Bull

S@N.IT!
User’s Guide

AIX, Linux, Solaris, Windows, GCOS7, GCOS8

ORDER REFERENCE
86 A2 59EF 06

Bull

S@N.IT!
User's Guide

AIX, Linux, Solaris, Windows, GCOS7, GCOS8

Software

December 2004

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

ORDER REFERENCE
86 A2 59EF 06

Printed in France

Suggestions and criticisms concerning the form, content, and presentation of this book are invited. A form is provided at the end of this book for this purpose.

To order additional copies of this book or other Bull Technical Publications, you are invited to use the Ordering Form also provided at the end of this book.

Trademarks and Acknowledgements

About This Book

This book will help you to install and use the S@N.IT! software on the AIX, Windows, Solaris, Linux, GCOS7 and GCOS8 servers involved in the application.

Who Should Use This Book

This book is written for the administrator who is to manage the SAN infrastructure.

Overview

This book is organized as follows:

- Chapter 1. Introduction
- Chapter 2. S@N.IT! Data Center Features
- Chapter 3. S@N.IT! Limited Edition Features
- Chapter 4. Installation Overview
- Chapter 5. Configuration and Launching
- Chapter 6. S@N.IT! Graphical User Interface (GUI)
- Chapter 7. S@N.IT! Command Line Interface (CLI)
- Chapter 8. S@N.IT! SNMP Agent
- Chapter 9. Best Practices
- Chapter 10. Troubleshooting
- Chapter 11. Supported SAN Components
- Chapter 12. Supported Platforms
- Glossary

Online Documentation

You can also find the present guide on the “Hypertext Library for AIX” CD-ROM. This online documentation is designed for use with an HTML version 3.2 compatible web browser.

Table of Contents

Table of Figures

Chapter 1. Introduction

This chapter is an introduction to the main concepts of a SAN. It provides the following information:

SAN Overview

A Storage Area Network, or SAN, is a high–speed network based on Fibre Channel technology. This 1 or 2 Gb/s data transfer interface maps several transport protocols including IP and SCSI, and hence allows you to merge high–speed I/O and networking functionality in a single connectivity technology.

The SAN connects servers and large storage subsystems, and therefore is mostly dedicated to high–speed data transfers between these servers and the storage units.

SANs enable storage points to be distributed around the network at the level of the company site. Servers can then access storage peripherals several hundred meters away with response times comparable to local, private connections.

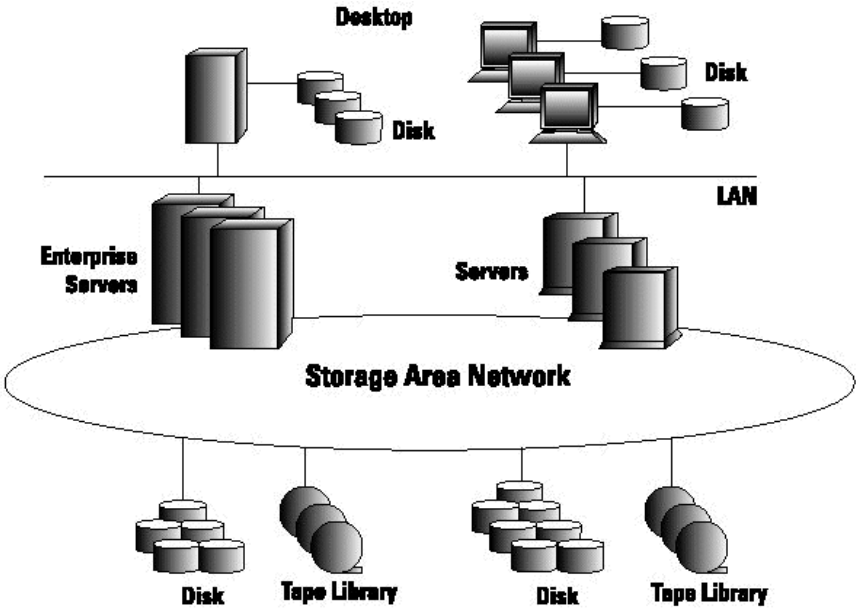

Figure 1. SAN overview

SANs provide major improvements to storage management strategies:

- Distributed storage resources: any host on the SAN may view and access any storage system on the network.
- Scalability: new storage resources or new servers may be easily added onto the SAN; each server may increase its own storage resources at any time.
- Performance: Fibre Channel switches allow for modular increases in bandwidth. If the nominal speed of a Fibre Channel link is 1 or 2 Gbs, hundreds of independent data paths can be installed on a SAN, leading to almost unlimited bandwidth.
- Integrity and Availability: access paths to storage peripherals can be easily multiplied and the storage space allocated to each application server connected to the SAN can be duplicated. Gateways provide communication with remote SANs, opening the way for complete disaster protection solutions.

SAN Topology

Bull's SANs interconnect servers, disks arrays, tape libraries, through different modes of connections:

- Point–to–point, with a single cable between a HBA (Host Bus Adapter) and a disk array port.
- Loop, through a hub which creates a (private) loop technology, allowing communications between all HBA and Disk Arrays port connected to the hub.
- Fabric, through a switch (or a set of switches interconnected by Inter–Switch Links – ISLs), which allows the creation of direct communication links (with full bandwidth availability for each of them) between all pairs of devices connected to it.

Solutions also exist to re–connect SCSI devices (Disk Arrays or Libraries) to a SAN using SCSI–Fibre Channel bridges.

SAN Management Needs

SAN networks have many advantages in terms of flexibility, scalability of bandwidth and storage resources. They also bring new management requirements:

- As the number of interconnected devices increases, it is necessary to check the network topology and the communication links between the Servers and the Storage devices.
- A centralized administration point as each SAN device has its own vendor–specific management tool.
- Network fault detection and localization.
- Network and devices change control
- Data protection: all servers connected to a SAN can access all LUNs of all storage devices connected to this SAN. However LUNs should not be shared between hosts, except in special cases such as cluster configurations. This is why a LUN Access Control mechanism is needed to prevent risks of data corruption from other operating systems.

Bull's S@N.IT! provides a solution to these requirements.

Summary of S@N.IT! Features

Topology and Devices Discovery

S@N.IT! automatically discovers the devices connected to the SAN and the links between them. It refreshes this information each time the SAN configuration is modified. S@N.IT! displays the SAN topology providing the S@N.IT! administrator with information about:

- physical links between all devices,
- logical paths between all devices,
- device properties.

The S@N.IT! administrator can also complete the automatic discovery by manually adding devices or editing device properties (to give an understandable name for example). See *SAN Topology discovery and display*, on page 2-1.

Configuration Changes

S@N.IT! keeps a track of all changes whether they impact the network topology (new connection for example) or the property of a device (firmware upgrade for example) thus giving the S@N.IT! administrator a way to detect and understand the evolution of his storage network and resources.

Monitoring

S@N.IT! monitors the devices connected to the SAN, by periodically checking their status (working/non–working). Fault detection is notified to the S@N.IT! administrator by:

- modification of the icon that represents the device in the S@N.IT! GUI,
- internal logging.

In addition the fault detection may be notified by:

- a record in the event log mechanism in the S@N.IT! Server (see *S@N.IT! Architecture*, on page 1-4 and *Monitoring*, on page 2-3)
- an SNMP trap (see Chapter 8. S@N.IT! *SNMP Agent*)
- the start of a user–provided command.

Centralized Administration

S@N.IT! provides a centralized administration point for the SAN, by allowing the launching of the management application specific to each SAN device from S@N.IT! GUI. See *Contextual Menus*, on page 6-27.

S@N.IT! allows the S@N.IT! administrator to define views, i.e. logical subsets, of the whole SAN configuration. See *View Management*, on page 6-8.

LUN Access Control

S@N.IT! automatically discovers the LUNs of all storage devices connected to the SAN, with their type (RAID level) and capacity. It also displays a summarized view of LUN allocation and how they are used by the operating system of the servers.

As far as fabrics are concerned, the S@N.IT! GUI provides a clear display of the zoning configurations.

S@N.IT! also provides a centralized LUN mapping mechanism that dynamically configures which LUNs are allocated to a server and which LUNs are hidden from it. This LUN mapping mechanism is compatible with clustering solutions (HA–CMP on AIX, MSCS on Windows) and disk array failover software such as EMC PowerPath. It is independent of LUN masking features that can be provided by storage devices such as Access Logix for EMC Clariion, but it means that the S@N.IT! software must be installed on all servers connected to the SAN.

IMPORTANT: The LUN mapping configuration remains available even when S@N.IT! software is not running and, in some cases, even when it is de-installed. See *LUN Access Control*,on page 1-8.

File System Monitoring

S@N.IT! displays information about the file systems contained in each server and monitors their usage rate according to a user specified threshold.

Reporting

S@N.IT! lets you define report templates and generate reports in HTML and CSV format.

User Interfaces

S@N.IT! provides the S@N.IT! administrator with interfaces that allow SAN network information to be displayed about and to change its configuration:

- a Graphical User Interface (GUI) that can be launched as a standalone application or through a Web browser (see Chapter 6. *S@N.IT! GUI*),
- a Command Line Interface (CLI) that allows scripts to be built (see Chapter 7. *S@N.IT! Command Line Interface*),
- an SNMP agent that implements a Fibre Alliance MIB and a proprietary MIB (see Chapter 8. *S@N.IT! SNMP Agent*).

Access Protection

S@N.IT! protects the access to the SAN configuration, by managing two levels of user rights, protected by encrypted passwords:

- Common users are allowed only to display information.
- S@N.IT! administrator is allowed to change the S@N.IT! configuration (see *User Interface*, on page 2-9).

S@N.IT! Features Packaging

S@N.IT! features are packaged in two separate products:

- The base package: **S@N.IT! Limited Edition**
- The full package: **S@N.IT! Data Center**

For more information about each package, refer to: Chapter 2. *S@N.IT! Data Center Features* and Chapter 3. *S@N.IT! Limited Edition Features*.

S@N.IT! Architecture

Three Functional Parts

The S@N.IT! application is made of three functional parts:

- S@N.IT! Agents: they interface the fibre channel drivers to discover SAN components and manage the way the driver shows SAN LUN's to the operating system.
- S@N.IT! Server: it gathers and consolidates information received from S@N.IT! Agents and from SNMP agents of the SAN components. It also processes request from clients and implements the S@N.IT! SNMP agent.
- S@N.IT! GUI / CLI: they provide the end users respectively with a Graphical User Interface (GUI) and a set of commands (Command Line Interface, CLI) to display information and to act on the SAN configuration.

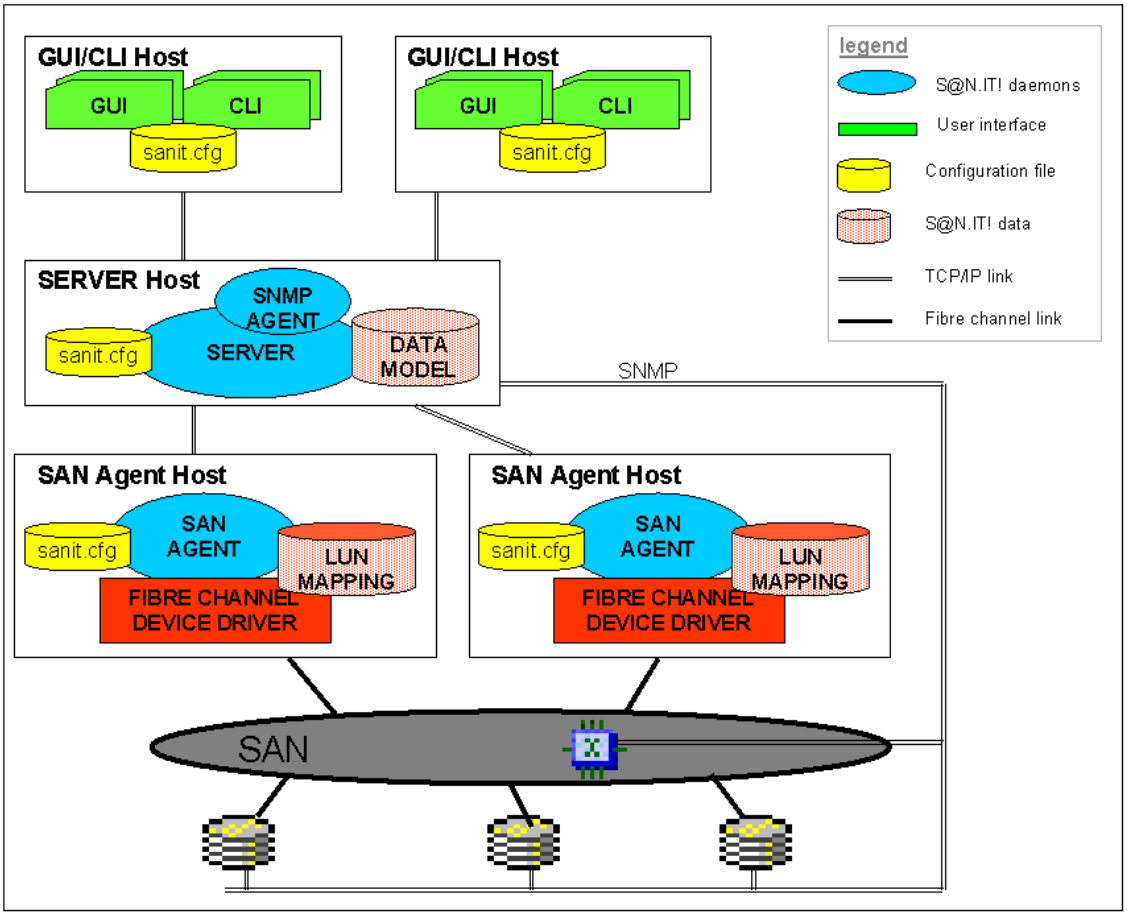

Figure 2. S@N.IT! architecture

The different parts of the S@N.IT! application communicate with each other using TCP/IP, and they can be distributed between several machines:

- The S@N.IT! GUI/CLI is available on each host where the S@N.IT! software is installed. The S@N.IT! GUI can also be launched remotely from a Web Browser that communicates with the S@N.IT! Server.
- There must be only one instance of S@N.IT! Server.
- There must be one instance of S@N.IT! Agent per host connected to the SAN.

But a single host can be both the S@N.IT! Server and a S@N.IT! Agent.

The S@N.IT! software relies on configuration parameters to identify the role (S@N.IT! Server, S@N.IT! Agent) of the host on which it is installed (see Chapter 5. *Configuration and Launching*) and to retrieve the S@N.IT! Server's address.

S@N.IT! administrate configurations that mix AIX, Linux, Solaris and Windows hosts.

The S@N.IT! Agent and the S@N.IT! Server run as daemons and are automatically launched at boot time:

- by the /**etc/rc.sanit** command on AIX, Linux and Solaris hosts,
- by the **S@N.ITScheduler** service on Windows hosts: as this service requires network access, it has to be launched under a specific user account, which is created when the software is installed (See *Windows installation*, on page 12-5).

It is necessary to restart the daemons when the configuration is updated (See Chapter 5. *Configuration and Launching*).

S@N.IT! Agents and S@N.IT! GUI/CLI instances declare themselves dynamically to the S@N.IT! Server, so it is not necessary to stop the whole application to add a new S@N.IT! Agent or a SAN component.

S@N.IT! GUI Main Window

The following figure shows the initial S@N.IT! GUI window and gives a general idea of this powerful interface. The S@N.IT! GUI is fully described in Chapter 6. *S@N.IT! GUI*.

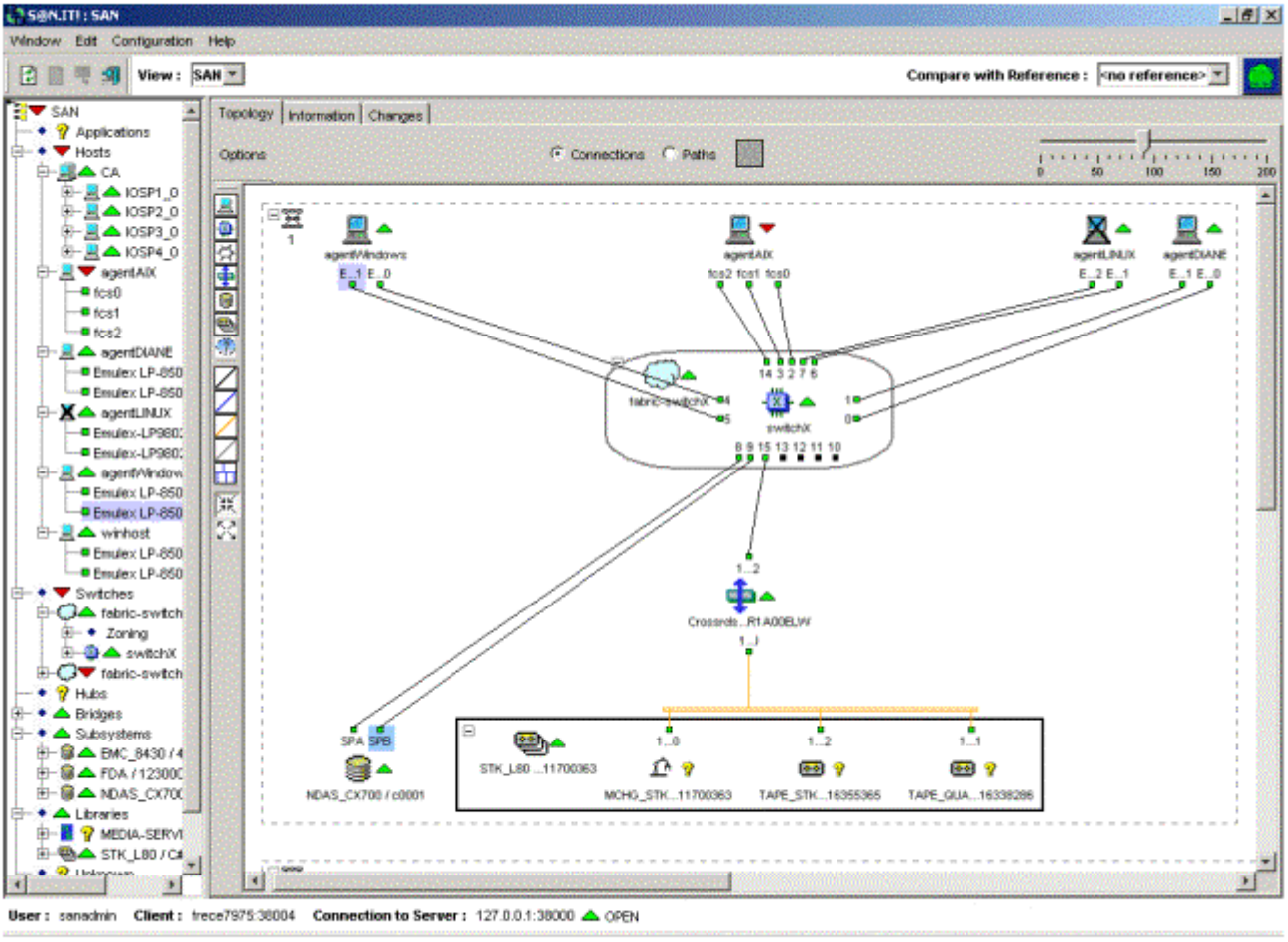

Figure 3. S@N.IT! GUI initial Window

Terminology

SAN Components

S@N.IT! manages three types of SAN components:

- Applications using the SAN to perform their tasks, such as backup applications.
- Physical SAN components connected to the SAN, which belong to the following families:
  - Hosts.
  - Disks arrays (called *subsystems* in S@N.IT! terminology)
  - Tape drives
  - Medium changers
  - Hubs
  - Switches
  - Fibre channel/SCSI bridges.
- Complex components:
  - Host Group: group of the nodes of a cluster, the partitions within a cabinet, the satellites of a GCOS8 platform
  - Library: group of robot and tape drives
  - Media Server: group of virtual libraries and virtual tape drives
  - Fabric: group of interconnected switches.

SAN Components Properties

Common Properties

The physical SAN components are described by a common set of properties:

- Family to which the component belongs (Host, Subsystem, TapeDrive, MediumChanger, Switch, Hub, Bridge, Fabric, HostGroup, Library, MediaServer, Application).
- Model name.
- Physical ID (usually the serial number).
- Logical name:
  - for a host, it is its hostname (see *S@N.IT! Agent configuration*, on page 5-7 for details),
  - for other components, this name can be set by the S@N.IT! administrator through the S@N.IT! GUI.
- Firmware level
- IP address:
  - for a host, it is the IP address used by its S@N.IT! Agent (see *S@N.IT! Agent configuration*, on page 5-7 for details),
  - for other components, it is the IP address of the SNMP agent that manages them.
- SNMP Port: the IP port on which the SNMP agent can be reached.
- Management application: the application(s) to be launched to check and/or modify the configuration and the internal state of the component.

- A list of ports (attachments points) to the SAN (Fibre Channel HBAs for hosts, directors for Symmetrix subsystems,...) represented by:
  - a name (device known by the operating system for an HBA),
  - a WWN (Worldwide Name),
  - a port–id: address of the port in the fibre channel network to which it is connected,
  - a location code,
  - the FW level and driver version for HBA.
- Monitoring Status: the operational status of the component (see *Monitoring*, on page 2-3).
- Status description: a comment giving details about the monitoring status .

Specific Properties

Some properties are specific to a component family, such as:

- for a host, its Operating System, and the version of the S@N.IT! software it runs,
- for a switch, the Domain–Id and WWN of the fabric to which the switch belongs,
- for a storage device, the list of LUNs attached to each of its ports,
- for complex components, the global monitoring status, which is a combination of the monitoring status of their members.

SAN Topology

Two levels of links between SAN ports are considered:

- Connections match the physical segments (cables) between two adjacent ports.
- Paths between hosts HBAs and storage device ports: a path is a logical connection between one host HBA and one storage device port that indicates if this host HBA “sees” that storage device port.

  So the existence of a path between one of its HBAs and a port of a storage device is a mandatory condition for a host to be able to access the data stored in that storage device.

LUN Access Control

To prevent a host from accessing a LUN in a SAN storage device, three levels of filtering are potentially usable:

- Zoning has to be performed within SAN infrastructure components (e.g. switches) and generally offers a limited granularity as it does not allow a per–LUN policy.
- Filtering within the storage device itself: this feature is called LUN masking, and is available as an option for high–end storage devices.
- Host–based LUN mapping interacts with the fibre channel driver to control which SAN LUNs it makes available (i.e. maps) to the operating system. This solution can be used whatever the SAN topology (no dependency versus the zoning features of a SAN infrastructure component) and whatever the storage device.

S@N.IT! LUN Access Control relies on host–based LUN mapping. It can be used within SAN configurations where zoning is implemented.

When used with storage devices that have an embedded LUN masking feature (EMC Access Logix, Volume Logix), pay attention to the fact that this feature limits the list of LUNs really reachable from a server. See *LUNs and LUN Groups Management*, on page 6-38 .

S@N.IT! also provides a way to configure embedded LUN masking for some models of Disk Arrays.

S@N.IT! displays the zoning configuration.

Chapter 2. S@N.IT! Data Center Features

This chapter describes how the main features of the S@N.IT! operate:

SAN Topology Discovery and Display

S@N.IT! automatically discovers the devices connected to the SAN, their properties and the links (connections and paths) between their ports.

This information is retrieved by:

- the S@N.IT! Agents that report information about the SAN devices they are connected to,
- the S@N.IT! Server, which discovers the SNMP agents of the SAN components and queries them for topology information.

The S@N.IT! administrator can supplement this information manually:

- by adding new SAN components, ports or connection,
- by editing the properties of the discovered components.

For more information see *Editing the SAN Components*, on page 6-10 and *S@N.IT! CLI Commands*, on page 7-3.

For this feature to be operational, the following requirements must be fulfilled:

1. A TCP/IP link must be available between the host on which the S@N.IT! Server runs and the SAN infrastructure components (switches, bridges and libraries), generally using a LAN connection as the switches have an Ethernet interface.
2. The SNMP agent of the devices (switches, librairies, disk arrays) must be configured properly and the host where the S@N.IT! Server runs must be allowed to access the related MIBs (refer to Chapter 11. *Supported SAN Components* for details).
3. The ranges of IP addresses to be polled for known MIBs must be configured on the S@N.IT! Server in the following parameters:

   OutbandDiscoveryStartAddress1
   OutbandDiscoveryStopAddress1
   OutbandDiscoveryPort1
   OutbandDiscoveryStartAddress2
   OutbandDiscoveryStopAddress2
   OutbandDiscoveryPort2
   OutbandDiscoveryStartAddress3
   OutbandDiscoveryStopAddress3
   OutbandDiscoveryPort3

These parameters provide three ranges of IP addresses and the SNMP port (default=161) for each range. See *Server Configuration*, on page 5-4 for more information.

Note that requirements 1 and 2 are also necessary for the switches and the libraries to be monitored.

The discovery information is refreshed periodically and the S@N.IT! GUI/CLI instances are notified when a modification of the SAN topology is detected, thus allowing the S@N.IT! GUI windows to be updated.

S@N.IT! SNMP traps are also sent when connections or SAN components are added or deleted. See S@N.IT! *SNMP Agent* on page 8-1.

The periodicity of the refresh is defined by the following parameters, defined in the S@N.IT! Server configuration:

- **DiscoveryRefreshPeriod**
- **DiscoveryScanningPeriod**

The S@N.IT! administrator can also request a refresh of the topology information either provided by a particular SAN component or globally, via the S@N.IT! GUI (see *Force full SAN discovery*, on page 6-28) or via S@N.IT! CLI (see *Command Line Interface*, on page 7-1).

