user defined compound string.
-------------------------------------------------------------------------
---

--    gfmiTrapAlarmRaisedFmt3 NOTIFICATION-TYPE
--        OBJECTS
--        {
--            gfmiAlarmAXX 9800 TMNId,
--            gfmiAlarmHlmId,
--            gfmiAlarmCompoundTxt
--        }
--        STATUS  current
--        DESCRIPTION
--            "Alarm raised trap."
--        ::= { gfmiTrapsFmt3 1 }

--   gfmiTrapAlarmClearedFmt3 NOTIFICATION-TYPE
--        OBJECTS
--        {
--            gfmiAlarmAXX 9800 TMNId,
--            gfmiAlarmHlmId,
--            gfmiAlarmCompoundTxt
--        }
--        STATUS  current
--        DESCRIPTION
--            "Alarm cleared trap."
--        ::= { gfmiTrapsFmt3 2 }

--   gfmiTrapEventFmt3 NOTIFICATION-TYPE
--        OBJECTS
--        {
--            gfmiAlarmAXX 9800 TMNId,
--            gfmiAlarmHlmId,
--            gfmiAlarmNeIpAddress,
--            gfmiAlarmNeIpAddressStr,
--            gfmiAlarmCompoundTxt
--        }
--        STATUS  current
--        DESCRIPTION
--            "Event trap."
--        ::= { gfmiTrapsFmt3 3 }
-------------------------------------------------------------------------
-- Trap definitions
-------------------------------------------------------------------------
-------------------------------------------------------------------------
---
-- Format #1 uses enums where possible. For programmatic access. Here,
strings
-- are used only when necessary.
-------------------------------------------------------------------------
---
    gfmiTrapAlarmRaisedFmt1 TRAP-TYPE
        ENTERPRISE gfmiTrapsFmt1
        VARIABLES
        {
```

```
            gfmiAlarmAXX 9800 TMNId,
            gfmiAlarmHlmId,
            gfmiAlarmAcked,
            gfmiAlarmAckSignature,
            gfmiAlarmAckTime,
            gfmiAlarmSeverity,
            gfmiAlarmNeIpAddress,
            gfmiAlarmNe,
            gfmiAlarmSource,
            gfmiAlarmNeTime,
            gfmiAlarmEmsTime,
            gfmiAlarmProbableCause,
            gfmiAlarmProbableCauseStr,
            gfmiAlarmProbableCauseQual,
            gfmiAlarmClearable,
            gfmiAlarmAlarmType,
            gfmiAlarmLocation,
            gfmiAlarmComments,
            gfmiAlarmAlarmId,
            gfmiAlarmTrend,
            gfmiAlarmLink,
            gfmiAlarmLayerRate,
            gfmiAlarmAddTxt,
gfmiMaxAlarmSeverity
        }
        DESCRIPTION
            "Alarm raised trap."
        ::= 1

    gfmiTrapAlarmClearedFmt1 TRAP-TYPE
        ENTERPRISE gfmiTrapsFmt1
        VARIABLES
        {
            gfmiAlarmAXX 9800 TMNId,
            gfmiAlarmHlmId,
            gfmiAlarmAcked,
            gfmiAlarmAckSignature,
            gfmiAlarmAckTime,
            gfmiAlarmSeverity,
            gfmiAlarmNeIpAddress,
            gfmiAlarmNe,
            gfmiAlarmSource,
            gfmiAlarmNeTime,
            gfmiAlarmEmsTime,
            gfmiAlarmProbableCause,
            gfmiAlarmProbableCauseStr,
            gfmiAlarmProbableCauseQual,
            gfmiAlarmClearable,
            gfmiAlarmAlarmType,
            gfmiAlarmLocation,
            gfmiAlarmComments,
            gfmiAlarmAlarmId,
            gfmiAlarmTrend,
            gfmiAlarmLink,
            gfmiAlarmLayerRate,
            gfmiAlarmAddTxt,
gfmiMaxAlarmSeverity
        }
        DESCRIPTION
            "Alarm cleared trap."
```

```
        ::= 2

    gfmiTrapEventFmt1 TRAP-TYPE
        ENTERPRISE gfmiTrapsFmt1
        VARIABLES
        {
            gfmiAlarmAXX 9800 TMNId,
            gfmiAlarmHlmId,
            gfmiAlarmNeIpAddress,
            gfmiAlarmNe,
            gfmiAlarmSource,
            gfmiAlarmNeTime,
            gfmiAlarmEmsTime,
            gfmiEventType,
            gfmiEventDescription,
            gfmiEventId
        }
        DESCRIPTION
            "Event trap."
        ::= 3

-------------------------------------------------------------------------------
---
-- Format #2 is equal to Format#1 but has supplementary strings strings for
all
-- enums. For use with e.g. OpenView.
-------------------------------------------------------------------------------
---
    gfmiTrapAlarmRaisedFmt2 TRAP-TYPE
        ENTERPRISE gfmiTrapsFmt2
        VARIABLES
        {
            gfmiAlarmAXX 9800 TMNId,
            gfmiAlarmHlmId,
            gfmiAlarmAcked,
            gfmiAlarmAckedStr,
            gfmiAlarmAckSignature,
            gfmiAlarmAckTime,
            gfmiAlarmAckTimeStr,
            gfmiAlarmSeverity,
            gfmiAlarmSeverityStr,
            gfmiAlarmNeIpAddress,
            gfmiAlarmNeIpAddressStr,
            gfmiAlarmNe,
            gfmiAlarmSource,
            gfmiAlarmNeTime,
            gfmiAlarmNeTimeStr,
            gfmiAlarmEmsTime,
            gfmiAlarmEmsTimeStr,
            gfmiAlarmProbableCause,
            gfmiAlarmProbableCauseStr,
            gfmiAlarmProbableCauseQual,
            gfmiAlarmClearable,
            gfmiAlarmClearableStr,
            gfmiAlarmAlarmType,
            gfmiAlarmAlarmTypeStr,
            gfmiAlarmLocation,
            gfmiAlarmComments,
            gfmiAlarmAlarmId,
            gfmiAlarmTrend,
```