S@N.IT! GUI displays the topology information and the SAN components properties (including the properties of their ports). The properties can be completed by the S@N.IT! Administrator (see *Contextual Menus / Properties*, on page 6-27). For example you can:

- Give a human understandable name to a disk array
- Set the IP address of a disk array or a library
- Give a label to a connection.

The S@N.IT! Server stores these modifications, but they may be overlapped by further discoveries.

If the SAN configuration contains components which are not supported by S@N.IT!, or hosts that have no SAN Agent software, they will be reported as UNKNOWN with an instance of UNKNOWN object per fibre channel port.

Fabric Discovery and Display

S@N.IT! displays the zoning configuration of the switches it has discovered. For this feature to be operational, some specific switch properties may need to be set, depending on the type of switch. See Chapter 11. *Supported SAN Components* for a list of switches that support this feature and configuration details.

The GUI displays zoning details in the **Zoning** tab associated with each fabric (see *Information / Zoning*, on page 6-36).

The topology frame also displays the ports involved in each zone (see *Zoning (S@N.IT! Data Center only*, on page 6-16).

Control of SAN Configuration Changes

S@N.IT! keeps a record of all the information it has discovered and is thus able to provide the S@N.IT! administrator with several means to understand what has changed in the network.

1. SNMP traps are sent when objects (SAN components, ports, connections, paths ...) are added or deleted. See S@N.IT! *SNMP Agent*, on page 8-1.

2. Objects (SAN components, ports, connections, paths ...) that are no longer discovered are still displayed by the GUI, but with a specific background that shows they have disappeared. They remain visible until the S@N.IT! administrator explicitly deletes them (see *Delete the objects no longer discovered*, on page 6-27 and *Deleting a SAN Component (Delete)*, on page 6-11.
3. The GUI provides a **Contents** tab for each group of objects. This tab indicates by which entity each object of the group has been discovered (see *Information / Contents*, on page 6-36).
4. The GUI provides a **Changes log** tab for each object. This tab describes all the modifications concerning this object (see *Information / Change Logs*, on page 6-36).
5. The S@N.IT! administrator can save a snapshot of the SAN configuration using the **Edit/Save current SAN as a reference** menu of the GUI. The administrator can then compare one of these saved references and the current state or compare two of these saved references. (See *Understanding SAN Changes*, on page 6-14 for details).

Changes are stored in a circular log file. The maximum size of this file is specified by the **DiscoLogSize** parameter of the S@N.IT! Server.

Monitoring

S@N.IT! monitors the supported SAN components it has discovered by periodically checking their status (working/non–working). (See Chapter 11. *Supported SAN Components* for details). For S@N.IT! Agents, the monitoring also includes a checking of LUN mappings against the SAN topology and the file systems usage rate. For SAN devices whose SNMP agent sends traps when their working status changes, S@N.IT! can act as an SNMP Manager and can use the received traps for monitoring purpose.

The following requirements must be fulfilled:

1. A TCP/IP link must be available between the host on which the S@N.IT! Server runs and the place where the SNMP Agent of each SAN component runs (either an administration station or the devices themselves).
2. The IP address of a SAN component is part of its S@N.IT! properties. It must be set to the address where the related SNMP agent resides. If not automatically discovered, this address must be set via the S@N.IT! GUI (see *Contextual Menus / Properties* on page 6-31).
3. The SNMP agent of each SAN component must be configured properly. The host on which the S@N.IT! Server runs must be allowed to access each related MIB and it must be configured as SNMP manager if monitoring via traps is to be used. Refer to Chapter 11. *Supported SAN Components* for details.

The periodicity of the refresh is defined by the **MonitoringPeriod** parameter of the S@N.IT! Server configuration. A refresh of the information can also be explicitely requested by the S@N.IT! administrator through the S@N.IT! GUI (see *Contextual Menus / Force Monitoring*, on page 6-29).

S@N.IT! manages the following monitoring status:

NORMAL	the SAN component is operational.
FAULTY	a fault has been detected. A host is reported as FAULTY when there is no more path to LUNs among those previously allocated to it, or one of its file systems is full.
NOT MONITORED	no monitoring is performed for the SAN component (model not supported), or IP address is not set.
UNKNOWN	the monitoring status is unknown because the monitoring method could not retrieve the operational status of the SAN Component.

The monitoring status can be UNKNOWN for the following components:

- a SAN component, that was previously discovered, is disconnected from the SAN or stopped, or its SNMP agent cannot be reached any longer.
- Hosts for which either the S@N.IT! Agent is stopped, re–starting or not installed, or the **ServerHost** parameter is not correctly set in the S@N.IT! Agent configuration.

The monitoring status and the status description are displayed as properties of each SAN component by the GUI and can be edited by the S@N.IT! administrator. For complex components, groups of components and views, the status is a combination of the status of the SAN components that they contain. These properties and the icons that represent the components are updated each time a change is detected.

A log of the monitoring status changes is available in the S@N.IT! GUI (**Monitor Log** tab). This information is stored in a circular file on the S@N.IT! Server: the file is common to all SAN components. Its size is defined by the **MonitoringLogSize** parameter of the S@N.IT! Server configuration.

S@N.IT! SNMP traps are also generated when a status change is reported (see *S@N.IT! SNMP agent* on page 8-1).

Faults detections can be registered in the error logging framework of the S@N.IT! Server. This feature is set using the **EnableSystemErrorLog** parameter of the S@N.IT! Server configuration.

S@N.IT! provides a mechanism for a user supplied command to be automatically launched on the S@N.IT! Server whenever a monitoring status change is detected. This command has to be provided by setting the **UserNotificationCommand** parameter of the S@N.IT! Server configuration. The expected interface of this command is described in *Server Configuration*, on page 5-4. Note that the command will be launched under the same user account as the S@N.IT! services (**root** under AIX, Linux and Solaris; **administrator** under Windows).

The following configuration parameters of the S@N.IT! Server are involved in the monitoring:

- **MonitoringPeriod**: defines the periodicity of monitoring refresh.
- **ActivateTrapListener**: indicates whether the S@N.IT! Server shall listen for traps from SAN components.
- **TrapListenerPort**: specifies the UDP port where the S@N.IT! Server shall listen for traps from SAN components.

Note: The default value (162) can be changed to prevent conflicts with another SNMP manager running on the host where the S@N.IT! Server runs. Pay attention that it is not always possible to configure the SAN components to send their traps to another port than this default value.

- **EnableSystemErrorLog**: indicates whether the S@N.IT! Server shall register fault detections in the error-logging mechanism of the operating system where it runs.
- **UserNotificationCommand**: path of the command to be launched on the S@N.IT! Server each time a monitoring status change is detected.
- **MonitoringLogSize**: maximum size of the circular file where the monitoring status changes are logged.

Centralized SAN Administration

S@N.IT! provides a centralized administration point for the SAN, by allowing to launch the management application specific to each SAN devices from S@N.IT! GUI.

S@N.IT! knows the management tools associated with each supported SAN component (for example Navisphere for EMC DAS, SymmConsole for EMC Symmetrix, Web Browser or telnet for Brocade switches,...).

A default value is set as a property of the SAN component when it is first discovered by S@N.IT!. This property can be modified by the S@N.IT! administrator (using the *Edit Properties* menu of the S@N.IT! GUI) to another value proposed by S@N.IT!: for example for a Brocade switch, the properties can be changed from "`WebBrowser`" to "`Telnet`".

S@N.IT! is also aware of the way to launch the selected tool (path of the command, launched on the platform where the S@N.IT! GUI has been started or on a S@N.IT! Agent,...). This information can be customized when the tool is launched, using the *Start Management Tool* menu of the S@N.IT! GUI (see *Start Management Tool*,on page 6-27).

A management tool can be launched only if its interface is compatible with the host where the S@N.IT! GUI is running:

- Windows application: it must reside on the Windows host on which the S@N.IT! GUI is started.
- Web application: a Web browser must be installed on the host on which the S@N.IT! GUI is started; a default path for the Web browser is provided within the S@N.IT! configuration (**WebBrowser** parameter) and must be modified if it does not match the current installation (see *GUI/CLI configuration*,on page 5-10).
- telnet application: a telnet client must be available on the host on which the S@N.IT! GUI has been launched.
- X–Windows application: a X11 server must be available on the user's display.

LUN Access Control

S@N.IT! provides a centralized LUN Access Control mechanism using host–based LUN mapping: it interacts with the fibre channel drivers on the S@N.IT! Agents to control which SAN LUNs they make available (i.e. map) to the operating system.

Note: The LUN Access Control feature is available in both S@N.IT! Data Center and S@N.IT! Limited Edition packagings. See Chapter 12. *Supported Platforms* for operating systems restrictions.

A S@N.IT! Agent must be installed on each host to be connected to the SAN.

S@N.IT! displays all the LUNs available behind each subsystem port and their properties:

- **LUN identifier**, as seen by the operating system (16 hexadecimal digits representing the SCSI–3 address of the LUN, shortened to the 4 first digits when displayed by the GUI).
- **Device LUN**, unique identifier of the LUN within the storage device. Its format depends on the storage device.

Note: For some storage devices, both values are identical. For others – generally the disk arrays that have an embedded LUN masking capability – they are not identical as this allows the same LUN identifier (for example LUN 0) to be presented for different storage spaces (and generally to different servers).

- **type** of LUN managed by the Disk Array: individual disk, RAID 1, RAID 5, ...
- **capacity**
- **reachability**.

S@N.IT! enables the S@N.IT! administrator to configure which host may access each LUN, and the path to be used (host Fibre Channel adapter – Subsystem port) and when a LUN is mapped to a host, it also displays information on the way the LUN is seen by the operating system (hdisk and volume group on AIX hosts, NT LUN and drive letter on Windows hosts).

The mapping configuration can be modified at any time, i.e. a LUN can be unmapped from a host or a new LUN mapped.

The mapping configuration of a host remains active even when the S@N.IT! services are not started on that host or the S@N.IT! Server is not reachable.

There are strong dependencies between the LUN mapping/unmapping operations and:

- the type of operating system (allocation/de–allocation of storage resources)
- the type of fibre channel HBA and associated driver (configuration).

Refer to Chapter 12. *Supported Platforms* for details on the procedures to be used for each supported operating system and fibre channel driver.

Note: it is highly recommended not to map a LUN on two heterogeneous hosts (AIX and Windows).

S@N.IT! registers a log of all LUN Access Control actions. The S@N.IT! GUI allows to read, reset and copy this log (*LUN Access Log* tab).

LUN Access Control Mechanism

The LUN access control mechanism may have the following states for a given S@N.IT! Agent.

Main states

- **ACTIVE**: access to the SAN LUNs is restricted to the LUNs that:
  - either have been explicitly mapped via S@N.IT!
  - or belong to subsystems (disk arrays) that are not supported by S@N.IT!
  - or are not disks (library robot or tape drives).
- **INACTIVE**: access to the SAN LUNs is not restricted by S@N.IT!
- **NOT_AVAILABLE**: the feature is not available because it is not supported for the operating system, or because there is no fibre channel adapter in the machine.

Intermediate states

Their occurrence may vary according to the operating system and fibre channel driver features.

- **ACTIVABLE**: access to the SAN LUNs is not modified by S@N.IT! but it will be restricted after the next reboot.
- **INCONSISTENT**: the LUN access control mechanism does not have the same status for all fibre channel adapters in the SAN agent generally because another tool (HBA dependent) has been used.
- **ENABLED**: the LUN access control mechanism is started but there is no connected device.
- **DISABLED**: the LUN access control mechanism is supported but managed outside of S@N.IT! (using another HBA–dependant tool).

The initial state after S@N.IT! software installation is ACTIVABLE, except in the case of migration or update where the previous state is unchanged, or if the operating system does not support this feature. It turns from ACTIVABLE to ACTIVE after a reboot.

The LUN Access Control mechanism can be activated / deactivated using the S@N.IT! GUI, the S@N.IT! CLI (**Activate** / **Deactivate** commands) or the S@N.IT! Agent local commands (**san_activate**, **san_deactivate**).

The operations to be performed after de-activation of the LUN access control mechanism depend on the operating system and the fibre channel. See *Hosts and Adapters*, on page 11-1 for details.

Important: for security reasons, a host shall be connected to the SAN only after a LUN Access Control mechanism has been set and configured.

Mapping LUNs

LUN mapping is the operation that allows a host operating system to access a LUN on a storage device. This operation can be performed by the S@N.IT! administrator via the S@N.IT! GUI, the S@N.IT! CLI **AllowAccess** command or the S@N.IT! Agent local commands (**san_map_lun**).

LUN mapping can be performed on several LUNs of the same subsystem port, at the same time, for the same host operating system.

LUNs can be mapped on several hosts, as this may be required in cluster configuration (HA–CMP on AIX or MSCS on Windows). But since S@N.IT! is not aware of the clustering configuration, this operation is not automatic and the S@N.IT! administrator must manually perform the multi–mapping of a LUN.

The path between the host and the subsystem must be selected (a couple Host Fibre Channel adapter – subsystem port).

The state of the system and the operations to perform after mapping LUNs are described in *Hosts and Adapters*, on page 11-1.

S@N.IT! provides a means to automatically launch user provided commands on a S@N.IT! Agent host, before and after each LUN mapping operation.

The paths for these commands are to be described in the configuration of each S@N.IT! Agent: this is the aim of the **UserPreMappingCommand** and **UserPostMappingCommand** parameters (see *S@N.IT! Agent Configuration*, on page 5-7).

The exit codes are managed as follows:

- if the **UserPreMappingCommand** fails, the mapping operation is not performed,
- if the mapping operation fails, the **UserPostMappingCommand** is not performed.

The consistency between the LUN mappings and the SAN topology is checked periodically as part of the monitoring of the S@N.IT! Agents (see Monitoring, on page 2-3).

See Chapter 9. *Best Practices* where is described the impact of the SAN evolutions on LUN Access Control.

Multi–paths management

If there are several paths between a S@N.IT! Agent and a subsystem, there is generally a multipathing software running on the agent, which requires that LUNs are mapped for each path (ie couple HBA – subsystem port). The S@N.IT! Administrator must repeat the LUN mapping operations for each path. S@N.IT! provides help to do this:

- The **AllowAccess** S@N.IT! CLI command and the **san_map_lun** command enable to specify several paths.
- The S@N.IT! GUI offers to map all the paths that it has detected between the S@N.IT! Agent host and the subsystem.

Unmapping LUNs

LUN unmapping is the operation that removes a host operating system's access to a LUN on a storage device. This operation can be performed by the S@N.IT! administrator via the S@N.IT! GUI, the S@N.IT! CLI **DenyAccess** command or the **san_unmap_lun** command.

LUN unmapping can be performed for several LUNs, several ports and several S@N.IT! agents at the same time.

The LUNs to be un–mapped must not be used by the operating system, so before unmapping a LUN the disk access activity to the LUN must be stopped. See *Hosts and Adapters*, on page 11-1 for more details.

S@N.IT! provides a means to automatically launch user provided commands on a S@N.IT! Agent host, before and after each LUN unmapping operation.

The paths for these commands must be described in the configuration of each S@N.IT! Agent using the **UserPreUnmappingCommand** and **UserPostUnmappingCommand** parameters (see *S@N.IT! Agent Configuration*, on page 5-7).

The exit codes are managed as follows:

- if the **UserPreUnmappingCommand** fails, the unmapping operation is not performed,
- if the unmapping operation fails, the **UserPostUnmappingCommand** is not performed.

See Chapter 9. *Best Practices* for a description of the impact of SAN evolutions on LUN Access Control.

Multi–paths management

If a LUN was mapped through several paths, it must be explicitly unmapped for each path, except for DAS and NDAS disk arrays for which S@N.IT! GUI automatically unmaps all paths at once.

The **DenyAccess** CLI command and the **san_unmap_lun** command enable to specify several paths.

Storage Arrays LUN Masking Configuration

S@N.IT! provides an interface to update the LUN masking configuration of some storage arrays. To implement this feature, note what follows:

- The storage array must be installed and configured to provide LUN masking.
- The access rights are attached to groups of LUNS. Thus the LUN masking configuration consists in:
  - Creating/deleting named groups of LUNs
  - Adding or removing LUNs into or from a group of LUNs
  - Allowing or forbidding a host base adapter to access a group of LUNs
- Depending on the array model, some specific properties may need to be set manually using the S@N.IT! GUI or the CLI (see Chapter 11. *Supported SAN Components* for details).
- As in the case of LUN access control, these operations have an impact on the SAN agent hosts that are connected to the array, because their operating systems need to discover or release storage resources.

The configuration of storage arrays LUN masking is available via the Command Line Interface CLI. The following commands are dedicated to this feature: **AddLunGroupContent**, **AddLunGroupPath**, **CreateLunGroup**, **LsDeviceSubsystem**, **LsLunGroup**, **LsLunGroupContent**, **LsLunGroupPath**, **LsPortLuns**, **RemLunGroup**, **RemLunGroupContent**, **RemLunGroupPath** and **SetAclMode**.

See *Command Line Interface*, on page 7-1 for more details.

File Systems Display and Monitoring

S@N.IT! provides information about the file systems in each SAN agent host:

- Name and mount point
- Type
- Location within the SAN storage arrays
- Allocated and used size.

This information is refreshed periodically and displayed within the GUI. (see *Information / File Systems* , on page 6-35).

S@N.IT! checks the usage rate of each file system as part of the host monitoring task. By default, the threshold is specified by a parameter configured in the S@N.IT! Server (see *Server Configuration*, on page 5-4); it can also be customized for each file system (see *Information / File Systems* , on page 6-35).

Reports Generation

S@N.IT! GUI and CLI allow to generate reports among a set of predefined reports. All reports files are available in HTML and CSV format. Reporting is customisable through templates (see *Configuration/Edit Report Templates* , on page 6-22).

A report is a set of tables describing information about a SAN component (or a group of SAN components):

- Properties of the component, ports, logical paths, connections
- Allocation of LUNs for hosts
- Allocation of LUNs for storage devices
- LUN access control log
- Monitoring log
- File systems for hosts
- Zoning
- Changes log.

The customisation enable to select tables and, for each table, to select and to sort columns.

User Interface

S@N.IT! provides four user interfaces:

- a Graphical User Interface (GUI) (see Chapter 6. S@N.IT! Graphical User Interface (GUI)),
- a Command Line Interface (CLI) available on any host on which the S@N.IT! software is installed (see Chapter 7. S@N.IT! Command Line Interface (CLI)),
- a set of user commands to run directly on a S@N.IT! Agent (see Chapter 10. Troubleshooting),
- an SNMP MIB that can be read from an SNMP manager (see Chapter 8. S@N.IT! SNMP Agent).

S@N.IT! GUI/CLI – Access Rights

The S@N.IT! GUI can be launched:

- as a standalone application on any host where the S@N.IT! software is installed,
- through a java enabled Web browser accessing a URL on the host on which the S@N.IT! Server runs (see *Enabling the applet mode for the GUI*, on page 5-6). But in that case:
  - it is not allowed to modify the S@N.IT! configuration via the S@N.IT! GUI,
  - it is not possible to save files (monitoring logs or LUN access logs).

Both S@N.IT! CLI and GUI communicate with the S@N.IT! Server to display information and action on the SAN configuration.

S@N.IT! manages two levels of access rights, via two predefined users protected by passwords:

- **common** user is only allowed to display information,
- **sanadmin** user is also allowed to modify the S@N.IT! configuration (management of LUN Access Control, edition of SAN components,...)

Each S@N.IT! CLI and GUI session is run under a S@N.IT! user's account:

- The S@N.IT! GUI is always initially launched under the **common** user account. The selection of **sanadmin** user is made through the *Configuration/Set user* menu (the password of the new user is requested).
- For the S@N.IT! CLI, the selection of a S@N.IT! user is made when the S@N.IT! CLI session is started (command `sanit -c Open`. See *To start a command line interface session*, on page 7-1).

Several instances of S@N.IT! GUI or CLI can run at the same time for the **common** user account. But to ensure SAN configuration consistency, only one S@N.IT! GUI or CLI instance can run under the **sanadmin** user account for one S@N.IT! Server.

S@N.IT! checks and stores the password for each user. Their initial values are:

- **common** user: no password,
- **sanadmin** user: **admin**.

Passwords for both users can be modified as follows: launch the S@N.IT! GUI under **sanadmin** account and use the *Configuration / Set password* menu. Have in mind the following rules:

- For security reasons, the **sanadmin** password must be modified after installation of the S@N.IT! Server.
- If the password of the **common** user is modified while instances of S@N.IT! GUI are running under **common** account, the S@N.IT! GUI will ask for the new password.
- If the password of the **common** user is modified while sessions of S@N.IT! CLI are running under **common** account, these sessions will exit.

Warning

When launched as a standalone application, the S@N.IT! GUI allows to modify or create files on the host where it has been launched (configuration file, copies of monitoring or LUN Access Control logs). These files are local to the current platform and their access rights are managed by a local policy. As the S@N.IT! GUI will inherit the privileges of the user (from the local operating system standpoint) that has launched it, it may be unable to modify or create files, if this user is not allowed to do so.

S@N.IT! Agent Local Commands

S@N.IT! Agent local commands do not interact with the S@N.IT! Server, so they can be launched even when the S@N.IT! Server is not reachable or when the S@N.IT! application is not launched on the current machine.

They only act locally and must be launched under **root** account on AIX, Solaris and Linux hosts, or **administrator** account on Windows hosts.

These commands allow to:

- Gather information for maintenance purpose (See Chapter 10. *Troubleshooting*).
- Map / Unmap LUNs on the local S@N.IT! Agent host (see *san_map_lun*, on page 10-4 and *san_map_lun*, on page 10-5). Note that as these commands do not interact with the S@N.IT! Server even if it is running, the server will not be able to notify running S@N.IT! GUI or CLI instances of LUN access configuration changes.
- Activate / Deactivate the LUN access control mechanism on the local host.

S@N.IT! SNMP Agent

The S@N.IT! SNMP agent is launched as part of the S@N.IT! Server. It manages two MIBs:

- the Fibre Alliance MIB, which provides information about the SAN topology
- the proprietary S@N.IT! MIB, which provides information about the storage resources and their allocation to S@N.IT! Agents (LUN mappings).

It also generates SNMP traps when monitor status or SAN topology changes are detected.

This feature can be turned on/off by setting the **RunSnmpAgentOnStartup** parameter of the S@N.IT! Server configuration. The **SnmpAgentPortNumber** parameter defines the UDP port where the S@N.IT! SNMP agent is waiting for SNMP requests. A default SNMP manager can be registered for receiving S@N.IT! SNMP traps using the following parameters:

- **SnmpManagerIP**
- **SnmpManagerPort**
- **SnmpManagerFilter**

Other SNMP managers can register themselves for receiving traps using SNMP interface (see Chapter 8. *S@N.IT! SNMP Agent* for details).

Note: see *Server Configuration*, on page 5-4 for the description of the S@N.IT! Server configuration parameters.

Chapter 3. S@N.IT! Limited Edition Features

S@N.IT! Limited Edition is a subset of S@N.IT! and provides features that are focused on LUN Access Control.

This chapter describes the S@N.IT! Limited Edition features:

SAN Topology Discovery and Display

S@N.IT! Limited Edition discovers the hosts and storage devices connected to the SAN, and the paths between the hosts fiber channel adapters and the storage devices ports. The discovery information is refreshed periodically and the S@N.IT! GUI/CLI instances are notified when a modification of the topology is detected.

The periodicity of the refresh is defined by the **InbandDiscoveryPeriod** configuration parameter on each S@N.IT! Agent. See *S@N.IT! Agent Configuration*, on page 5-7 for more information.

The topology information and the properties of the SAN components are displayed by the S@N.IT! GUI and can be edited by the S@N.IT! administrator (for example to give a user–understandable name to a storage device).

S@N.IT! keeps a record of all SAN components it has discovered : a component (host or storage device) remains visible in the S@N.IT! GUI even if it has been disconnected, until it is explicitly deleted by the S@N.IT! administrator.

LUN Access Control

S@N.IT! Limited Edition fully manages LUN Access Control for supported storage devices (see the description of *LUN Access Control* for S@N.IT! Data Center, on page 2-5).

Monitoring

S@N.IT! Limited Edition monitors the S@N.IT! Agents, by periodically checking:

1. the connection between the S@N.IT! Server and the S@N.IT! Agents,
2. the paths between the S@N.IT! Agents and the storage devices where each S@N.IT! Agent has allocated (mapped) LUNs.

The refresh periodicity of S@N.IT! Agents status is defined by the **MonitoringPeriod** parameter of the S@N.IT! Server configuration.

S@N.IT! Limited Edition manages the following monitoring status for the hosts:

NORMAL	the S@N.IT! Agent is operational, no LUN mapping path missing.
FAULTY	there is no more discovered path to some LUNs that have been previously mapped to the S@N.IT! Agent.
NOT MONITORED	storage devices are always reported as “NOT MONITORED”.
UNKNOWN	the S@N.IT! Agent is not registered by the S@N.IT! Server.

S@N.IT! GUI displays the monitoring status as a property of a S@N.IT! Agent, which can be edited by the S@N.IT! administrator. The value of this property and the icon that represents the S@N.IT! Agent are updated each time a change is detected.

A log of the monitoring status changes is available in the S@N.IT! GUI (*Monitor Log* tab). This information is stored in a circular file on the S@N.IT! Server. This file is common to all S@N.IT! Agents and its size is defined by the **MonitoringLogSize** parameter of the S@N.IT! Server configuration.

Faults detections can be registered in the error–logging framework of the S@N.IT! Server. This feature is set using the **EnableSystemErrorLog** parameter of the S@N.IT! Server configuration.

S@N.IT! provides a mechanism that automatically launches a user supplied command on the S@N.IT! Server whenever a monitoring status change is detected: this command has to be set in the **UserNotificationCommand** parameter of the S@N.IT! Server configuration. See *Server Configuration*, on page 5-4 . Note that this command will be launched under the same user account (then with the same privileges) than the S@N.IT! services.

Note: see *Server Configuration*, on page 5-4 for the description of the S@N.IT! Server configuration parameters.

User Interface

S@N.IT! Limited Edition provides three user's interface:

- A Graphical User Interface (GUI) that can be launched either:
  - as a standalone application on any host where S@N.IT! software is installed,
  - or through a java enabled Web browser accessing a URL on the host where the S@N.IT! Server runs (see *Enabling the Applet Mode for the GUI*, on page 5-6).
- A Command Line Interface (CLI) that can be launched on any host where S@N.IT! software is installed (see Chapter 7. S@N.IT! Command Line Interface (CLI)).
- A set of commands that act locally on a host.

S@N.IT! GUI/CLI

- Only a subset of the S@N.IT! GUI features are available in S@N.IT! Limited Edition:
  - Editing the SAN components:
    - Modifying a SAN component (Properties)
  - S@N.IT! GUI menus:
    - Window menu
    - Configuration menu
    - Edit menu / Delete
    - Contextual Menus:
      - Activate LUN Access Control (S@N.IT! Agent selected)
      - De–activate LUN Access Control (S@N.IT! Agent selected)
      - Delete (SAN component selected)
      - Properties (SAN component selected)
  - S@N.IT! GUI Information Frame:
    - Topology restricted to paths
    - Information/Properties
    - Information/Local Topology (SAN component selected)
    - Information/Ports (SAN component selected)
    - Information/Monitor Log (S@N.IT! Agent selected)
    - Information/Subsystem LUNs (Storage device selected)
    - Information/Host LUNs (S@N.IT! Agent selected)
    - Information/LUN access logs (S@N.IT! Agent selected)
  - LUN and LUN groups management.
- S@N.IT! Limited Edition provides the same level of access rights than S@N.IT! Data Center. Refer to *S@N.IT! GUI/CLI – Access Rights*, on page 2-10 for a complete description.

S@N.IT! Agent Local Commands

S@N.IT! Limited Edition provides the same S@N.IT! Agent local commands than S@N.IT! Data Center. Refer to *S@N.IT! Agent Local Commands*, on page 2-11 for a complete description.

Chapter 4. Installation Overview

This chapter is an overview of the installation and configuration tasks.

The S@N.IT! features are packaged in two separate products and CDROMs:

- *S@N.IT! Limited Edition*
- *S@N.IT! Data Center*

Each of these packages contains software for all supported operating systems. As far as software installation is concerned:

- *S@N.IT! Limited Edition* is a prerequisite of *S@N.IT! Data Center.*
- The installation of *S@N.IT! Data Center* is required and meaningful only on the host that will act as the S@N.IT! Server.
- S@N.IT! Limited Edition software must be installed on all hosts, whatever their roles in the application (S@N.IT! Agents, S@N.IT! Server, or S@N.IT! GUI/CLI).

Note: The S@N.IT! GUI can also be accessed through a Web Browser. In that case there is no need to install software on the host where the Web Browser is launched (except if this host is also connected to the SAN).

Although the general configuration of the application can be modified at any time, it is recommended to follow the steps described below:

1. Configure the TCP/IP network.
2. Determine which host will be the S@N.IT! Server.
3. Install and configure S@N.IT! Server (including installation of *S@N.IT! Data Center* software).
4. Install and configure S@N.IT! Agent hosts BEFORE to connect them to the SAN (to prevent from unwanted access to SAN LUNs).
5. Reboot S@N.IT! Agent hosts.
6. Connect S@N.IT! Agent hosts to the SAN.
7. Launch the S@N.IT! GUI as **sanadmin** user and complete the configuration (device properties, LUN mappings).

Warning

On operating systems supporting S@N.IT! LUN access control, the LUN access control mechanism becomes active after the first reboot following the installation. This means that the host cannot access any SAN LUN until an explicit mapping is performed using the S@N.IT! GUI. **For host booting on the SAN, check that the boot LUN is mapped before rebooting the system.**

During a first installation, the **sanit.cfg.template** is installed and copied into the **sanit.cfg** configuration file.

Refer to Chapter 12. *Supported Platforms* for details about the installation procedure for a specific operating system.

For the initial installation, the software must be installed on each host explicitly. For further upgrades of S@N.IT! Agents running version 6 (or higher) of the software, it will be possible to push software updates from the S@N.IT! Server. For this feature to be operational on a S@N.IT! Agent, the **AllowRemoteUpdate** parameter of the agent must be set to **true**. See *S@N.IT! update*, on page 6-30 for details.

Chapter 5. Configuration and Launching

This chapter describes the following configuration tasks:

Configuration Overview

To configure the whole S@N.IT! application, perform these steps:

1. Determine which host will run the S@N.IT! Server.
2. Determine the IP addresses or names to be used for each host (see *Relationship between S@N.IT! Configuration and TCP/IP Configuration*, on page 5-2).
3. Determine the IP address ranges of the devices to be discovered by the S@N.IT! Server (switches, librairies ...) and make sure these devices have their SNMP agent properly configured.
4. Configure the S@N.IT! Server.
5. Restart the S@N.IT! services on the S@N.IT! Server.
6. Launch the S@N.IT! GUI.

Note: For Windows hosts, the essential part of the configuration – which determines the role (S@N.IT! Agent – S@N.IT! Server) and the IP naming of the host – is done during software installation. Thus, steps 7 and 9 are not necessary for Windows hosts.

7. For each host connected to the SAN:
   - Configure the host as a S@N.IT! Agent.
   - Restart the S@N.IT! services on the local host.
   - Check that the host appears in the S@N.IT! GUI.
8. Edit and complete the properties of the devices (contextual menu/properties) and especially check that the IP address of each device is set, otherwise the monitoring of the device may not be operational.
9. For each other host where the S@N.IT! GUI or CLI is to be launched:
   - Configure only the Client part.
   - Check that the connection to the server is successful.

To configure S@N.IT! on a host, perform these steps:

1. Log in as **root** (AIX, Linux or Solaris) or **administrator** (Windows) on this host.
2. Launch S@N.IT! GUI.
3. Select the *Configuration/EditConfiguration* menu.
4. Edit the fields that need to be modified.
5. Save the configuration.
6. If the host is the S@N.IT! Server or a S@N.IT! Agent, restart the S@N.IT! services.

The configuration information is stored in the **sanit.cfg** file located under:

- /**etc**/**sanit**/ on AIX, Linux or Solaris hosts.
- **<%installation directory>** on Windows hosts.

To restart the S@N.IT! services, use:

- the /**etc**/**rc.sanit** command on AIX, Linux or Solaris hosts,
- the Stop/Start **S@N.ITScheduler** Windows service on Windows hosts.

Relationship Between S@N.IT! Configuration and TCP/IP Configuration

The different hosts involved in the S@N.IT! application communicate with each other using TCP/IP connections:

- between the S@N.IT! Server and the S@N.IT! Agents,
- between the S@N.IT! GUI or CLI instances and the S@N.IT! Server.

For these connections to be established, ensure that the following requirements are met:

- Both ends of the connection are linked by an IP network (either because they belong to the same IP subnetwork, or because IP routes are defined between them),
- Each end of the connection knows the IP address of the other one.
- If a firewall is present, check that the TCP ports used for the communications are open. These ports are described by the following configuration parameters:
  - **ServerPort** and **ServerRMIPort** on the S@N.IT! Server for communications from the Agents and the Clients to the Server
  - **ClientRMIPort** on the Clients for communications from the Server to the GUI/CLI instances. Note that if several GUI or CLI instances are launched concurrently on the client machine, they will use several ports that are allocated in sequence starting from ClientRMIPort value
  - **AgentRMIPort** on the Agents for communications from the Server to the Agents.

The IP adresses used are configured using the **ServerHost** and **HostName** parameters in the **sanit.cfg** configuration file on each host.

ServerHost IP name of the S@N.IT! Server; this parameter is used locally by S@N.IT! Agents and S@N.IT! GUI or CLI instances for their first connection to the S@N.IT! Server. It may be provided as:

– a dotted IP address (e.g. 123.123.123.123),

– a string (e.g. server): in this case the name resolution (mapping of this string into an IP address) is done locally.

HostName IP name to be used by other machines to connect to the current one – this parameter is transmitted as is to other machines – it may be provided as:

– a dotted IP address (e.g. 123.123.123.123),

– a string (for example `sanhost`): in this case the name resolution is performed on remote machines.

If this field is left blank, the default is the hostname of the current machine.

- For S@N.IT! GUI or CLI clients and S@N.IT! Agents the **HostName** parameter is transmitted to the S@N.IT! Server that will use it to establish callback connections. This means that **HostName** must be a name that will be resolved by the S@N.IT! Server into an IP address accessible from the S@N.IT! Server or left blank if the hostname fulfils this condition.

- For the S@N.IT! Server, the **HostName** parameter is transmitted to all S@N.IT! GUI or CLI clients and S@N.IT! Agents. This means that **HostName** must be a name that will be resolved on each of those machines into an IP address that allows them to access the S@N.IT! Server.

  If the S@N.IT! Server has several IP interfaces (and addresses), this may raise a problem as there is only one possible value of **HostName** that must be solved into different IP addresses, for each S@N.IT! GUI or CLI instance and S@N.IT! Agent, depending on the network they use to connect to the S@N.IT! Server.

  Warning: The **HostName** or **ServerHost** values must correspond to IP addresses that will be reachable during the whole system life as S@N.IT! services are started at boot time and should normally not be stopped once they have been configured.

  The IP address to be used to resolve the **HostName** value configured on the S@N.IT! Server must be specified on each machine running S@N.IT! software (S@N.IT! Agents, S@N.IT! GUI/CLI): if no DNS is used or if the main IP address provided by the DNS is not the one to be used, a private S@N.IT! Server **HostName** must be used and declared in the **hosts** file (for Windows) or **/etc/hosts** file (for AIX, Linux or Solaris) of each machine involved in the application, as shown in figure 4 below.

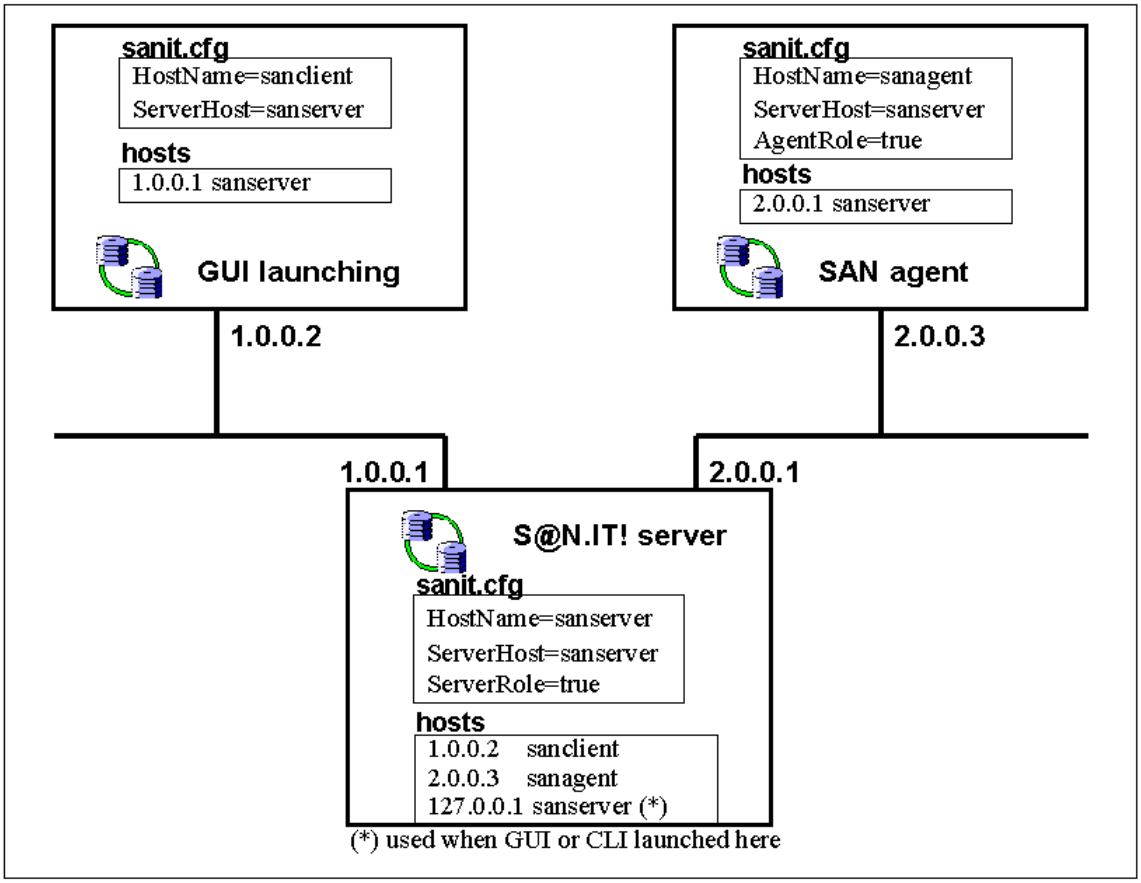

Figure 4. TCP/IP configuration with multiple networks

Note: In any case, to control IP address resolution, only one IP address must correspond to a given **HostName** in DNS (Main IP address).

On Windows 2000 do not let the other adapter(s) register their (secondary) addresses in the DNS. Use a **hosts** file on each host that will communicate with the S@N.IT! Server through another subnet than the one corresponding to the S@N.IT! Server's main IP address.

Check that pinging the S@N.IT! Server's **HostName** from each S@N.IT! Agent and S@N.IT! GUI/CLI, uses the proper IP address.

Server Configuration

Configuration Parameters for S@N.IT! Limited Edition

S@N.IT! Limited Edition uses the following parameters:

HostName IP name to be used by other machines to connect to the current one; this parameter may be provided as:

– a dotted IP address (for example `123.123.123.123`),

– a string (for example `sanhost`): in this case the name resolution is performed on remote machines.

If this field is left blank, the default is the hostname of the current machine.

ServerRole must be set to true.

UserNotificationCommand (optional) full path of the command to be called each time a monitoring status change is detected.

The command will be launched by the S@N.IT! Server with the following syntax:

```
<command path> model physical-identifier state "description"
```

`command_path`	value of the **UserNotificationCommand** field in **sanit.cfg.**
`model`	model of the SAN component whose monitoring status has changed.
`physical_identifier`	identifier of the SAN component whose monitoring status has changed.
`state`	new monitoring status (NORMAL, FAULTY, UNKNOWN, NOT_MONITORED).
`description`	free text describing the state change.

Example on AIX

```
/bin/ksh /etc/sanit/NotifyCommand.template "DAS_5700" "3002295"
"FAULTY" "Fault LED set"
```

The command is launched as a daemon, with the privileges of the user that has started the S@N.IT! services on the current host (usually **root** on AIX, Linux and Solaris and S@N.IT! **service account** on Windows). The command may not be interactive. The exit code of the command is not taken into account.

For details on the syntax of the <command path>, see Chapter 12. *Supported Platforms*.

EnableSystemErrorLog if set to true, a message will be recorded in the error login system of the current host each time a SAN component is detected as faulty; if set to false, faulty components are not recorded.

The following parameters will generally not have to be modified:

ServerPort TCP port where the S@N.IT! Server waits for connections from S@N.IT! Agents or S@N.IT! GUI/CLI instances. The default value (38000) should not be modified, else the S@N.IT! Agents or S@N.IT! GUI/CLI instances won't be able to connect to the S@N.IT! Server.

ServerRMIPort TCP port used by the Server for its communication with the S@N.IT! Agents and the S@N.IT! GUI/CLI instances.

MonitoringPeriod periodicity (seconds) for the polling of the monitoring status of the SAN components.

MonitoringLogSize maximum size (octets) of the circular file where the monitoring status changes are logged.

LACLogSize maximum size (octets) of the circular file where the LUN Access Control changes are logged.

TraceDirectory directory where the S@N.IT! traces will be stored.

TraceSize maximum size (octets) of each trace file.

JavaPath path of the Java run time to be used to launch the S@N.IT! software, when the default java version is not convenient. See details in Chapter 12. *Supported Platforms*

Configuration Parameters for S@N.IT! Data Center

S@N.IT! Data Center software uses the same configuration parameters as S@N.IT! Limited Edition, plus the following ones:

OutbandDiscoveryStartAddress1 dotted IP address (for example `123.123.123.123`).

OutbandDiscoveryStopAddress1 dotted IP address (for example `123.123.123.20`).

OutbandDiscoveryPort1 SNMP agents port (default=161).

The S@N.IT! discovery method will poll all IP addresses between **OutbandDiscoveryStartAddress1** and **OutbandDiscoveryStopAddress1** at UDP port **OutbandDiscoveryPort1** to check if a SNMP agent managing a SAN component resides there.

A second range of IP addresses for the S@N.IT! discovery method is provided with the following parameters:

OutbandDiscoveryStartAddress2
OutbandDiscoveryStopAddress2
OutbandDiscoveryPort2

A third range of IP addresses for the S@N.IT! discovery method is provided with the following parameters:

OutbandDiscoveryStartAddress3
OutbandDiscoveryStopAddress3
OutbandDiscoveryPort3

The three ranges of IP addresses must contain the IP addresses of the switches, libraries and storage devices supporting outband discovery (FDA storage devices).

DiscoveryRefreshPeriod periodicity (seconds) for the S@N.IT! discovery method, when checking for changes of discovered objects. If **DiscoveryRefreshPeriod** = 0, the discovery is disabled.

DiscoveryScanningPeriod periodicity (seconds) for the S@N.IT! discovery method when looking for new objects. If **DiscoveryScanningPeriod** = 0, the discovery is disabled.

ActivateTrapListener If set to true, the Server acts as an SNMP Manager and listens for traps coming from the SAN components for monitoring purpose.

TrapListenerPort	UDP port number where the Server waits for traps from the SAN components. **Note:** The default value (162) can be changed to prevent conflicts with another SNMP manager running on the host where the S@N.IT! Server runs. Pay attention that it is not always possible to configure the SAN components to send their traps to another port than this default value.
RunSnmpAgentOnStartup	If set to true, the S@N.IT! SNMP agent is started thus making the MIB management and the sending of SNMP traps operational.
SnmpAgentPortNumber	UDP port number where the S@N.IT! SNMP agent waits for SNMP GET/SET requests from SNMP Managers. If the default value (1610) is changed, the new value must not conflict with other UDP applications (such as the operating system SNMP services which usually use port 161), otherwise the S@N.IT! services would not be able to start.
SnmpManagerIp	Optional dotted IP address (for example `123.124.124.1`) of the SNMP manager to which SNMP traps must be sent (other SNMP managers can register themselves thru SNMP SET requests. See Chapter 8. S@N.IT! SNMP Agent).
SnmpManagerPort	Port of the default SNMP manager.
SnmpManagerFilter	Trap filtering level as defined in Fibre Alliance Mib (see Chapter 8. S@N.IT! SNMP Agent) for the default SNMP manager.
FileSystemMonitoringPeriod	Periodicity (seconds) for the refreshment of file systems information (location, contents). If FileSystemMonitoringPeriod = 0, the refreshment is disabled.
FileSystemUseRateThreshold	Default threshold (percentage) for the file systems usage rate: if a file system of a SAN agent host is over this limit, this host will be reported as faulty.This value can be customized for each file system (see *Information / File Systems* , on page 6-35), if needed.

Changing the sanadmin Password

For security reasons, it is highly recommended to modify the default password for **sanadmin** user. To change the sanadmin password, follow these steps:

1. Launch the S@N.IT! GUI.
2. Enter in the *Configuration/Set User* menu, and set current user to **sanadmin** (default password is **admin**).
3. Enter in the *Configuration/Set Password* menu, and change the password.

Enabling the Applet Mode for the S@N.IT! GUI

The S@N.IT! software delivery does not include a Web server but it contains an applet that can be used within a user–provided Web server (or the Web server included in the operating system). The **sanit.html** applet file is delivered into the following directory:

- /**usr/sanit/java** on AIX, Linux, and Solaris hosts,
- **<%installation directory>\java** on Windows hosts.

S@N.IT! Agent Configuration

The following parameters are used to configure a S@N.IT! Agent:

HostName IP name to be used by the S@N.IT! Server to connect to the S@N.IT! Agent; this parameter may be provided as:

– a dotted IP address (for example `123.123.123.123`),

– a string (for example `sanhost`): in this case the name resolution is performed on the S@N.IT! Server.

If this field is left blank, the default is the host name of the current machine.

ServerHost IP name to be used to reach the S@N.IT! Server– this parameter may be provided as:

– a dotted IP address (for example `123.123.123.123`),

– a string (for example `sanserver`): in this case the name resolution is performed locally.

AgentRole must be set to true.

InbandDiscoveryPeriod periodicity (seconds) for the S@N.IT! Agent discovery method. If this field is set to 0, discovery is disabled.

UserPreMappingCommand (optional) full path of the command to be called before each LUN(s) mapping action.

The command will be launched by the S@N.IT! Agent with the following syntax:

```
<command_path> model physical_identifier -l LUN_ID [LUN_ID]
-m portWWN adapterWWN [-m portWWN adapterWWN]
```

`command_path` value of the **UserPreMappingCommand** field in ***sanit.cfg.***

`model` model of the Disk subsystem to which the LUN belongs.

`physical_identifier` identifier of the Disk subsystem to which the LUN belongs.

`portWWN` WWN of the subsystem port.

`adapterWWN` WWN of the host HBA.

`LUN_ID` string of 16 hexadecimal digits (SCSI 3 address) that identifies a LUN to be mapped (Host LUN ID).

Example on Windows

```
"C:/WINNT/system32/wscript" "C:/myDir/myPreMappingScript.js"
"DAS_5700" "3002295" "-l" "0003000000000000" "0004000000000000"
"-m" "200000601636025C" "10000000C92134C0" "-m"
"200000601636CDD9" "10000000C92188EB"
```

The command is launched as a daemon, with the privileges of the user that has started the S@N.IT! services on the current host (usually **root** on AIX, Linux and Solaris and S@N.IT! **service account** on Windows). The command may not be interactive.

The exit code of the command must be (0) in case of success, greater than zero in case of failure. The mapping action is not executed in case of failure.

For details on the syntax of <command_path>, see Chapter 12. *Supported Platforms*.

UserPostMappingCommand = (optional) full path of the command to be called after each LUN(s) mapping action.

The command will be launched by the S@N.IT! Agent, with the following syntax:

```
<command_path> model physical_identifier -l LUN_ID [LUN_ID] -m
portWWN adapterWWN [-m portWWN adapterWWN]
```

`command_path` value of the **UserPostMappingCommand** field in **sanit.cfg**,

`model` model of the Disk subsystem to which the LUN belongs,

`physical_identifier` identifier of the Disk subsystem to which the LUN belongs,

`portWWN` WWN of the subsystem port,

`adapterWWN` WWN of the host HBA,

`LUN_ID` string of 16 hexadecimal digits (SCSI 3 address) that identifies a LUN to be mapped (Host LUN ID).

Example on Windows

```
"C:/WINNT/system32/wscript" "C:/myDir/myPostMappingScript.js"
"DAS_5700" "3002295" "-l" "0003000000000000" "0004000000000000"
"-m" "200000601636025C" "10000000C92134C0" "-m"
"200000601636CDD9" "10000000C92188EB"
```

The command is launched as a daemon, with the privileges of the user that has started the S@N.IT! services on the current host (usually **root** on AIX, Linux and Solaris and S@N.IT! **service account** on Windows). The command may not be interactive.

The exit code of the command must be (0) in case of success, greater than zero in case of failure.

For details on the syntax of <command_path>, see Chapter 12. *Supported Platforms*.

UserPreUnmappingCommand = (optional) full path of the command to be called before each LUN(s) unmapping action.

The command will be launched by the S@N.IT! Agent with the following syntax:

```
<command_path> model physical_identifier -l LUN_ID [LUN_ID] -m
portWWN adapterWWN [-m portWWN adapterWWN]
```

`command_path` value of the **UserPreUnmappingCommand** field in **sanit.cfg**.

`model` model of the Disk subsystem to which the LUN belongs.

`physical_identifier` identifier of the Disk subsystem to which the LUN belongs.

`portWWN` WWN of the subsystem port.

`adapterWWN` WWN of the host HBA.

`LUN_ID` string of 16 hexadecimal digits (SCSI 3 address) that identifies a LUN to be unmapped (Host LUN ID).

Example on AIX

```
/bin/ksh /etc/sanit/PreUnMappingCommand.template" DAS_5700"
"3002295" "-l" "0003000000000000" "0004000000000000" "-m"
"200000601636025C" "10000000C92134C0" "-m" "200000601636CDD9"
"10000000C92188EB"
```

The command is launched as a daemon, with the privileges of the user that has started the S@N.IT! services on the current host (usually **root** on AIX, Linux and Solaris and S@N.IT! **service account** on Windows). The command may not be interactive.

The exit code of the command must be (0) in case of success, greater than zero in case of failure. The unmapping action is not executed in case of failure.