```
            gfmiAlarmTrendStr,
            gfmiAlarmLink,
            gfmiAlarmLayerRate,
            gfmiAlarmLayerRateStr,
            gfmiAlarmAddTxt,
gfmiMaxAlarmSeverity,
gfmiMaxAlarmSeverityStr
        }
        DESCRIPTION
            "Alarm raised trap."
        ::= 1

   gfmiTrapAlarmClearedFmt2 TRAP-TYPE
        ENTERPRISE gfmiTrapsFmt2
        VARIABLES
        {
            gfmiAlarmAXX 9800 TMNId,
            gfmiAlarmHlmId,
            gfmiAlarmAcked,
            gfmiAlarmAckedStr,
            gfmiAlarmAckSignature,
            gfmiAlarmAckTime,
            gfmiAlarmAckTimeStr,
            gfmiAlarmSeverity,
            gfmiAlarmSeverityStr,
            gfmiAlarmNeIpAddress,
            gfmiAlarmNeIpAddressStr,
            gfmiAlarmNe,
            gfmiAlarmSource,
            gfmiAlarmNeTime,
            gfmiAlarmNeTimeStr,
            gfmiAlarmEmsTime,
            gfmiAlarmEmsTimeStr,
            gfmiAlarmProbableCause,
            gfmiAlarmProbableCauseStr,
            gfmiAlarmProbableCauseQual,
            gfmiAlarmClearable,
            gfmiAlarmClearableStr,
            gfmiAlarmAlarmType,
            gfmiAlarmAlarmTypeStr,
            gfmiAlarmLocation,
            gfmiAlarmComments,
            gfmiAlarmAlarmId,
            gfmiAlarmTrend,
            gfmiAlarmTrendStr,
            gfmiAlarmLink,
            gfmiAlarmLayerRate,
            gfmiAlarmLayerRateStr,
            gfmiAlarmAddTxt,
gfmiMaxAlarmSeverity,
gfmiMaxAlarmSeverityStr
        }
        DESCRIPTION
            "Alarm cleared trap."
        ::= 2

   gfmiTrapEventFmt2 TRAP-TYPE
        ENTERPRISE gfmiTrapsFmt2
        VARIABLES
        {
```

```
            gfmiAlarmAXX 9800 TMNId,
            gfmiAlarmHlmId,
            gfmiAlarmNeIpAddress,
            gfmiAlarmNeIpAddressStr,
            gfmiAlarmNe,
            gfmiAlarmSource,
            gfmiAlarmNeTime,
            gfmiAlarmNeTimeStr,
            gfmiAlarmEmsTime,
            gfmiAlarmEmsTimeStr,
            gfmiEventType,
            gfmiEventTypeStr,
            gfmiEventDescription,
            gfmiEventId
        }
        DESCRIPTION
            "Event trap."
        ::= 3

-------------------------------------------------------------------------------
---
-- Format #3 uses a user defined compound string.
-------------------------------------------------------------------------------
---


       gfmiTrapAlarmRaisedFmt3 TRAP-TYPE
            ENTERPRISE gfmiTrapsFmt3
            VARIABLES
            {
                gfmiAlarmAXX 9800 TMNId,
                gfmiAlarmHlmId,
                gfmiAlarmNeIpAddress,
                gfmiAlarmSeverity,
                gfmiAlarmCompoundTxt,
   gfmiMaxAlarmSeverity
            }
            DESCRIPTION
                "Alarm raised trap."
            ::= 1

      gfmiTrapAlarmClearedFmt3 TRAP-TYPE
            ENTERPRISE gfmiTrapsFmt3
            VARIABLES
            {
                gfmiAlarmAXX 9800 TMNId,
                gfmiAlarmHlmId,
                gfmiAlarmNeIpAddress,
                gfmiAlarmSeverity,
                gfmiAlarmCompoundTxt,
   gfmiMaxAlarmSeverity
            }
            DESCRIPTION
                "Alarm cleared trap."
            ::= 2

      gfmiTrapEventFmt3 TRAP-TYPE
            ENTERPRISE gfmiTrapsFmt3
            VARIABLES
            {
```

```
                gfmiAlarmAXX 9800 TMNId,
                gfmiAlarmHlmId,
                gfmiAlarmNeIpAddress,
                gfmiAlarmSeverity,
                gfmiAlarmCompoundTxt
            }
            DESCRIPTION
                "Event trap."
        ::= 3

-------------------------------------------------------------------------------
---
-- Trap objects not defined earlier (event stuff).
-------------------------------------------------------------------------------
---
    gfmiEventType OBJECT-TYPE
        SYNTAX     EventType
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Event Type (enum variant)."
        ::= { gfmiTrapObjects 1 }

    gfmiEventTypeStr OBJECT-TYPE
        SYNTAX     EventTypeStr
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Event Type."
        ::= { gfmiTrapObjects 2 }

    gfmiEventDescription OBJECT-TYPE
        SYNTAX     Description
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Event Description text."
        ::= { gfmiTrapObjects 3 }

    gfmiEventId OBJECT-TYPE
        SYNTAX     EventId
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "This field contains a unique ID of this alarm,
            the format is typically 2.1.10 (string)."
        ::= { gfmiTrapObjects 4 }

END
```