For details on the syntax of the <command path>, see Chapter 12. *Supported Platforms*.

UserPostUnmappingCommand = (optional) full path of the command to be called after each LUN(s) unmapping action.

The command will be launched by the S@N.IT! Agent, with the following syntax:

```
<command_path> model physical_identifier -l LUN_ID [LUN_ID] -m
portWWN adapterWWN [-m portWWN adapterWWN]
```

`command_path` value of the **UserPostUnmappingCommand** field in **sanit.cfg**.

`model` model of the Disk subsystem to which the LUN belongs.

`physical_identifier` identifier of the Disk subsystem to which the LUN belongs.

`portWWN` WWN of the subsystem port.

`adapterWWN` WWN of the host HBA.

`LUN_ID` string of 16 hexadecimal digits (SCSI 3 address) that identifies a LUN to be unmapped (Host LUN ID).

Example on AIX

```
/bin/ksh /etc/sanit/PostUnMappingCommand.template" DAS_5700"
"3002295" "-l" "0003000000000000" "0004000000000000" "-m"
"200000601636025C" "10000000C92134C0" "-m" "200000601636CDD9"
"10000000C92188EB"
```

The command is launched as a daemon, with the privileges of the user that has started the S@N.IT! services on the current host (usually **root** on AIX, Linux and Solaris and S@N.IT! **service account** on Windows). The command may not be interactive.

The exit code of the command must be (0) in case of success, greater than zero in case of failure.

JavaPath path of the Java run time to be used to launch the S@N.IT! software, when the default java version is not convenient. See details in Chapter 12. *Supported Platforms*.

AllowRemoteUpdate if set to true, indicates that the S@N.IT! Administrator is allowed to push S@N.IT! software updates from the S@N.IT! Server for this agent.

The following parameters will generally not have to be modified:

ServerPort TCP port where the S@N.IT! Server waits for connections.

AgentRMIPort TCP port used by the S@N.IT! Agent for its communication with the S@N.IT! Server.

AgentPort TCP port used for internal communication between the S@N.IT! Agent processes. The default value (38001) must not be modified.

TraceDirectory directory where the S@N.IT! traces will be stored.

TraceSize maximum size (octets) of each trace file.

S@N.IT! GUI/CLI Configuration

The following parameters are used to configure a S@N.IT! GUI or CLI client:

HostName IP name to be used by the S@N.IT! Server to connect to the S@N.IT! GUI or CLI instance; this parameter may be provided as:

– a dotted IP address (for example `123.123.123.123`),

– a string (for example `sanhost`): in this case the name resolution is performed on the S@N.IT! Server.

If this field is left blank, the default is the hostname of the current machine.

ServerHost IP name to be used to reach the S@N.IT! Server– this parameter may be provided as:

– a dotted IP address (for example `123.123.123.123`),

– a string (for example `sanserver`): in this case the name resolution is performed locally.

ClientDisplay (only for AIX, Linux and Solaris) default DISPLAY value for the S@N.IT! GUI (optional).

ClientBrowser full path of the Web Browser application to be used within the S@N.IT! GUI to start a Web–based management application. The default value may need to be overridden depending on the current host installation.

JavaPath path of the Java run time to be used to launch the S@N.IT! software, when the default java version is not convenient. See details in Chapter 12. *Supported Platforms*

The following parameters will generally not have to be modified:

ServerPort TCP port where the S@N.IT! Server waits for connections.

ClientRMIPort TCP port used by the S@N.IT! GUI and CLI instances for their communication with the S@N.IT! Server. If several instances of the GUI or the CLI are running at the same time on the same machine, they use ports in sequence starting from this value.

ClientTimeout number of seconds after which a S@N.IT! CLI session will be closed if no command has been launched on that session. 0 stands for infinite.

Chapter 6. S@N.IT! Graphical User Interface (GUI)

This chapter describes the S@N.IT! GUI. The following items are detailed:

Launching the S@N.IT! GUI

The S@N.IT! GUI can be launched as a standalone application on any platform where the S@N.IT! software is installed:

On AIX, Linux and Solaris hosts:

Set the DISPLAY environment variable. If this variable is not set, the value of the field **ClientDisplay** in the *S@N.IT!* configuration will be used.

Launch the **sanit** command. See Chapter 7. *To start the* S@N.IT! *GUI on AIX, Linux and Solaris host*s.

On Windows hosts:

Use the **Start/Programs/S@N.IT!** menu.

From a Web Browser:

The S@N.IT! GUI is also accessible from a Java 1.4 enabled Web Browser, through the following URL:

http://< S@N.IT! Server hostname>/sanit.html

The Web server of the S@N.IT! Server must be correctly configured (see *Enabling the applet mode for the S@N.IT! GUI*, on page 5-6).

The S@N.IT! GUI always starts under **common** user.

S@N.IT! GUI Structure

S@N.IT! GUI Screens

The S@N.IT! GUI window looks as follows:

- The upper part contains general-purpose menus, shortcut buttons and the **View Selection** and **Reference Selection** combo boxes.
- The bottom part contains information about the GUI session (S@N.IT! user, state of the connection with the server).
- The middle part is divided vertically into a left part and a right part:
  - On the left side, the *View management* frame shows a tree list whose root is the view itself and where the branches build a hierarchical list of topology elements that have been discovered.
  - On the right side, the *Information* frame has three main tabs:

    Topology shows topology information for the whole view,

    Information provides access to a set of tabs, depending on the current selection:
    - **Properties**
    - **Local Topology** (SAN Component selected)
    - **Ports (**SAN Component selected)
    - **Monitor Log**
    - **Subsystem LUNs** (storage device selected)
    - **Host LUNs**
    - **LUN Access Log** (host selected)
    - **File Systems** (host selected)
    - **Change Logs**
    - **Contents** (view, group or complex component selected)
    - **Zoning** (fabric selected)

    Changes provides a set of tabs to understand the SAN evolutions:
    - **Reference**
    - **Comparison**
    - **Log**

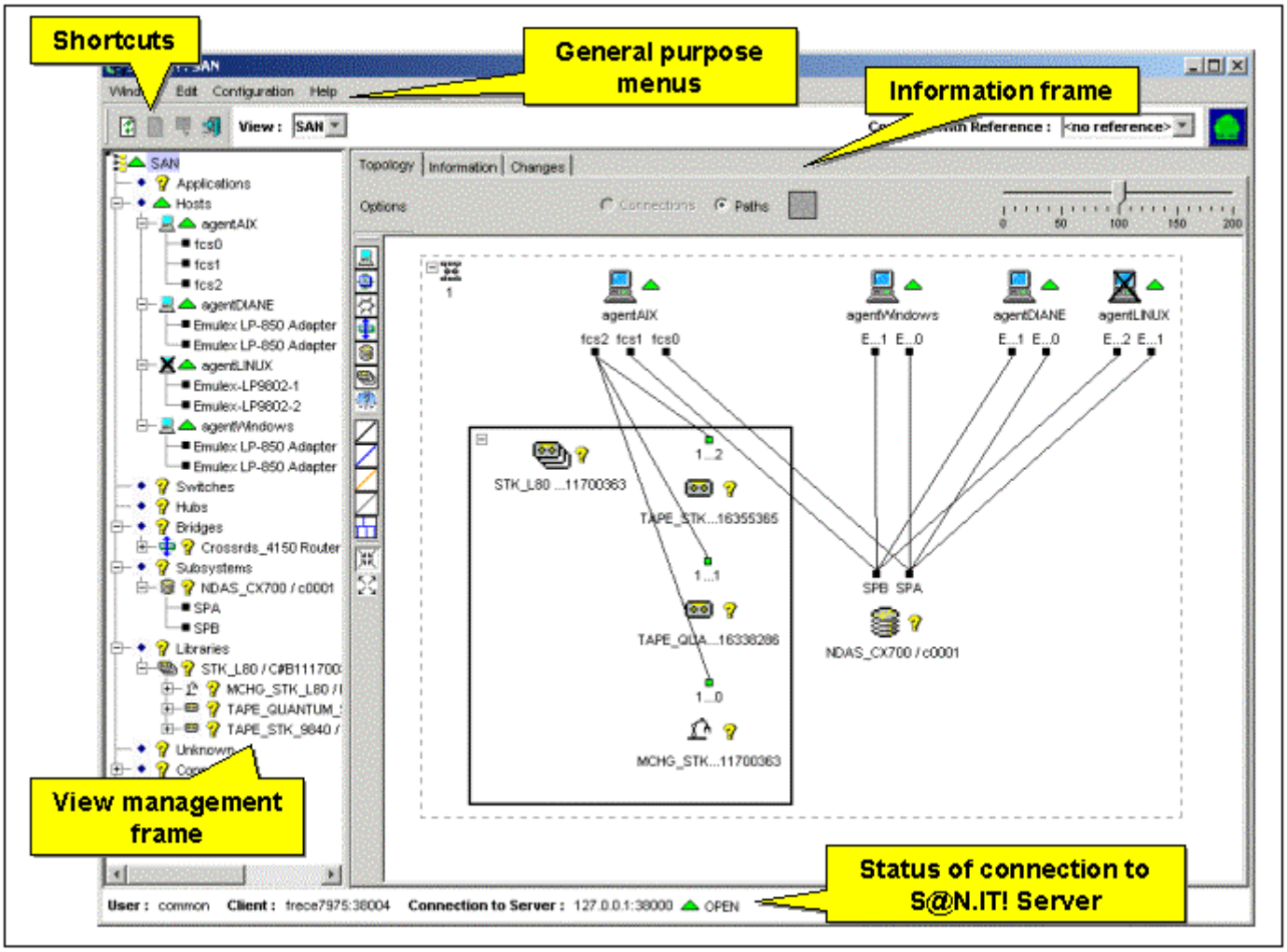

Figure 5. GUI initial window for S@N.IT! Limited Edition

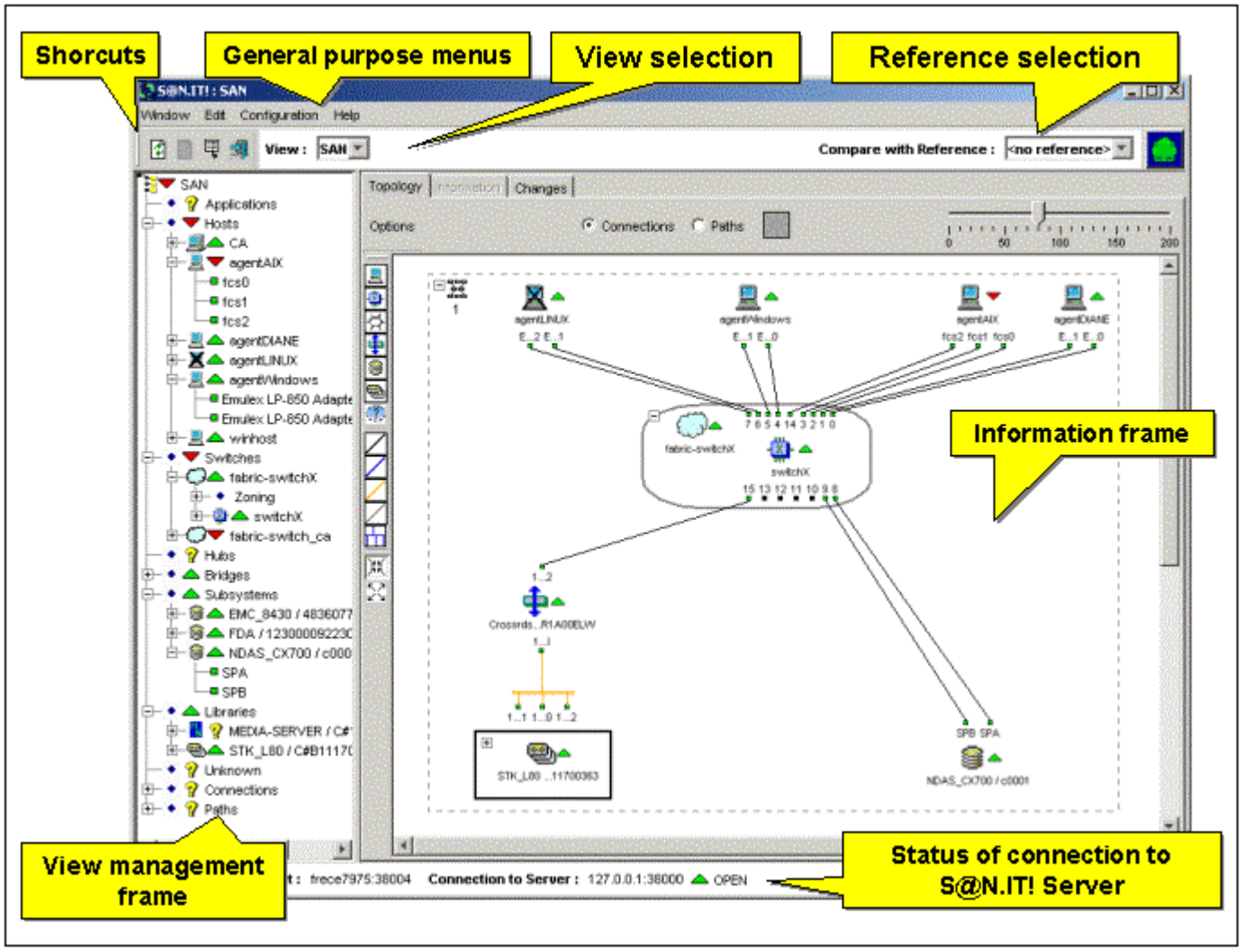

Figure 6. GUI initial window for S@N.IT! Data Center

Figure 7 shows the GUI window when a SAN component is selected.

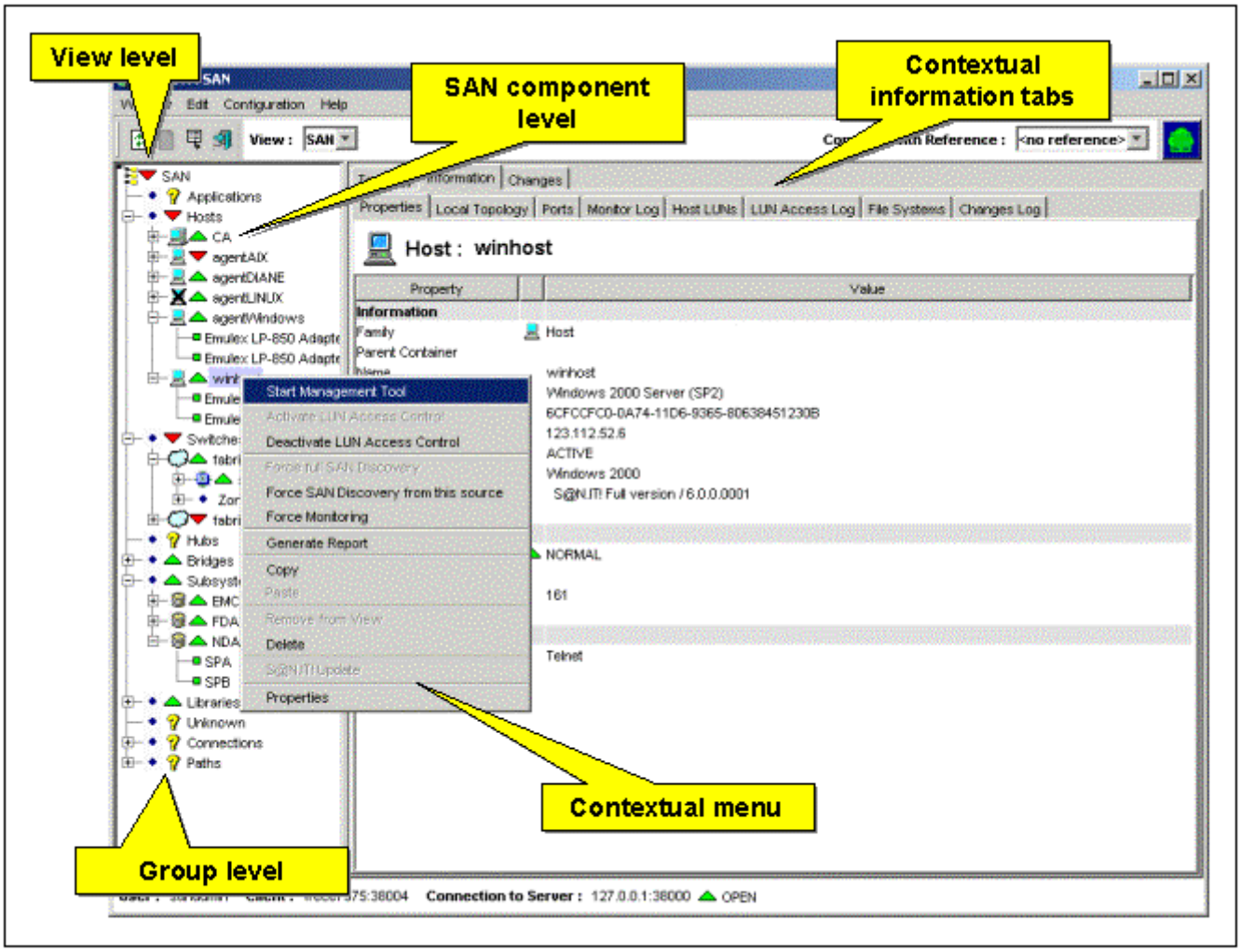

Figure 7. S@N.IT! GUI: information tab when a component is selected

Mouse Usage

Unless explicitly specified, the mouse buttons work as follows:

- A “Left–click” on an object selects the object.
- A “Right–click” displays a contextual pop–up menu.
- Actions on tables (*Ports*, *Monitor Log, Subsystem LUNs*, *Host LUNs, LUN Access Log*):
  - A “Click” on a column header sorts the table in ascending order for that column.
  - A “Shift+Click” on a column header sorts the table in descending order for that column.
  - A “Drag and drop” on a column header moves the column.

Icons

Icons are used to represent the SAN components and their monitoring status.

The icon representing a host running a S@N.IT! Agent depends on the state of the LUN Access Control mechanism:

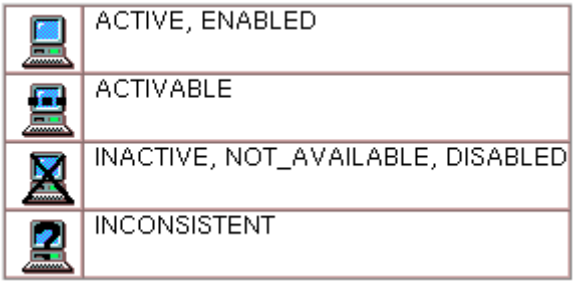

Figure 8. Host icons and LUN Access Control state

The icons representing other SAN components are the following:

Figure 9. SAN components icons

The monitoring status of a SAN component is represented by the following symbols:

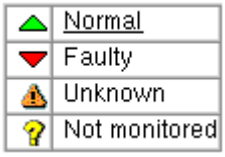

Figure 10. Monitoring status icons

Color Meaning in the Topology Frames

Square symbols are used to represent the fibre channel attachement – HBA (Host Bus Adapter), ports of switches or storage devices – in the topology frames. The square color indicates the attachement state, as described in the following table:

Color of the square	State of the fibre channel attachement
Green	normal
Black	not connected
Orange (S@N.IT! Agents and storage devices only)	unknown state (not yet analyzed by S@N.IT!) or no existing path from the HBA or to a storage device port
Red (S@N.IT! Agents and storage devices only)	a LUN mapping was performed from the HBA, but the corresponding path has disappeared.

The **lines** used to represent the physical connections in the topology frames take different colors according to the protocol used.

View Management (S@N.IT! Data Center only)

A *view* is a named subset of the SAN configuration that makes sense to the S@N.IT! administrator who defines it.

The default basic view (*SAN* view) contains all SAN components. The selection of a view (using the **View Selection** combo box in the upper part of the GUI) limits the display of information to the contents of the view.

S@N.IT! manages two types of views :

Collection their contents is limited to the user–selected SAN components.

Domain they contain two parts:

- an initial group of user-selected SAN components,
- a dynamic part that contains the “neighbours” of the SAN components in the initial groups: all SAN components that are either physically (connection) or logically (path) connected to one of the components of the initial group.

A view is defined using the *Edit / Create View* menu.

Definition of a View (Edit / Create View)

The *Edit/Create View* menu (available only to the *sanadmin* user) allows the creation of a new view. The user provides the following information:

Name the name of the view (must be unique for a S@N.IT! Server)

Comments optional field

Contents type can be either **Collection** or **Domain**

Once created the new view definition is stored by the S@N.IT! Server and is available in all running and future S@N.IT! GUI instances.

The view is created empty. It must be organised (optional) and filled (see *Copy/Paste* below).

Organisation of a View (Edit / Create Group)

The *Edit / Create Group* menu allows to create a named sub–tree level within a view. This operation requires **sanadmin** authority. It is not allowed in the “*SAN*” view and in “*Domain*” views.

Adding Components to a View (Copy / Paste)

The *Copy/Paste* operation requires **sanadmin** authority. The *Paste* operation is not allowed in the default view (*SAN*).

These operations may be easier to perform with two S@N.IT! GUI windows opened: one with the source view and the other with the target view. Use the *Window/New* window menu to open a new window.

To add a SAN component or group (sub–tree) to a “Collection” view:

1. Select the default view (*SAN*) or any other existing view in the **View Selection** combo box, then select a SAN component or a group (sub–tree) in that view and **Copy** it using either the **Edit**/**Copy** menu or the **Copy** item in pop–up menu (right–click on selected component or group).
2. Select the **collection** view in the **View Selection** combo box.
3. In the view frame, select the group where you want to add the copied items. It can be:
   - the view itself (root of the tree),
   - a group created using the **Edit**/**Create Group** menu.

4. **Paste** the copied items using the **Edit**/**Paste** menu or the **Paste** item in pop–up menu (right–click on selected group).

To add a SAN component to a "Domain" view:

1. Select the default view (*SAN*) or any other existing view in the **View Selection** combo box, then select a SAN component in that view and **Copy** it using either the **Edit**/**Copy** menu or the **Copy** item in pop–up menu (right–click on selected component).
2. Select the *domain* view in the **View Selection** combo box.
3. In the view frame, select the "*Initial Components*" group where you want to add the copied items.
4. **Paste** the copied items using the **Edit**/**Paste** menu or the **Paste** item in pop–up menu (right–click on "*Initial Components*" group).

Removing Components or Groups from a View (Remove from view)

The *Remove from view* operation requires **sanadmin** authority. It is not allowed in the "*SAN*" view.

To remove a SAN component or group (sub–tree) from a view:

1. Select the view in the **View Selection** combo box.
2. Select the SAN component or group ("*collection*" views only) in that view.
3. Remove it from the view it using the **Remove from view** item in the **Edit** menu or the pop–up menu (right–click on selected component).

Deleting a View

The *Delete a view* operation requires **sanadmin** authority. It is not allowed in the "*SAN*" view.

To delete a view:

1. Select the view in the **View Selection** combo box.
2. Select the root of the tree (the view itself).
3. Delete it using the **Edit**/**Delete** menu or the Delete item in the pop–up menu (right–click).

Editing the SAN Components (S@N.IT! Data Center only)

Adding a SAN Component (Edit/Create Component)

The *Edit/Create Component* menu allows to create a component that has not been automatically discovered. This new component can be a physical component, a complex component or an application.

This operation requires **sanadmin** authority.

To add a SAN component:

1. Select the component family.
2. Enter the component properties:
   - **Logical name**
   - **Model** (must be a supported model otherwise the component will not be monitored; see Chapter 11. *Supported SAN Components*)
   - **Physical identifier** (mandatory)
   - **IP address** and **SNMP port** (for monitoring and for out–band discovery purposes)
   - **Management tool**
   - **HW/FW level** (except for hosts)
   - **Parent identifier**: if the new component belongs to an existing set of physical components, this attribute contains the physical identifier of the set (Library, Host group or Fabric).

   For hosts:

   - **Operating system level**

   For switches:

   - **Role** in the Fabric (1 for master, 2 for slave)
   - **Domain identifier**
   - **Fabric WWN**.

The Physical identifier is the key used by S@N.IT! to uniquely identify SAN components (except for hosts that are identified by their names): two SAN components cannot have the same identifier. If the Physical identifier provided is already known by S@N.IT!, the user must confirm that he wants to supersede the existing component description.

Once created the new component definition is stored in the S@N.IT! Server and becomes available for all S@N.IT! GUI and CLI sessions. If the component is then discovered by the S@N.IT! Agents or the S@N.IT! Server, its initial properties will be updated, but an automatic discovery will not turn a SAN component into an "*Unknown*" SAN component.

Note that a manual creation is the only discovery method for hubs without SNMP agent and for applications.

Modifying a SAN Component (Properties)

This operation requires **sanadmin** authority.

To modify the properties of a SAN component:

1. Select the component in the "view management frame" or in the "topology" tab.
2. Right–click then select **Properties**.
3. Update the desired fields, having in mind the following rules:
   - **Family** can only be modified for "*Unknown*" SAN components.
   - **Physical identifier** can only be modified for manually created components.
   - The **Logical Name** of a host can only be modified for manually created hosts.
   - LUN Access Control of a host cannot be modified thru this menu. Use the contextual menu if needed.

– Modifying or setting the parent identifier moves the SAN component into an existing set of physical components (Library, Host group or Fabric).

The following figure shows a Properties window for a host.

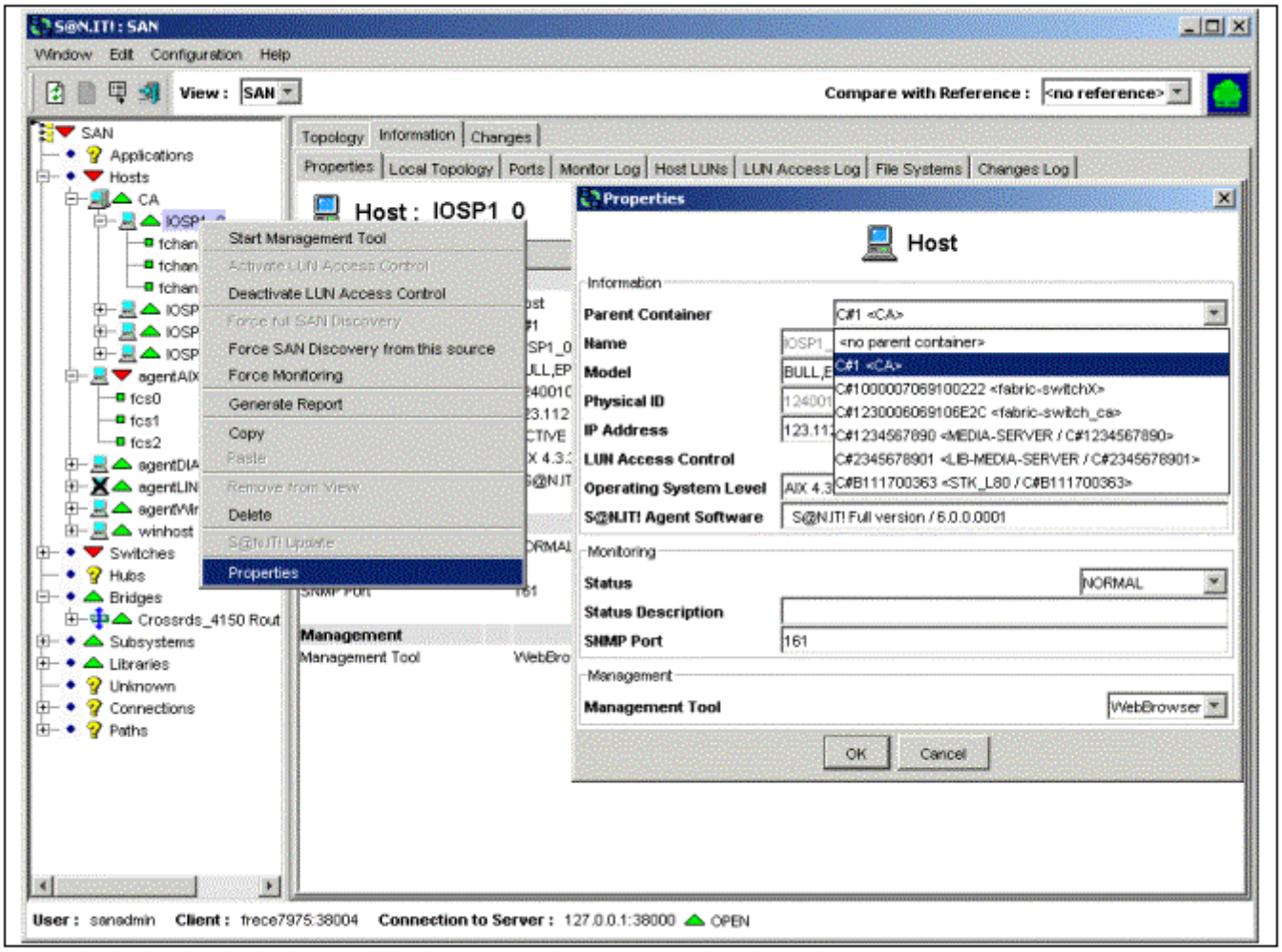

Figure 11. Host Properties window

Deleting a SAN Component (Delete)

This operation requires **sanadmin** authority.

1. Select the component in the "*view management frame*" or the "*Topology*" tab.
2. Right–click and select **Delete** item in the pop–up menu or in the **Edit** menu. A confirmation is requested.

The component, its ports and the related connections are deleted from the S@N.IT! Server database, but they may re–appear if they are discovered by a S@N.IT! Agent or the S@N.IT! Server.

Adding a Port (Edit/Create Port)

Adds a port to an existing SAN component. This operation requires **sanadmin** authority. The port properties must be specified. The following properties are mandatory:

– **Parent component**: identifies the SAN component to which the new port belongs.
– **WWN** (world wide name): must be unique.

Once created the new port definition is stored in the S@N.IT! Server and becomes visible to all GUI and CLI instances.

Modifying a Port (Properties)

This feature is available when a port is selected in the topology frame or in the View Management frame.

The pop–up window allows to modify all properties (except for world–wide name), thus also allowing to attach the port to another SAN component.

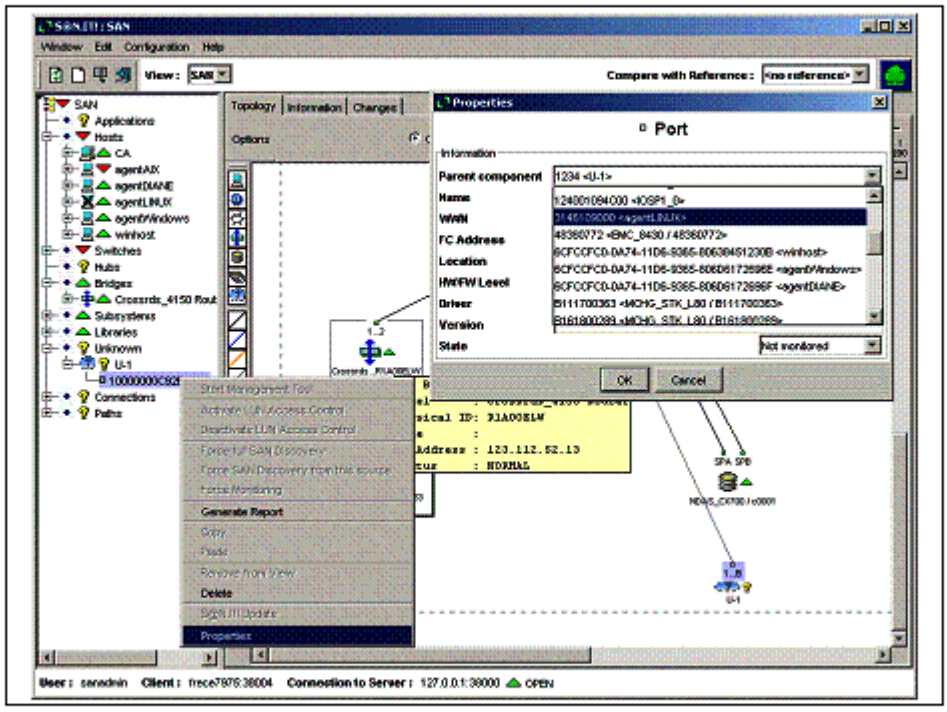

Figure 12. Port Properties menu

A further refresh of the topology will move the port back to its initial owner unless the initial owner is reported as an ”unknown” component. In that case, the ”unknown” component will disappear and the port will remain attached to the parent specified by the S@N.IT! Administrator.

Deleting a Port (Delete)

This feature is available when a port is selected in the Topology frame or in the View Management frame.

The port is deleted from the S@N.IT! Server, but it may re–appear if discovered by S@N.IT! Agents or Server.

Adding a Connection (Edit/Create Connection)

Creates a connection between two Fibre Channel ports. This operation requires **sanadmin** authority.

The pop–up windows allows to select both ends of the connection among existing ports and to specify a label for the connection.

Modifying a Connection (Properties)

This feature is available when a connection is selected in the Topology frame or in the View Management frame.

The pop–up window allows to add (or to modify) a label to the connection.

Deleting a Connection (Delete)

This feature is available when a port is selected in the Topology frame or in the View Management frame.The connection is deleted from the S@N.IT! Server, but it may appear again if it is discovered by S@N.IT! Agents or Server.

Deleting a Path (Delete)

This feature is available when a path is selected in the Topology frame or in the View Management frame. It requires **sanadmin** authority. The path is deleted from the S@N.IT! Server, but it may appear again if it is discovered by a S@N.IT! Agent.

Understanding SAN Changes

This section describes how to create, compare, delete SAN configurations references.

Saving the Current State of the SAN Configuration as a Reference

The state of the SAN configuration (list of discovered SAN component and their properties; paths and connections) can be saved as a reference at any time using the *Edit/Save current SAN* menu and specifying a name and a description. This reference is stored in the S@N.IT! Server and can be selected for further comparison.

This operation requires **sanadmin** authority.

Comparing the Current SAN State to a Reference

A reference for comparison can be selected using the **Reference Selection** combo box. This box displays all references that have been saved.

Note: This feature is available only for the default SAN view.

What happens once a reference has been chosen:

- The **Topology** tab background changes and the differences against the reference are highlighted:
  - Objects that have been modified, deleted and created are displayed with a different background.
  - The tooltip information that appears when moving the mouse over a topology object indicates whether this object has changed.
- The **Changes** tab is modified:
  - The **Log** sub–tab lists all modifications since the selected reference has been saved.
  - The **Reference** sub–tab describes the contents of the selected reference.
  - The **Comparison** sub–tab becomes active (see below).

Deleting a Reference

Note: This operation requires **sanadmin** authority.

To delete a reference, proceed as follows:

- Select the reference to be deleted using the **Reference Selection** combo box.
- Select the *Changes/Reference* tab.
- Check the **Delete this reference** button.
- Click **Yes** in the confirmation pop–up window.

The reference is deleted on the S@N.IT! Server and can no longer be selected.

Comparing Two References

The *Changes/Comparison* tab compares two references. Proceed as follows:

- Select the older reference using the **Reference Selection** combo box.
- Select the *Changes/Comparison* tab.
- Select the newer reference (or the **Current SAN**) in the **Compared SAN** combo box.

The comparison tab now displays all changes detected between both references (one line per modified property). The objects are identified by the **Item** (type of object) and **Identifier** columns. The **Change** column describes the type of modification:

- ADDED
- REMOVED
- NOT_DISCOVERED

- DISCOVERED
- MODIFIED.

When a * character follows this information, it means that, although the property is identical in both references, it has changed then has returned to its initial value.

The **Property** column describes which property has been modified. The reference value and the compared value columns display the value of the property in both references

Note that both values can be identical if the property has changed then has returned to its initial value.

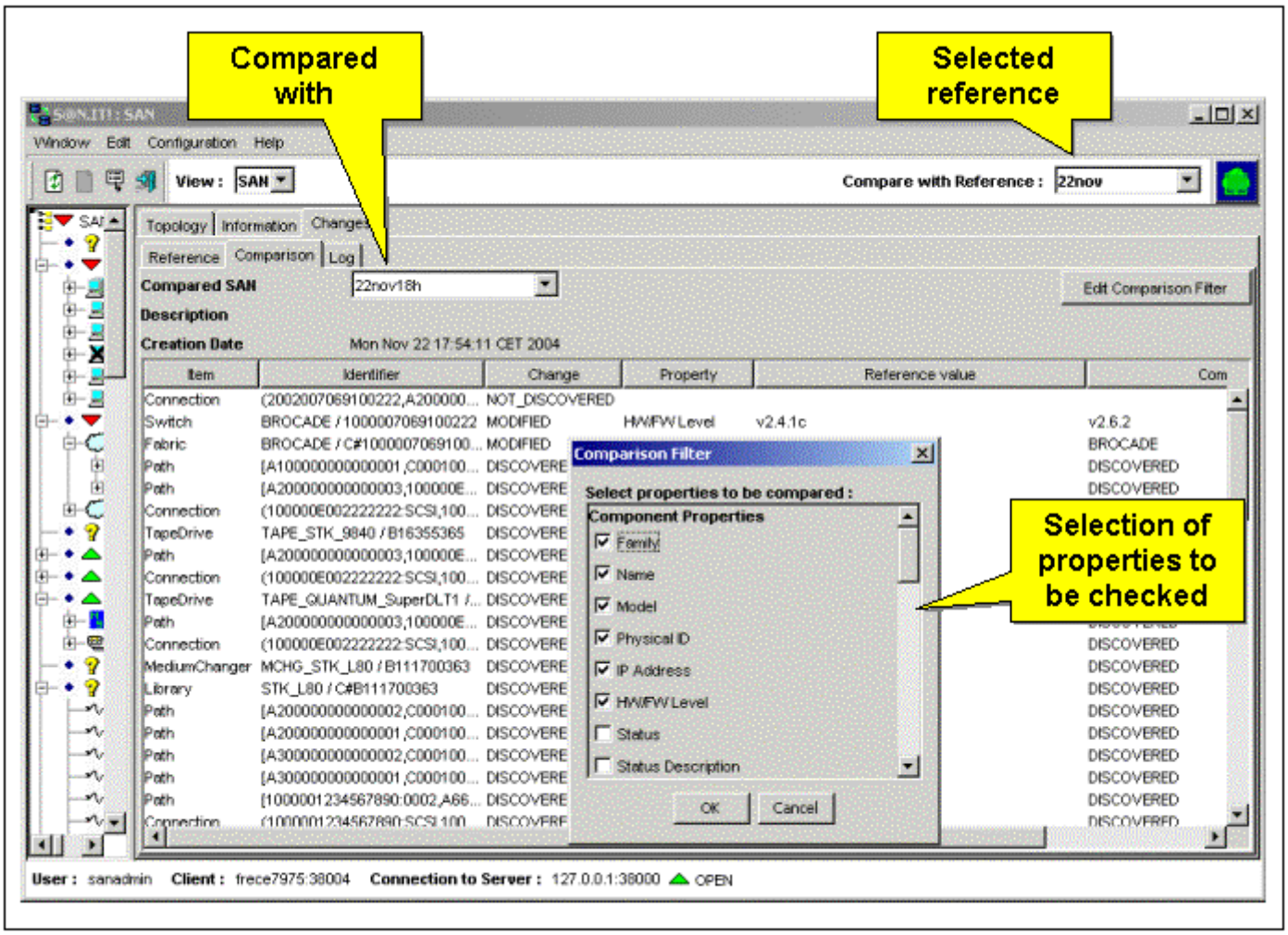

Figure 13. Comparison of two references

It is possible to select the properties to be compared by clicking the **Edit Comparison Filter** button in the *Changes/Comparison* tab. The **Comparison Filter** window that appears lets you select these properties, as in the above figure.

Zoning (S@N.IT! Data Center only)

The GUI allows understanding the zoning configuration of a fabric:

The *Information/Zoning* tab, available when a fabric is selected, describes how zoning is configured (see *Information / Zoning (fabric selected)*, on page 6-36).

The **View Management** frame displays zoning information as subtrees of a fabric, using the following icons:

C (green)	Active configuration
C (black)	Inactive configuration
Z	Zone
A	Alias
M	Member

When the **Topology** frame is displayed, and you click a Zone in the View Management tree, the ports contained in the zone are highlighted, as illustrated in the figure below.

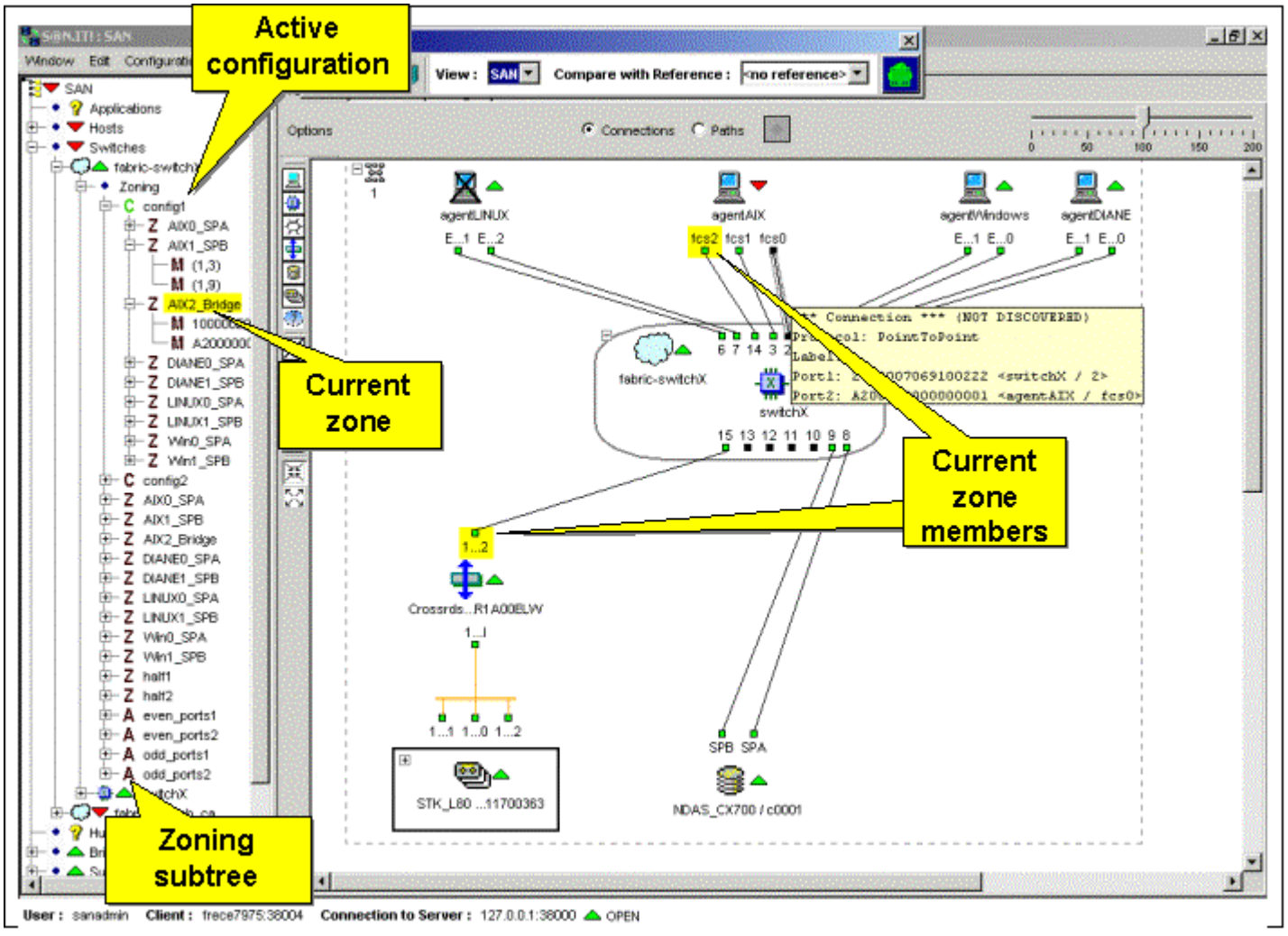

Figure 14. Zoning display

S@N.IT! GUI Menus

This section describes the general purpose menus of the S@N.IT! GUI:

- **Window** menu
- **Configuration** menu
- **Edit** menu

It also describes the contextual menus, available by a right-click when a SAN component is selected.

Window menu

Window / **New Window**
Creates a new Window for the same S@N.IT! GUI session.

Window / **Refresh**
Forces the S@N.IT! GUI to refresh the displayed frames. The duration of this operation depends on the information tab currently displayed.

Window / **Erase Log**
Deletes the entries in the currently displayed log (Changes/log, Information/LUN access log, Information/Monitor log or Information/Change log) from the S@N.IT! Server. This operation requires **sanadmin** authority.

Window / **Save table to a file**
Saves the currently displayed table in a file on the machine where the GUI is launched. The pop–up window allows you to select the file name, its location and its format (CSV or HTML). This feature is available wherever a table is displayed in the information frame (log tabs, contents tab).

Window / **Close**
Closes the current Window.

Window / **Exit**
Closes all Windows and exit the S@N.IT! GUI session.

Configuration menu

Configuration / Edit Current S@N.IT! GUI Configuration

This menu allows changes to the configuration for the current S@N.IT! GUI session:

- selection of the S@N.IT! Server,
- selection of the Web Browser to be used when launching Web applications.

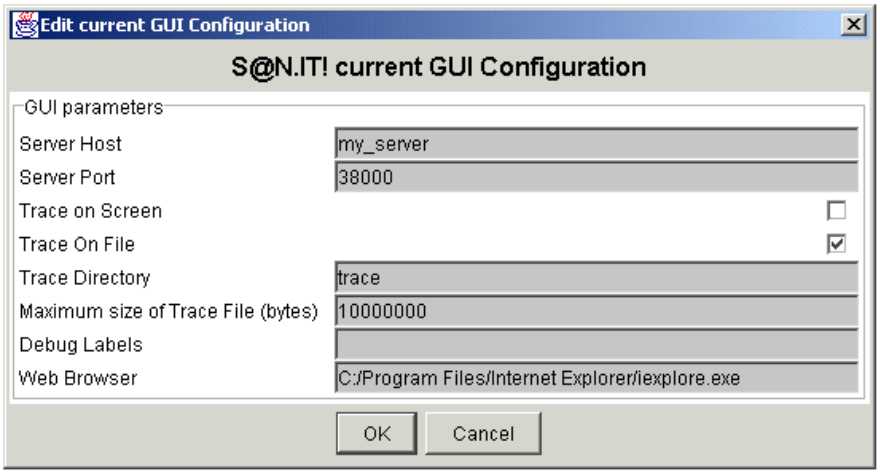

Figure 15. Configuration menu – Current S@N.IT! GUI Configuration

Configuration / Edit S@N.IT! Configuration

This menu allows changes to the S@N.IT! configuration for the local platform.

As this feature leads to a modification of the **sanit.cfg** configuration file:

- the S@N.IT! GUI must have been launched with **root** (AIX, Solaris or Linux) or **administrator** (Windows) privileges,
- the S@N.IT! services must be re–started after the modifications have been saved (/**etc**/**rc.sanit** on AIX, Linux and Solaris ; start/stop S@N.IT! Scheduler services on Windows).

This menu displays a window, where the configuration parameters are arranged in four tabs:

- *Common parameters*, (Figure 16)
- *Client parameters* (S@N.IT! GUI/CLI), (Figure 17)
- *Server parameters* (if *Enable Server Role* is checked in the *Common parameters* tab), (Figure 18)
- *Agent parameters* (if *Enable Agent Role* is checked in the *Common parameters* tab) (Figure 19).

Refer to Chapter 5. *Configuration and Launching* for a detailed description of each field.

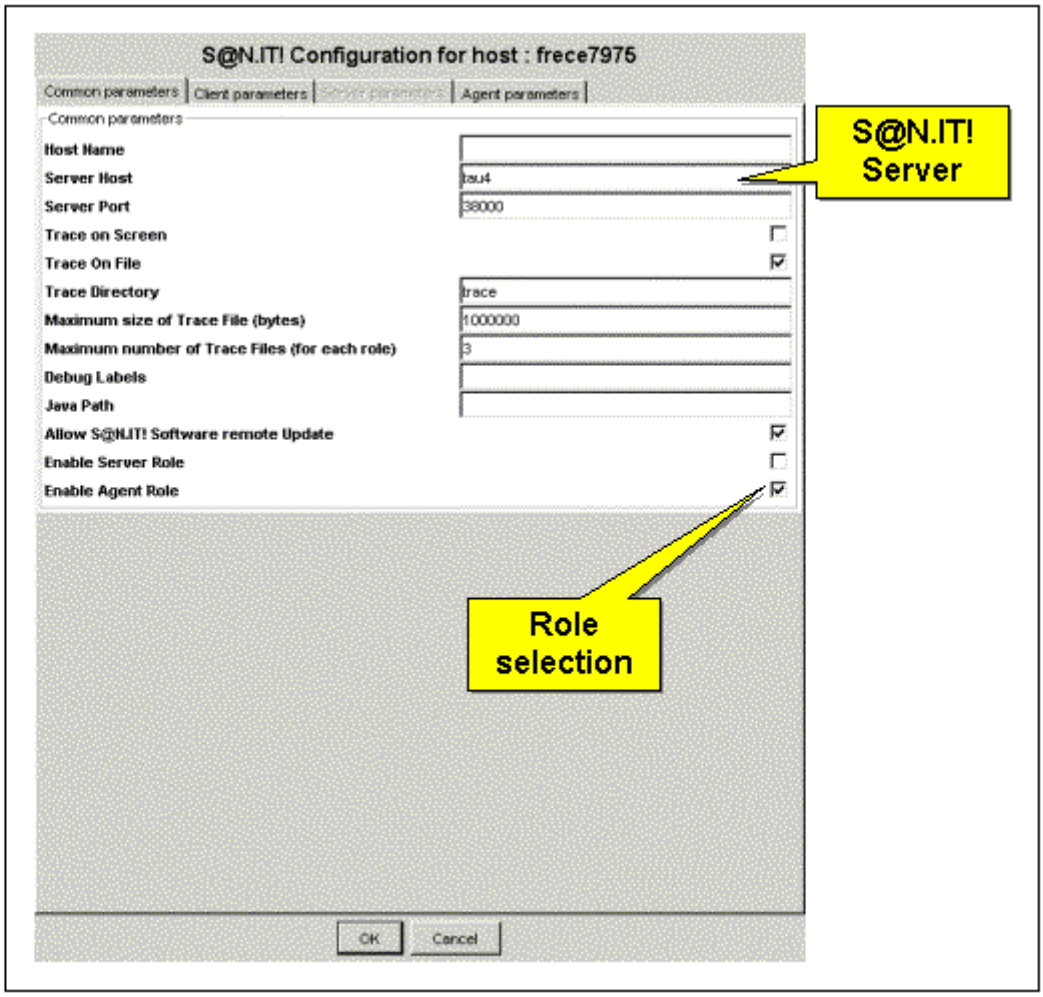

Figure 16. Configuration menu – Common parameters

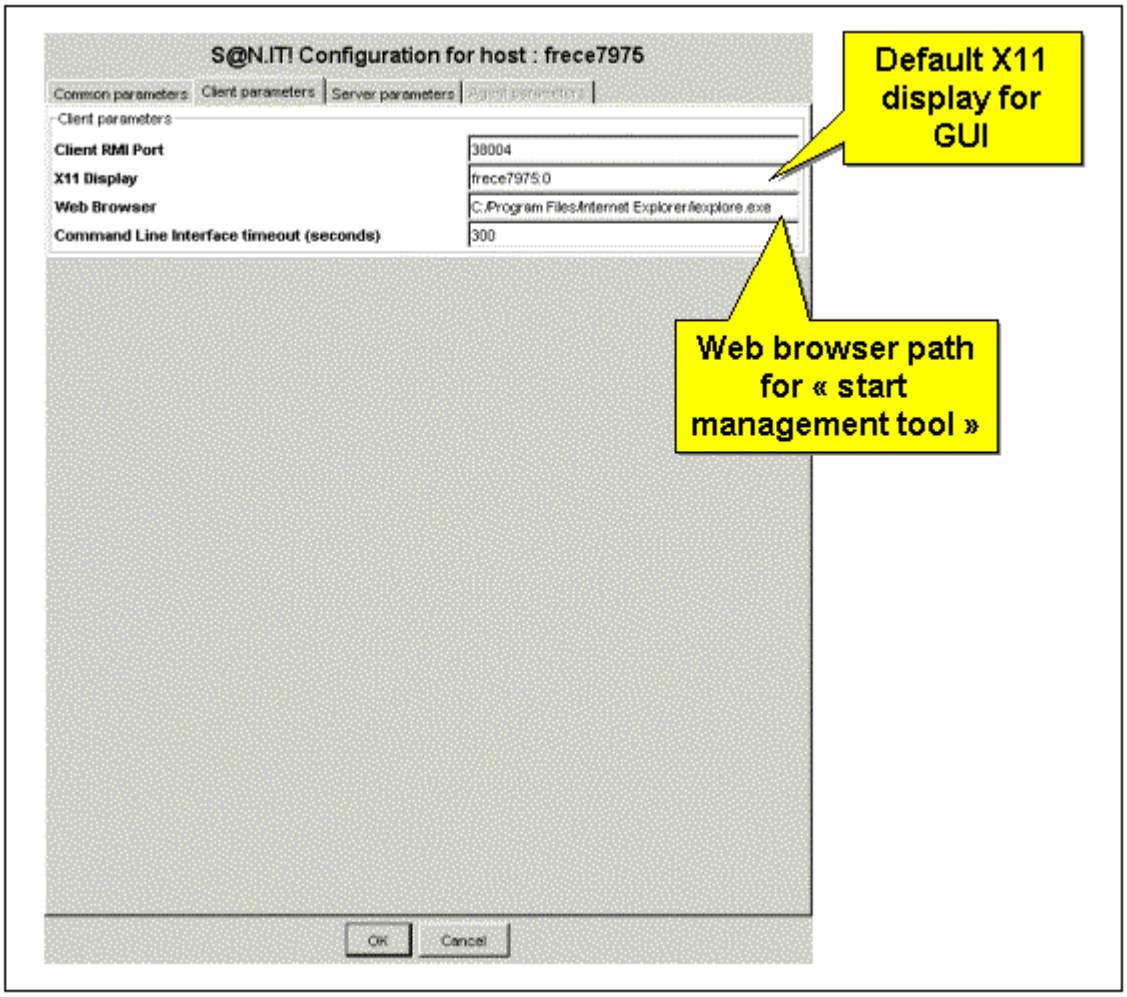

Figure 17. Configuration menu – Client parameters

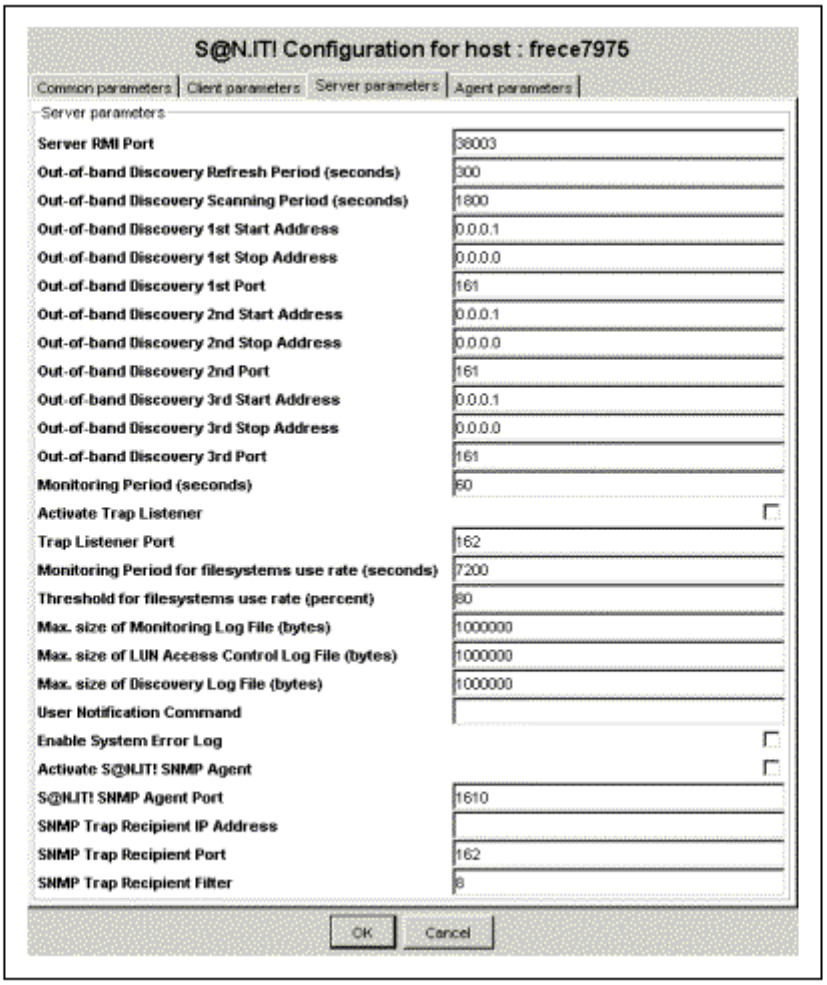

Figure 18. Configuration menu – Server parameters

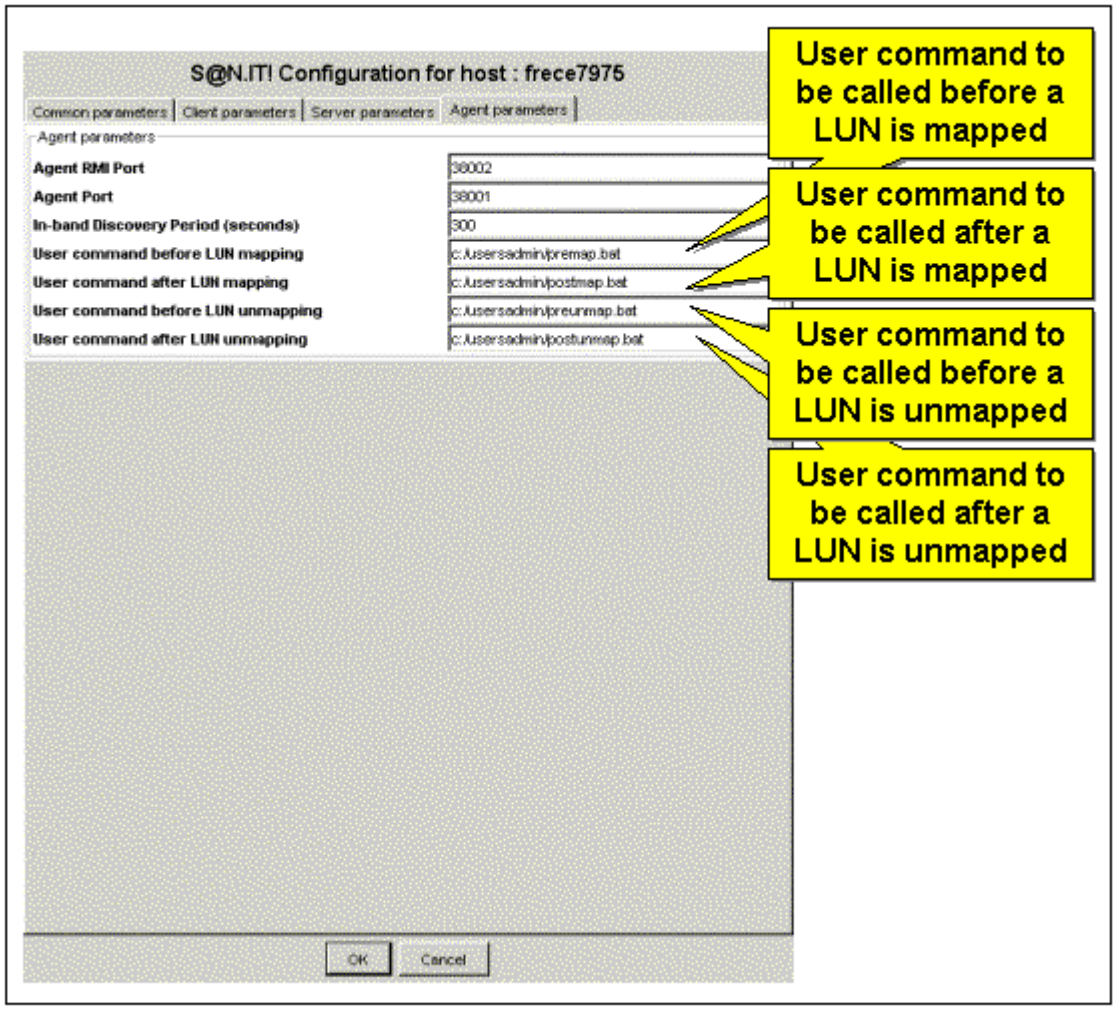

Figure 19. Configuration menu – Agent parameters

Configuration / Show Current Sessions

This menu launches a pop–up menu that displays the current connections to the S@N.IT! Server (GUI, CLI and SAN agents).

Configuration / Edit Report Templates

This menu allows to create, modify and delete report templates.

A template definition contains:

- The list of tables to be included in the reports
- For each table, the columns to be included and their order.

When generating a report (see *Generate Report* , on page 6-29), the template definition will be used to choose the information displayed in the final report for the selected component (or group of components, or all components of the selected view).

Tables are classified into the following report types:

- Properties:
  - Component properties table
  - Ports table
  - Paths table
  - Connections table

- Allocation of LUNs for hosts
  - Allowed LUNs table
  - All LUNs table
- Allocation of LUNs for storage devices
- LUN access control log
- Monitoring log
- File systems for hosts:
  - File systems usage rate table
  - File systems location table
  - GCOS7 volumes table
- Zoning for fabrics
  - Configurations
  - Zones
  - Aliases
  - Members
- Change logs

A default template (**SanFullReport**) is delivered: it contains all tables with all columns. Template definitions are stored on the S@N.IT! Server, and are thus available for all S@N.IT! GUI and CLI sessions.

To create a new template do as follows:

1. Select the ”Report Templates” line in the left part of the pop–up window.
2. Enter the name of the template (no spaces nor tabs allowed).
3. Select the template to be used as a model.
4. Press the ”OK” button.

The new template must then be modified to select information.

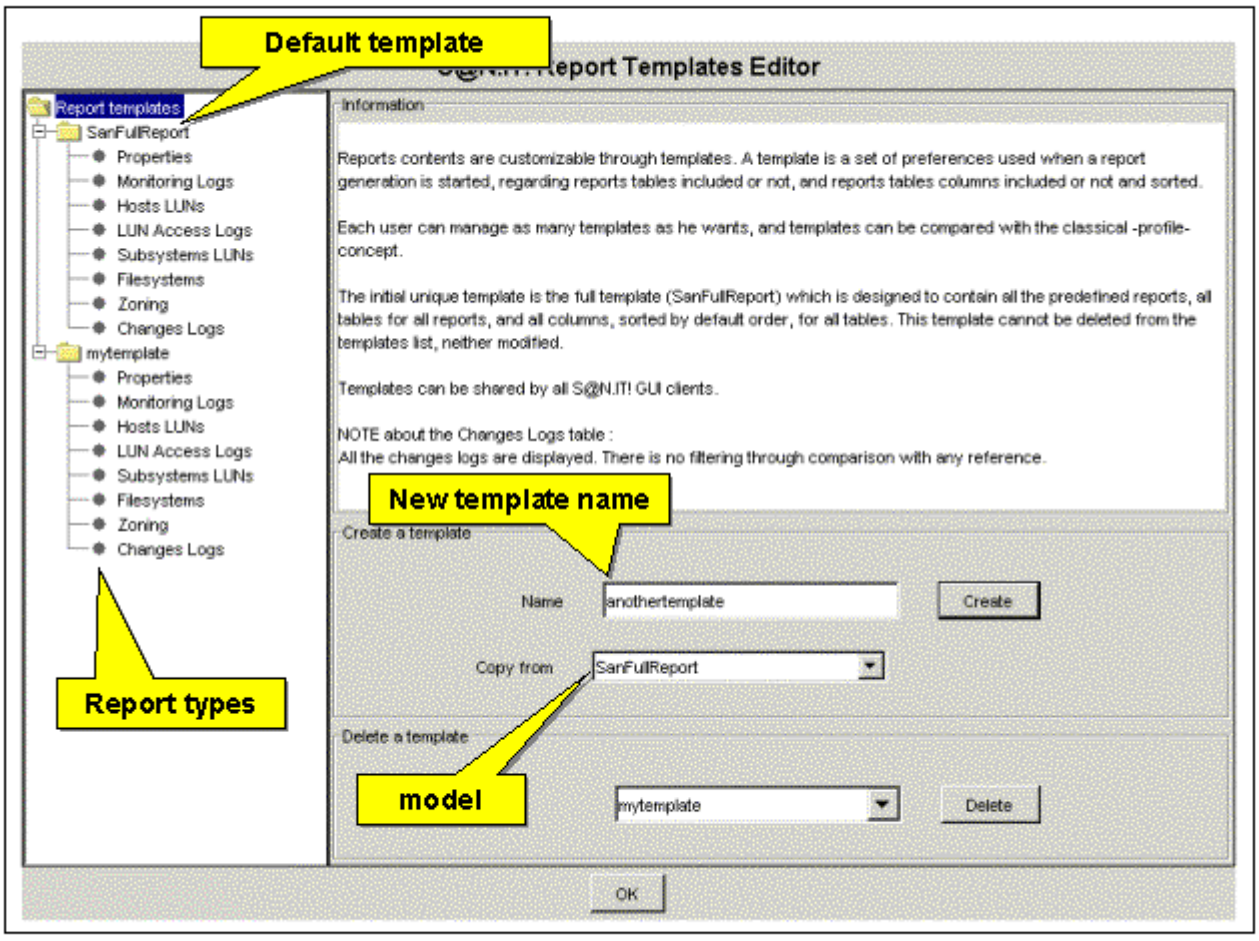

Figure 20. Creation of a report template

To modify a template (forbidden for SanFullReport template), do as follows:

1. Select the report types to be modified under the template name in the left part of the pop–up window.
2. For each table in a report type:
   a. Check or uncheck the “included in report” checkbox, depending on whether the table must appear in the reports.
   b. Select the columns that will appear in the reports, by moving the column names between the “Visible columns” list and the “Hidden Columns” list.
   c. Modify the columns order using “drag–and–drop” operations.
3. Press the **Save template** button.

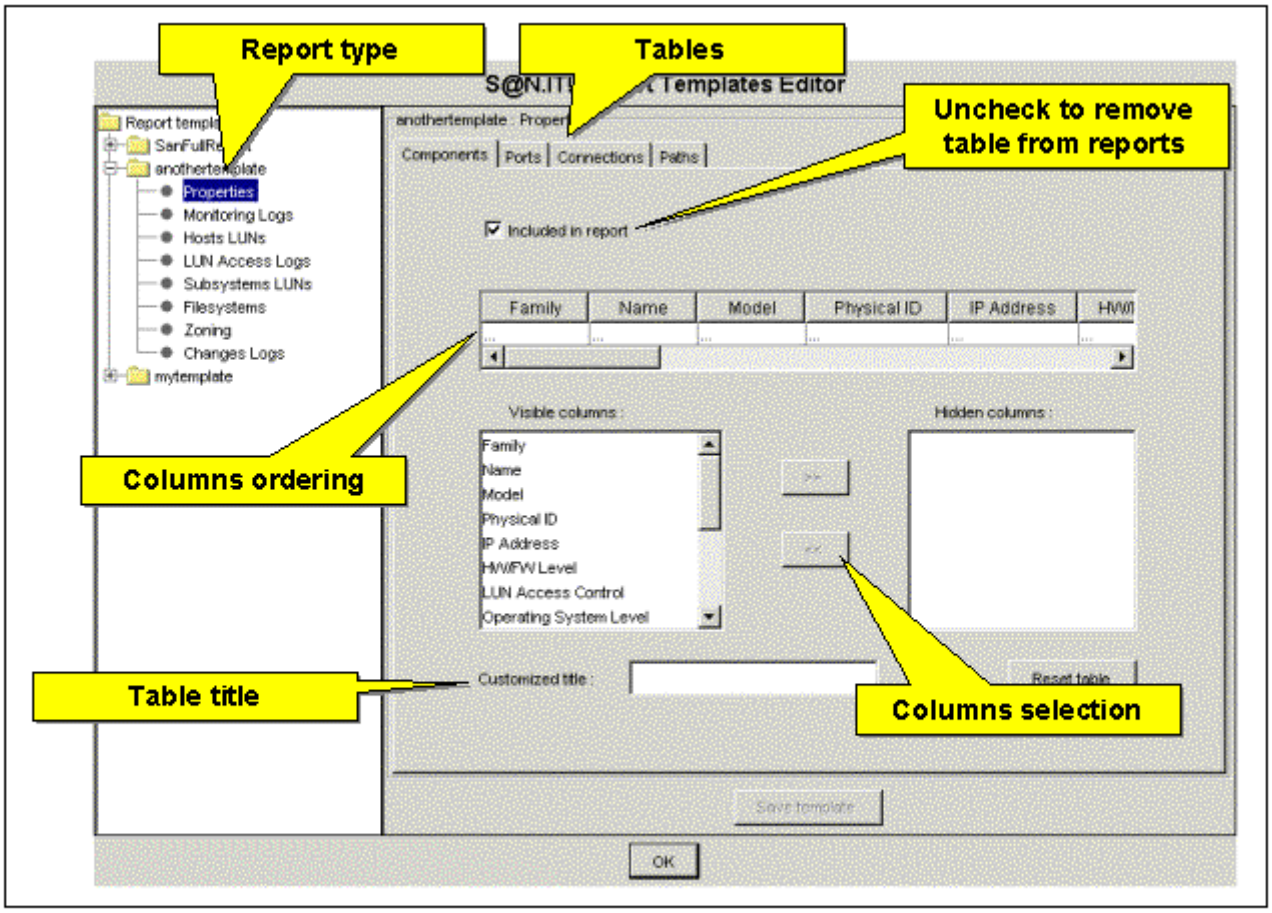

Figure 21. Modification of a report template

To check a template definition, do as follows:

1. Select the template name in the left part of the pop–up window.
2. Press the **preview** button in the right part of the window.

Configuration / Set User

This menu launches a new window that allows to select the S@N.IT! account (**sanadmin** or **common**) under which the S@N.IT! GUI is run.

This operation is mandatory to:

- move to **sanadmin** account, as the S@N.IT! GUI is always launched under *common* account,
- or to enter the **common** account's password if one has been set by the S@N.IT! administrator.

Figure 22. S@N.IT! user selection menu

Once the operation has been validated (OK button), the S@N.IT! GUI is refreshed or an error message is displayed if either the password is incorrect, or there is already a S@N.IT! GUI or CLI session running under **sanadmin** account for the same S@N.IT! Server.

Configuration / Set password

This menu is available only when the S@N.IT! GUI is running under **sanadmin** account. It allows to modify the **sanadmin** password or to set a password for the **common** account.

If the password for the **common** account is modified:

- all CLI sessions launched under this account are closed,
- all S@N.IT! GUI instances launched under this account have their connection to the S@N.IT! Server closed and receive a message requesting to enter the **common** password (using the Configuration / Set User menu).

Edit Menu

Edit / Create View

See *Organisation of a view*, on page 6-8.

Edit / Create Group

See *Organisation of a view*, on page 6-8.

Edit / Create Component

See *Adding Components to a View*, on page 6-8.

Edit / Create Port

See *Adding a Port*, on page 6-11.

Edit /Create Connection

See *Adding a Connection*, on page 6-12.

Edit / Find

Searches for an exact matching of the string provided as search criteria among the properties of all SAN components, ports and connections.

The '*' character is the wildcard character.

The objects matching the search criteria are displayed in the lower part of the pop–up window. The selection of one object in that window allows to select the object in the topology frame.

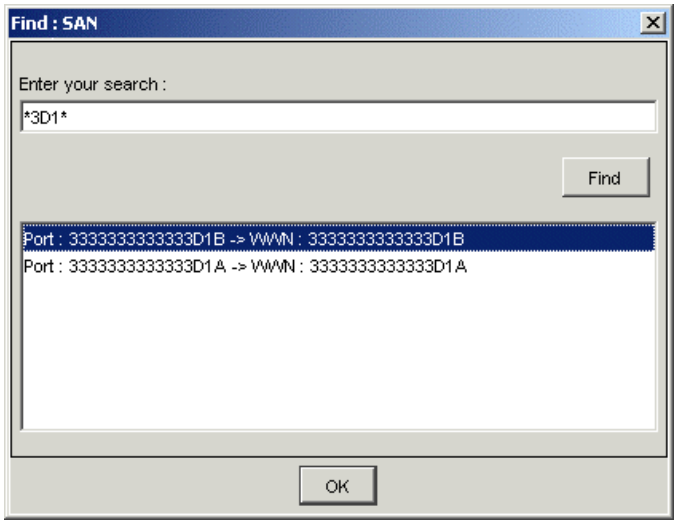

Figure 23. Find pop–up window

Edit / Copy

See *Adding Components to a View*, on page 6-8.

Edit / Paste

See *Adding Components to a View*, on page 6-8.

Edit / Remove from view

See *Removing Components or Groups from a View*, on page 6-9.

Edit / Delete

This operation requires *sanadmin* authority. It deletes the object currently selected from the S@N.IT! Server database, and it removes the object from all views. The object will re–appear in the default “SAN” view if it is further discovered.

Edit / Delete the objects no longer discovered

This menu is similar to the Edit / Delete menu but all objects (SAN components, connections, paths) that are no longer discovered (these objects are highlighted in the topology tab) are deleted at once.

Edit / Save the current SAN as a reference

See *Saving the Current State of the SAN Configuration as a Reference*, on page 6-14.

Contextual Menus

These menus appear when a right–click is performed while an object is selected in the *View management frame* or in the *Topology* tab.

Start Management Tool (S@N.IT! Data Center only)

This menu allows to launch the management tool associated with a particular SAN component. This operation requires **sanadmin** authority.

A first window is displayed that allows to:

- select the management application to be run (depending on the SAN component and the host where the S@N.IT! GUI is running, several applications can be available),
- select the platform where to launch the application (local S@N.IT! GUI machine or S@N.IT! Agent),
- modify the command line or URL (for a Web type application) if necessary (the changes are valid for the current action only),

- modify the template of the command line or URL (the changes will be saved and will be valid for next calls if the Apply button is pressed).

Some URL/Command templates contain the following variables (their actual value is visible in the URL/Command field):

  - %SanConfig.ClientDisplay% : refers to the current X11 display used by the S@N.IT! GUI
  - %SanObject.CompoIpAddress% : refers to the IP Address property of the SAN component to be managed.

Notes:

- If the command is to be executed locally it will be launched with the privileges of the user that has run the S@N.IT! GUI.
- If the command is to be executed on a S@N.IT! Agent, it will be launched with the privileges of the user that has started the S@N.IT! services on that agent (**root** on AIX, Linux and Solaris; **service account** on Windows).
- If the S@N.IT! GUI is launched in applet mode through a Web Browser, it is not possible to launch the management tool on the local machine (the one where the Web Browser is running).

The command is then launched as a separate process; a new window is created that reports information about the process start. This window can be closed as soon as the management tool is started. The tool itself becomes independent and must be closed explicitely.

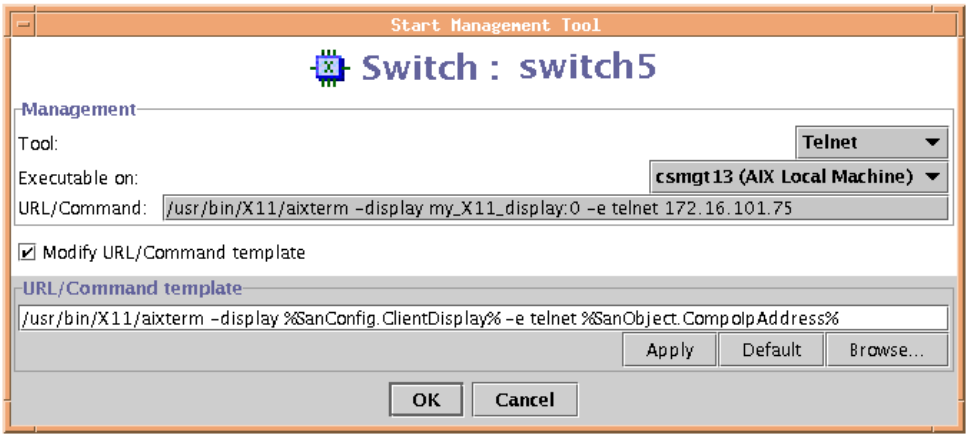

Figure 24. Start management tool window

Activate LUN Access Control (host selected)

Activates the LUN Access Control mechanism on the selected S@N.IT! Agent. The result of the operation is displayed in a pop–up window. The LUN Access Control mechanism on the host moves to the ACTIVABLE state and the host must be rebooted for the LUN Access Control to become ACTIVE (see *LUN Access Control mechanism*, on page 2-6).

De–activate LUN Access Control (host selected)

De–activates the LUN Access Control mechanism on the selected S@N.IT! Agent. The result of the operation is displayed in a pop–up window. The LUN Access Control mechanism on the host moves to the INACTIVE state (see *LUN Access Control mechanism*, on page 2-6). Remember that for safety reasons, this operation should not be performed while the host is connected to the SAN.

Force full SAN discovery (“SAN” view selected) (S@N.IT! Data Center only)

Request the full S@N.IT! discovery process to be restarted: exploration of the range of IP addresses specified in the configuration of the S@N.IT! Server, refresh of the information provided by the SAN components (mainly S@N.IT! Agents and switches) already discovered.

Force SAN Discovery from this source (S@N.IT! Data Center only)

If a host is selected, a pop–up window appears and the S@N.IT! Administrator may decide to force a reconfiguration of the operating system (the reconfiguration process depends on the type of operating system), before launching the discovery method of the S@N.IT! Agent.

For other families of SAN components, the S@N.IT! Server refreshes the discovery information provided by these components. A warning message is issued if no discovery information can be got from one component (IP address not set, no discovery method for the component model...).

Force Monitoring (SAN component selected)

Requests to refresh the monitoring status.

Generate Report (S@N.IT! Data Center only)

Generates a report for the selected object (component, group, view).

The first popup window allows to select a report template and report types (see *Configuration/Edit Report Templates* , on page 6-22 for details).

The second popup window allows to select the report format (HTML, CSV...) and the file where the report is saved.

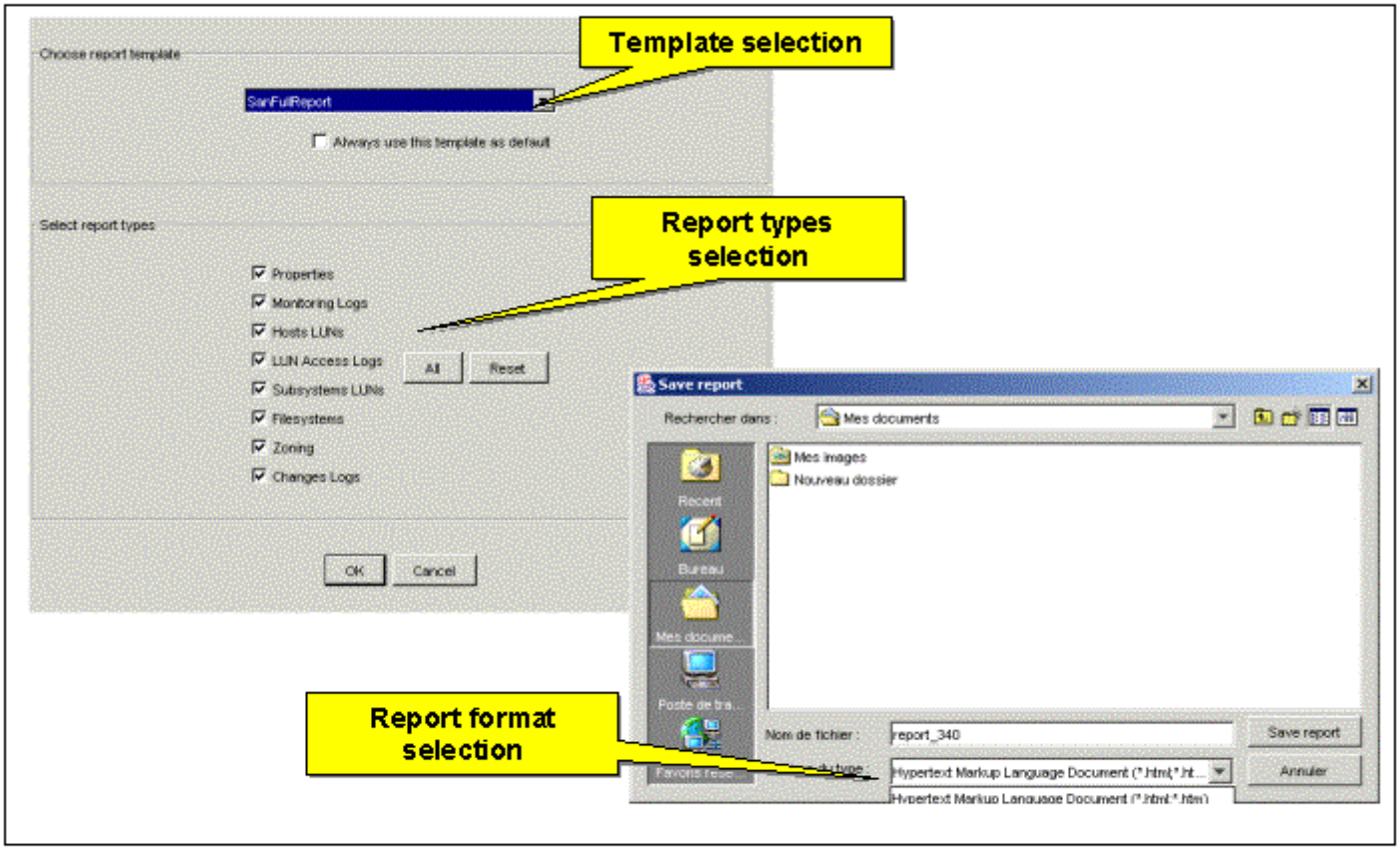

Figure 25. Report generation

Copy

See *Adding Components to a View*, on page 6-8.

Paste

See *Adding Components to a View*, on page 6-8.

Remove from view

See *Removing Components or Groups from a View*, on page 6-9.

Delete

See *Edit / Delete*, on page 6-27.

S@N.IT! update (host selected)

Updates the S@N.IT! software on the selected host. This feature requires **sanadmin** authority. It is available only for S@N.IT! Agents running version 6 of S@N.IT! (or higher) and for which the **AllowRemoteUpdate** configuration parameter has been set to true.

The new software must be present on the S@N.IT! Server machine: either the S@N.IT! CD–ROM is mounted or the new software has been copied in a directory.

The first pop–up window lets you:

- Select the location where the new software resides on the S@N.IT! Server
- Select the software to be transferred
- Enter the password of the root or Administrator user on the target agent.

When you click **OK** the software is copied from the Server to the Agent and a second pop–up window asks for confirmation of the installation. The upgrade is then performed on the agent and the S@N.IT! daemons are restarted on the agent.

The following figure summarizes the operations:

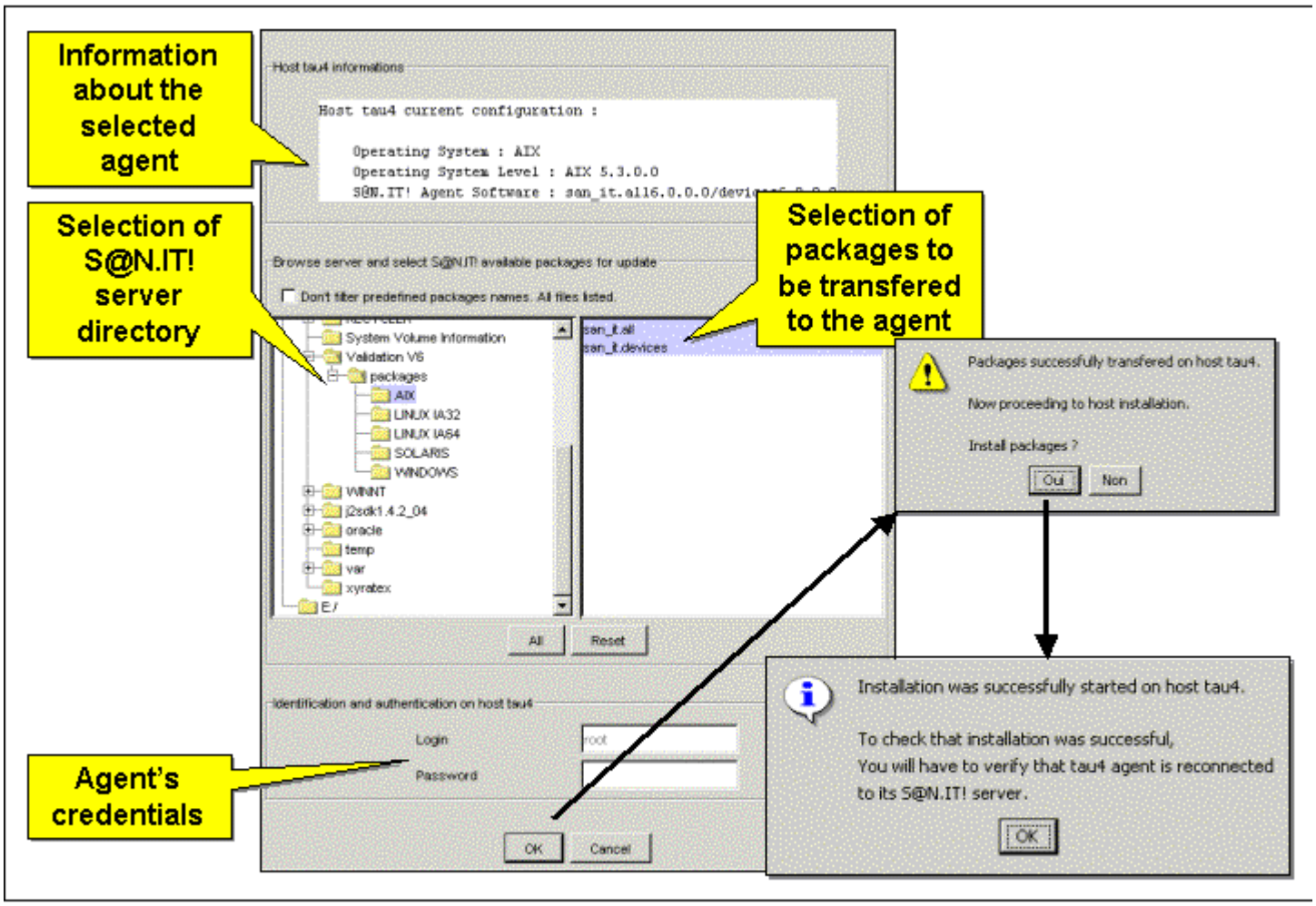

Figure 26. Updating S@N.IT! software on an Agent

Properties

See *Modifying a SAN component (Properties)*, on page 6-10.

Change Parent Component

See *Changing the owner of a port*, on page 6-11.

S@N.IT! GUI Information Frame

The information frame is displayed on the right part of the S@N.IT! GUI window. Three tabs are available: *Topology*, *Information* and *Changes*. Their contents depend on the object currently selected in the view frame.

Topology

Displays the topology of the current view. The user can choose to display:

- **Connections**
- **Paths** between hosts and storage devices.

Using the “Options” menu and the buttons on the left side, the user can:

- Modify the colours used for the background and the fonts.
- Choose to expand or contract labels.
- Hide or show icons and labels.
- Activate/de–activate the display of tool tips when moving the mouse above an object.
- Decide whether non–connected SAN portions are shown in different sub graphs.
- Decide whether topology or status modifications are shown immediately or after an explicit warning and an action from the user.

Acting on the graph itself, the user can move the objects (by drag–and–drops), expand or contract sets of components (host groups, libraries, fabric), select an object and access to contextual menus.

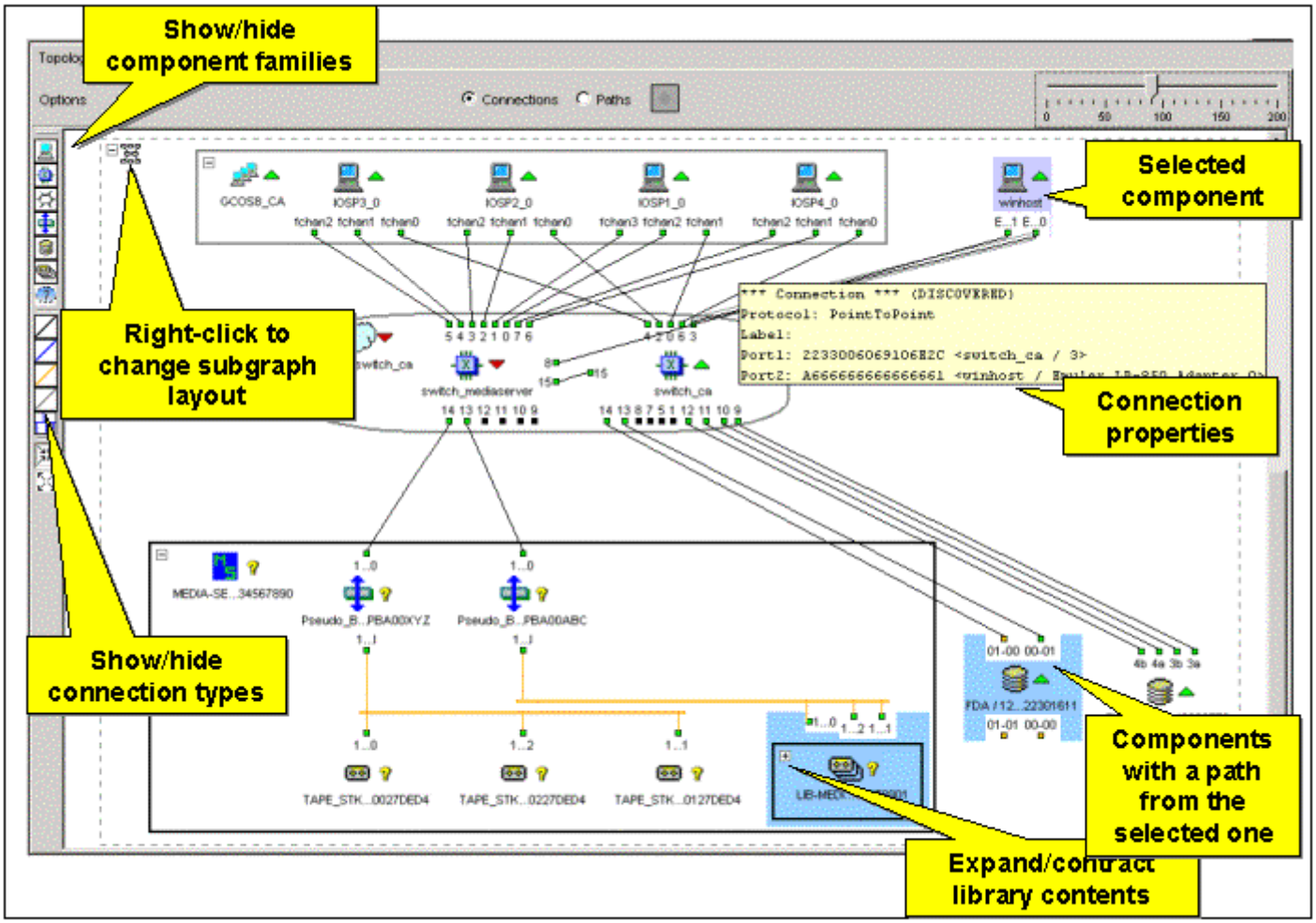

Figure 27. Connection properties display in Topology frame

Figure 28 shows the *Paths* view in the *Topology* tab.

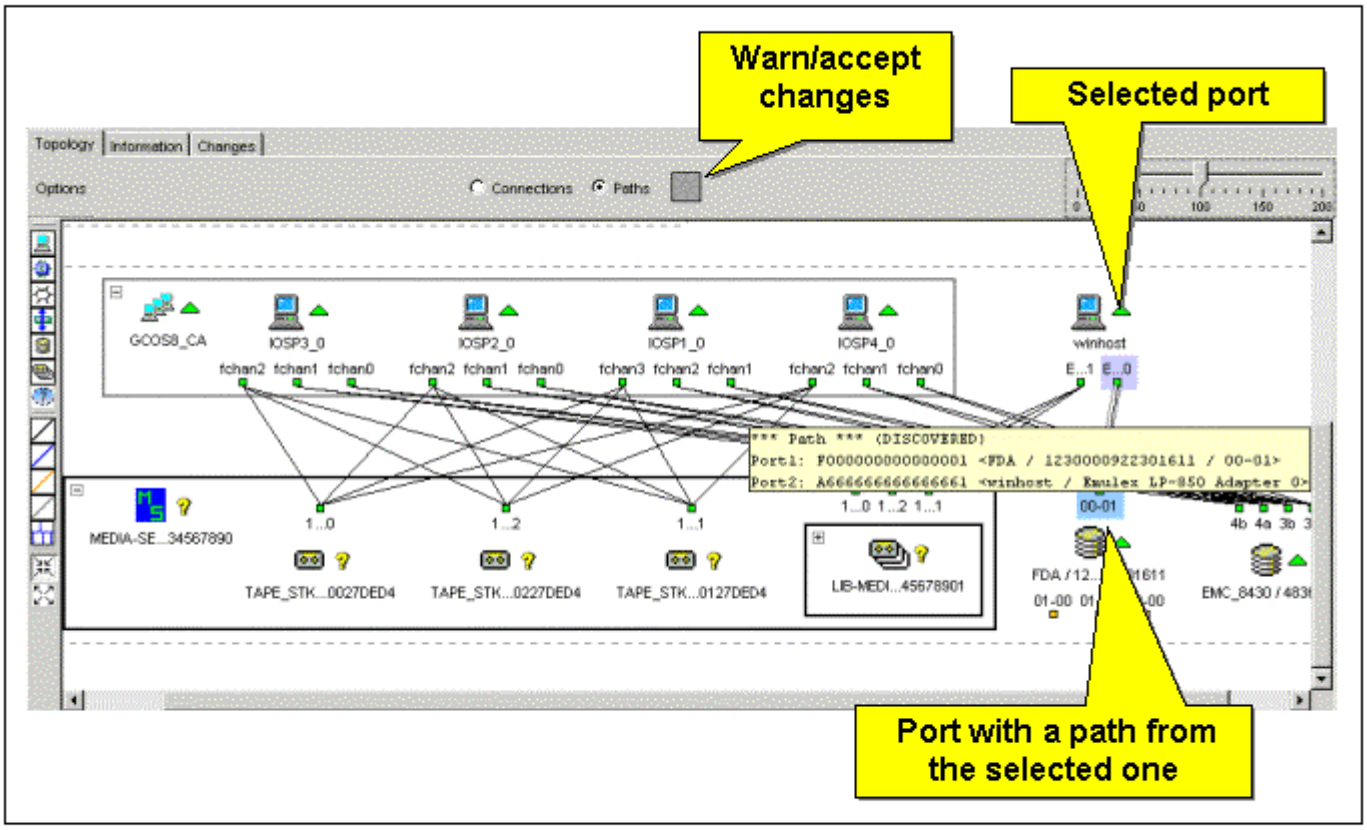

Figure 28. Path properties display in Topology frame

Information / Properties

Displays the properties of the currently selected component.

Information / Local Topology (SAN component selected)

Similar to the *Topology* tab but focused on the currently selected component: only the components that have a path or a connection to the current one are displayed.

Mouse usage

- A “Left-click” selects an object locally to the frame.
- A “Shift+Left-click” selects the object in the View management frame. The information tab is thus related to the newly selected component.

Information / Ports (SAN component selected)

Displays the fibre channel ports of the selected component.

Information / Monitor Log

Displays the monitoring status changes for the currently selected object. This information can be:

- Saved in a file (on the machine where the GUI has been launched), using the **Window/Save table to a file** menu.
- Erased from the S@N.IT! Server, using the **Window/Erase Log** menu.
- Reloaded from S@N.IT! Server, using the **Window/Refresh** menu.

Information / Subsystem LUNs

See *LUNs and LUN Groups Management*, on page 6-38.

Information / Host LUNs

See *LUNs and LUN Groups Management*, on page 6-38.

Information / LUN Access Log

See *LUNs and LUN Groups Management*, on page 6-38.

Information / File Systems (Host selected)

Displays two or three tables, depending on the Agent operating system:

- The first table describes the file systems of the host with their mount point, type, allocated and used size, current usage rate and threshold (maximum usage rate).

  The default value for this threshold is the value of the **FileSystemUseRateThreshold** configuration parameter of the S@N.IT! Server. It can be customized for each file system.

- The second table, describes the location of the file system currently selected in the first table.

- The third table, if any, describes the GCOS7 volumes (if the platform is a DPS 7000/XTA) contained in the selected file system.

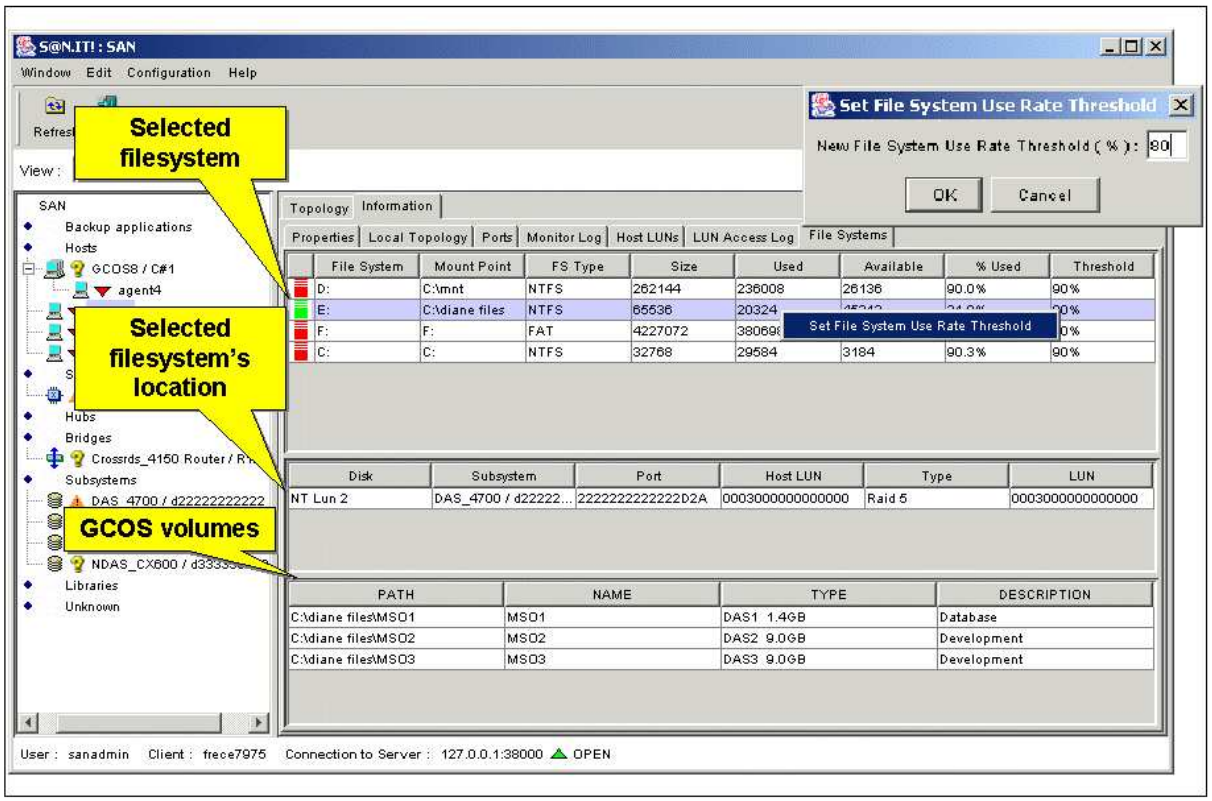

Figure 29. File systems display

Information / Change Logs

Displays the changes of the currently selected object, since its first discovery or since the currently selected reference was saved (see *Saving the Current State of the SAN Configuration as a Reference*, on page 6-14).

This information can be:

- Saved in a file (on the machine where the GUI has been launched), using the **Window**/**Save** table to a file menu.
- Erased from the S@N.IT! Server, using the **Window**/**Erase Log** menu.
- Reloaded from the S@N.IT! Server, using the **Window**/**Refresh** menu.

Information / Contents (view, group or complex component selected)

Displays all the applications, SAN components, port, paths and connections contained by the currently selected object, with their type, identifier and name, and the way they have been discovered.

This information can be:

- Saved in a file (on the machine where the GUI has been launched) using the **Window**/**Save** table to a file menu.
- Reloaded from the S@N.IT! Server using the **Window**/**Refresh** menu.

Information / Zoning (fabric selected)

Displays four tables describing the zoning of the fabric:

- Configurations with their state (ACTIVE or INACTIVE) and members
- Zones
- Aliases
- Members

This information can be:

- Saved in a file (on the machine where the GUI has been launched) using the **Window**/**Save table to a file** menu.
- Reloaded from the S@N.IT! Server using the **Window**/**Refresh** menu.

Changes / Reference (S@N.IT! Data Center only)

This tab is available only when a reference has been selected in the **Reference Selection** combo box. It describes the contents of the reference. See *Understanding SAN Changes*, on page 6-14 for details about using references.

This information can be:

- Saved in a file (on the machine where the GUI has been launched) using the **Window**/**Save table to a file** menu.
- Reloaded from the S@N.IT! Server using the **Window**/**Refresh** menu.

Changes / Comparison (S@N.IT! Data Center only)

This tab is available only when a reference has been selected in the **Reference Selection** combo box. It describes the differences between the selected reference and either the current state of the SAN or a newer reference. See *Understanding SAN Changes*, on page 6-14 for details about using references.

This information can be:

- Saved in a file (on the machine where the GUI has been launched) using the **Window**/**Save table to a file** menu.
- Reloaded from the S@N.IT! Server using the **Window**/**Refresh** menu.

Changes / Log (S@N.IT! Data Center only)

Displays all the changes, either since the first start of the S@N.IT! Server or since the currently selected reference was saved (see *Saving the Current State of the SAN Configuration as a Reference*, on page 6-14).

This information can be:

- Saved in a file (on the machine where the GUI has been launched) using the **Window**/**Save table to a file** menu.
- Erased from the S@N.IT! Server using the **Window**/**Erase Log** menu.
- Reloaded from the S@N.IT! Server using the **Window**/**Refresh** menu.

LUNs and LUN Groups Management

LUNs can be allocated and de–allocated to S@N.IT! Agents via:

- The **Information** / **Subsystem LUN** tab. This tab is available when a subsystem is selected. It allows to allocate and de–allocate LUNs to all S@N.IT! Agents that have a path to the subsystem.
- Or the **Information** / **Host LUNs** tab. This tab is available when a S@N.IT! Agent is selected. It allows to allocate and de–allocate LUNs to this S@N.IT! Agent in all subsystems connected to it by a path.

The *Host LUNs* tab shall be used in SAN configurations where subsystems have embedded LUN masking (DAS with Access Logix, Symmetrix with Volume Logix).

Two identifiers represent the LUNs:

- **LUN**: 16 hexadecimal digits SCSI3 – identifier – shortened to the 4 first digits when displayed by the GUI. It is the value seen by the operating systems SCSI layer.
- **Device LUN**: unique identifier within the subsystem. Its format depends on the subsystem model or vendor.

Information / Subsystem LUNs (Subsystem Selected)

This frame contains two parts.

- The upper part is a tree list:
  - The root of the tree is the **subsystem**
  - The second level are the subsystem **ports**
  - A third (and last) level may be defined by creating named groups of LUNs.
- The lower part displays information about the LUNs. Its contents depends on the element currently selected in the upper part :
  - If the subsystem is selected, all LUNs of the subsystem are displayed.
  - If a subsystem port is selected, all LUNs managed via this port are displayed. LUNs that are not currently reachable (or “owned”) by the port are also displayed if they have been mapped for this port on a S@N.IT! Agent.
  - If a LUN group is selected, all the LUNs that belong to the group are displayed.

  If a LUN is mapped on several agents, there is one line per mapping.

  For subsystems that support the “LUN masking” feature (Access Logix, Volume Logix), the list of LUNs is the sum of the visibility of all S@N.IT! Agents.

  Information about LUNs – used by the S@N.IT! Agents – is not automatically refreshed when a modification occurs on a server (for example when a Volume group is created on AIX). An explicit refresh (**Refresh** button or *Window / Refresh* menu) must be requested if the information needs to be updated.

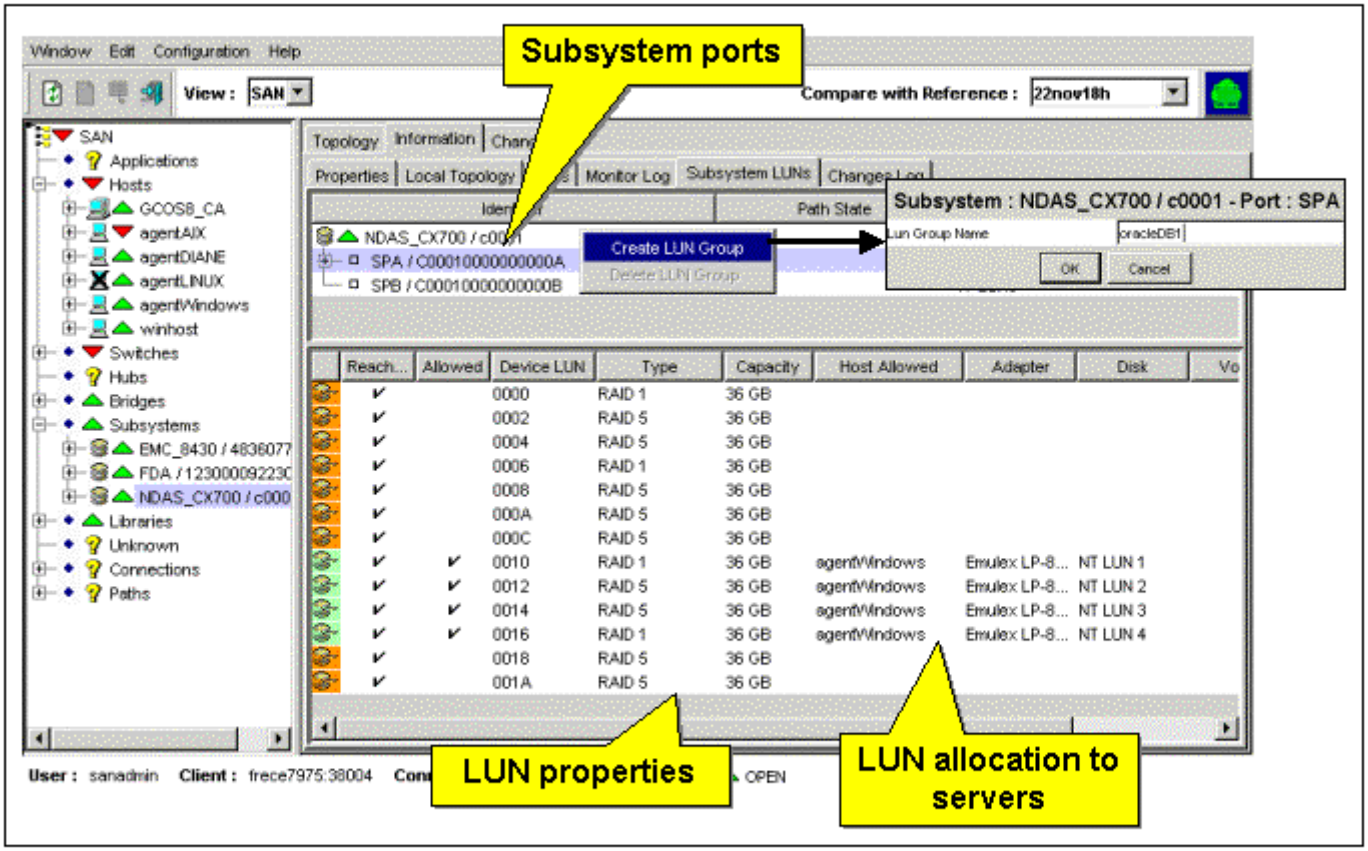

Figure 30. Subsystem LUNs Information

LUN Group Management

The **sanadmin** user can:

- **Create a LUN group** for a subsystem port (right–click in the upper frame while the subsystem port is selected).
- **Delete a LUN group** for a subsystem port (right–click in the upper frame while the LUN group is selected)**.**
- **Add LUNs to a LUN group** as follows:

1. Select the subsystem port in the upper frame (left–click).
2. Select LUNs in the lower frame (left–click).
3. Right–click and select “Add in LUN group” in the pop–up menu.
4. Select the target LUN group.

- **Remove LUNs from a LUN group** as follows:

1. Select the LUN group in the upper frame (left–click).
2. Select the LUNs in the lower frame (left–click).
3. Right–click and select “Remove from LUN group” in the pop–up menu.

LUN Mapping

LUN mapping operation is possible only when a subsystem port or a LUN group is selected in the upper part of the frame. See *Mapping LUNs*, on page 2-7 for more information.

To map LUNs proceed as follows:

1. Select the subsystem port or the LUN group in the upper frame (left–click).
2. Select LUNs in the lower frame (left–click).
3. Right–click and select "*Allow Access*" in the pop–up menu.
4. Select the host and the host adapter in the pop–up windows.

   If the subsystem is a DAS and if there are several paths between the host and the DAS subsystem, a checkbox ("*Configure all possible paths?*") enables to decide whether the LUN must be mapped for all paths.
5. A new pop–up window appears that provides information and recommendations about the execution of the command on the S@N.IT! Agent host.
6. The frame is updated.

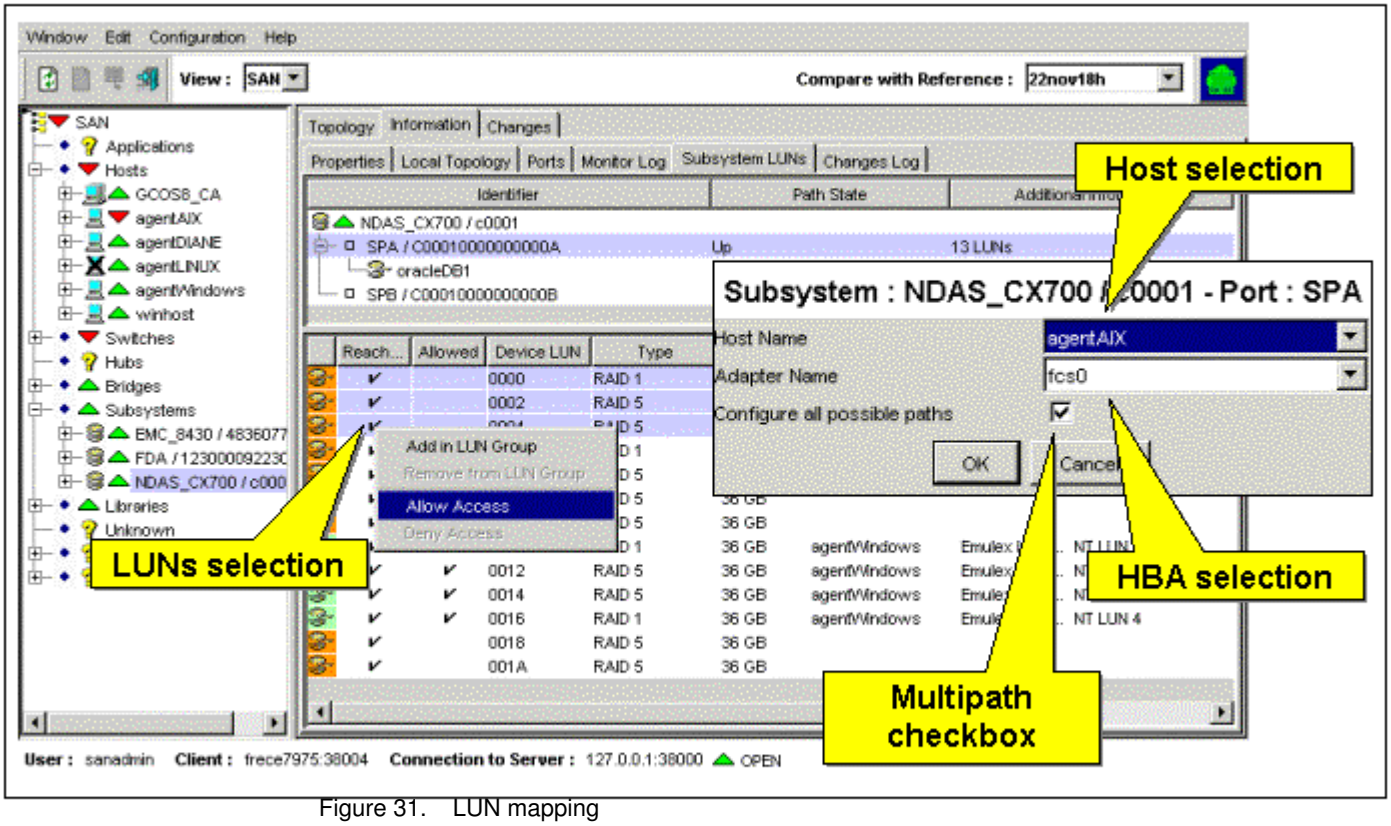

Figure 31. LUN mapping

LUN Unmapping

To unmap LUNs, proceed as follows:

1. Select a line in the upper frame (left–click).
2. Select LUNs in the lower frame (left–click).
3. Right–click and select "*Deny Access*" in the pop–up menu.
4. Confirm the action in the pop–up windows.

5. A new pop–up window appears that provides information and recommendations about the execution of the command on the S@N.IT! Agent host.
6. The frame is updated.

Information / Host LUNs (S@N.IT! Agent selected)

This frame contains two parts.

- The upper part is a tree list:
  - The root of the tree is the **host** (S@N.IT! Agent).
  - The second level nodes are the **subsystems** to which the host has a path. Subsystems with no path (temporary disconnection for example) are listed only if some of their LUNs have been allocated to the host.
  - The third level nodes are the subsystem **ports** to which the host has a path. Ports with no path (temporary disconnection for example) are listed only if some of their LUNs have been allocated to the host.
  - The leaves are set of LUNs: **Allowed LUNs**, and **Forbidden LUNs**.
- The lower part of the frame displays information about the LUNs. Its content depends on the element currently selected in the upper part:
  - If the host is selected, all the LUNs allocated to the host are displayed.
  - If a subsystem is selected, all the LUNs of the subsystem are displayed.
  - If a subsystem port is selected, all the LUNs managed via this port are displayed. LUNs that are not currently reachable (or “owned”) by the port are also displayed if they have been mapped for this port.
  - If the “*Allowed LUNs*” leaf is selected, only the LUNs allocated to the host for the current port are displayed.
  - If the “*Forbidden LUNs*” leaf is selected, only the LUNs not allocated to the host for the current port are displayed.

  For subsystems that support the “LUN masking” feature (Access Logix, Volume Logix), the LUNs that are masked to the current S@N.IT! Agent are not displayed.

  Information about LUN – used by the S@N.IT! Agent – is not automatically refreshed when a modification occurs on one server (for example when a Volume group is created on AIX). An explicit refresh (**Refresh** button or *Window / Refresh* menu) must be requested if the information needs to be updated.

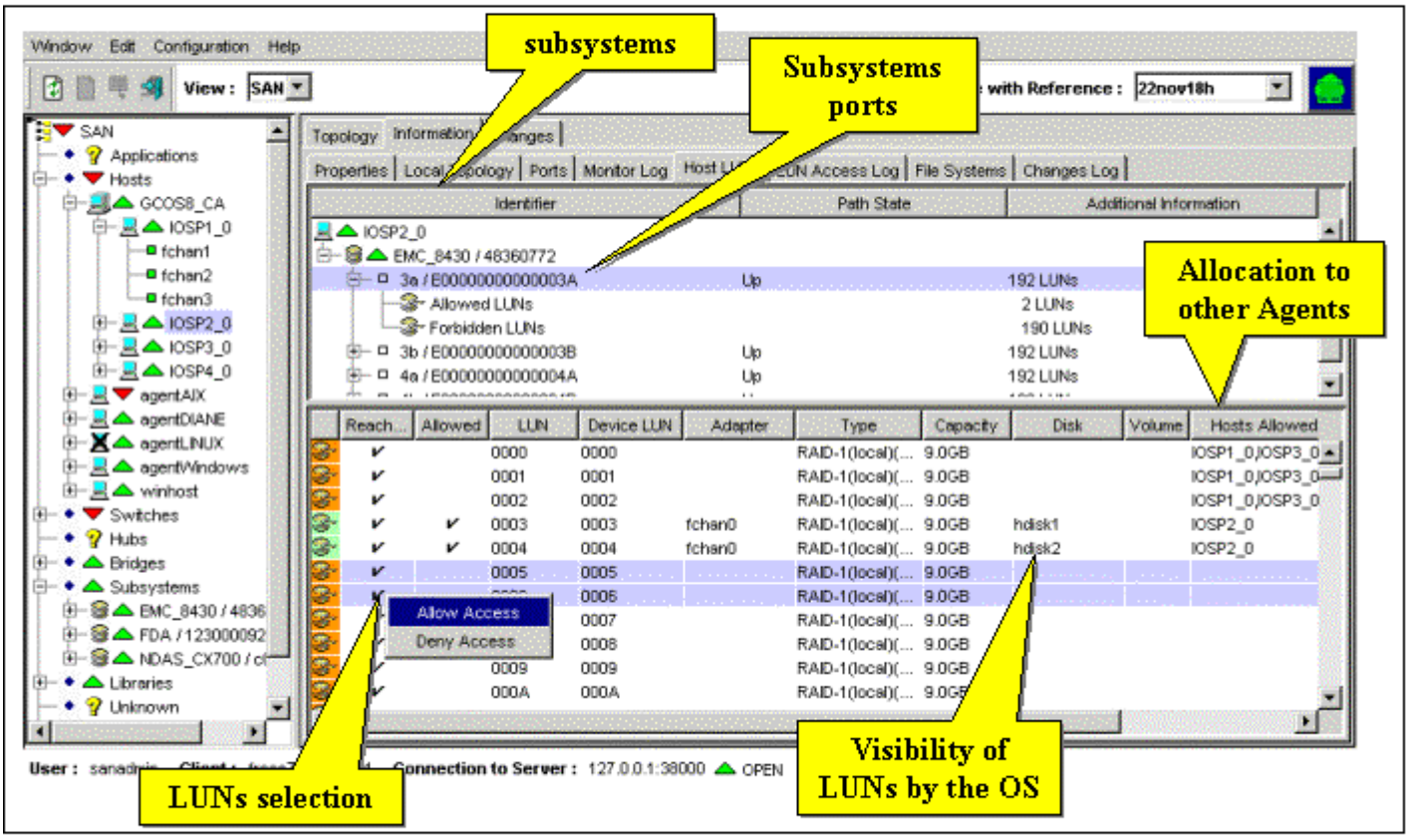

Figure 32. Host LUNs window

LUN Mapping

LUN mapping operation is possible only when a subsystem port or a leaf (*Allowed LUNs*, *Forbidden LUNs*) is selected in the upper part of the frame. See *Mapping LUNs*, on page 2-7 for more information.

To map LUNs perform these steps:

1. Select the subsystem port or its "*Forbidden LUNs*" leaf in the upper frame (left–click).
2. Select LUNs in the lower frame (left–click).
3. Right–click and select "*Allow Access*" in the pop–up menu.
4. Select the host adapter in the pop–up windows.
5. If the subsystem is a DAS and if there are several path between the host and the DAS subsystem, a checkbox ("*Configure all possible paths ?*") enables to decide whether the LUN must be mapped for all paths.
6. A new pop–up window appears that provides information and recommendations about the execution of the command on the S@N.IT! Agent host.
7. The frame is updated.

LUN Unmapping

To unmap LUNs perform these steps:

1. Select a line in the upper frame (left–click).
2. Select LUNs in the lower frame (left–click).
3. Right–click and select "*Deny Access*" in the pop–up menu.
4. Confirm the action in the pop–up windows.
5. A new pop–up window appears that provides information and recommendations about the execution of the command on the S@N.IT! Agent host.
6. The frame is updated.

Information / LUN Access Log (Host Selected)

Displays the LUN Access Control changes for the currently selected host.

This information can be:

- Saved in a file (on the machine where the GUI has been launched), using the **Window**/**Save table to a file** menu.
- Erased from the S@N.IT! Server, using the **Window**/**Erase Log** menu.
- Reloaded from the S@N.IT! Server, using the **Window**/**Refresh** menu.

Chapter 7. S@N.IT! Command Line Interface (CLI)

This chapter describes the following:

sanit Command

The **sanit** command can be used for two purposes:

- To start the S@N.IT! GUI from AIX, Linux or Solaris hosts.
- To launch and use the S@N.IT! Command Line Interface (CLI). Proceed as follows:
  - Open a session to the S@N.IT! server (**sanit –c open**)
  - Run commands (and retrieve the results) on this session (**sanit –s <session–id> –c <command>**).
    See *S@N.IT! CLI commands*, on page 7-3 for the description of these commands.
  - Close the session (**sanit –s <session–id> –c close**).

Command path:

AIX, Linux or Solaris path: **/usr/sanit/bin**

Windows path: **<%installation directory%>\bin**

Starting the S@N.IT! GUI on AIX, Linux and Solaris hosts

```
sanit [-p parameter=value [-p parameter=value ...]]
```

`parameter` can be one of:

```
HostName
ServerHost
ServerPort
TraceOnScreen
TraceOnFile
TraceDirectory
TraceSize
DebugLabels
```

A new parameter value will override the corresponding parameter of the **sanit.cfg** file of the host from which the command is launched. (See Chapter 5. *Configuration and Launching*).

If the DISPLAY environment variable is not set, the **ClientDisplay** parameter will be used to determine the X–Windows display into which the S@N.IT! GUI will appear.

Opening a command line interface session

```
sanit [-p parameter=value [-p parameter=value ...]]
[-s session_id] -c Open [user [password]]
```

`parameter` can be one of:

```
HostName
ServerHost
ServerPort
TraceOnScreen
TraceOnFile
TraceDirectory
TraceSize
```

```
DebugLabels
ClientTimeout
```

A new parameter value will override the corresponding parameter of the **sanit.cfg** file of the host where the command is launched. (See Chapter 5. *Configuration and Launching*).

Note: There must be no space around the = (equals) sign.

`session_id` user selected session identifier (must be a free TCP/IP port on the local platform). If this parameter is not provided, S@N.IT! will generate one.

`user` name of the S@N.IT! user to be used for that session (**common** or **sanadmin**).

`password` password of the S@N.IT! user to be used for that session.

If `user` or `password` parameters are not specified, the command interactively asks for them.

For the **sanadmin** user, there can be only one S@N.IT! CLI or GUI session at the same time for the same S@N.IT! server. There is no limitation for the **common** user, but this user is only allowed to display information.

The command output contains a line that indicates the S@N.IT! CLI session identifier. This value must be stored since it will be required to launch further S@N.IT! CLI commands and to stop the S@N.IT! CLI session. It is local to the current host and thus can only be used locally.

Examples:

```
# sanit -c open sanadmin admin
S@N.IT!: CLI session [52089].

# sanit -c open common
S@N.IT! common's password:
S@N.IT!: CLI session [52097].
```

Stopping a command line interface session

```
sanit -s session_id -c Close
```

`session_id` identifier of the S@N.IT! CLI session opened by the `sanit -c Open` command.

Displaying the current S@N.IT! CLI sessions

```
sanit -c Ls
```

This command displays the identifier of the S@N.IT! CLI session currently opened by the local operating system's user. For example:

```
# sanit -c ls
52097
52089
```

Running a S@N.IT! CLI command

```
sanit -s session_id -c <command>
```

`session_id` identifier of the S@N.IT! CLI session opened by the `sanit -c Open` command.

`<command>` Command to be executed, with its parameters.

The general syntax for the `<command>` parameter is:

```
Command_name [parameters]
```

The `command_name` parameter is not case sensitive but the `parameters` are case sensitive.

The **sanit** command exits with code set to 0 in case of success, and greater than 0 in case of failure.

S@N.IT! CLI commands

? or help

Displays the S@N.IT! CLI command usage.

Activate hostname

Activates the LUN Access Control mechanism on the given host. This command is not allowed if the S@N.IT! CLI session has been opened for user **common**.

The output of the command contains a diagnostic of the execution on the S@N.IT! Agent and recommendations for subsequent operations to be performed.

AddLunGroupContent –i serialnb –n LunSetName –l <lun_id> [lun_id*n*]

Adds LUNs to the set of LUNs specified by the disk array serial number (**–i** parameter), the name of the set of LUNs (**–n** parameter). The LUNs are specified by their **lun_id** (**–l** parameter). Each **lun_id** is a number, separated by a space. The **LunSetName** syntax depends on the subsystem model (see Chapter 11. *Supported SAN Components*).

This command is not allowed if the S@N.IT! CLI session has been opened for **common** user.

- Example:

```
# sanit –s 123 –c addlungroupcontent –i 0000000927909999 –l 0 3
```

AddLunGroupPath –i serialnb –n LunSetName –w [wwn_id wwn_n1 ...]

Adds the world wide name(s) of a HBA to a set of LUNs specified by the disk array serial number (**–i** parameter) and the name of the set of LUNs (**–n** parameter). Each wwn is separated by a space. The **LunSetName** syntax depends on the subsystem model (see Chapter 11. *Supported SAN Components*).

This command is not allowed if the S@N.IT! CLI session has been opened for **common** user.

- Example:

```
# sanit –s 123 –c addlungrouppath –i 0000000927909999 –n AX:Server_c
–w 10000000CC99999 10000000CC99900
```

AllowAccess hostname subsystem_model subsystem_physical_id –m adapterWWN portWWN [–m adapterWWN portWWN ...] –l LUN_ID [LUN_ID ...] [–g LUN_GROUP [LUN_GROUP ...]]

Maps LUNs of a Disk subsystem on the given host. This command is not allowed if the S@N.IT! CLI session has been opened for user **common**.

`hostname` — name of the host where the mapping operation must be performed.

`subsystem_model` — model of the Disk subsystem that contains the LUNs.

`subsystem_physical_id` — physical identifier of the Disk subsystem that contains the LUNs.

`–m` flags — are followed by a description of a path for which the mappings are to be performed (for the whole list of LUNs):

- `adapterWWN` WWN of the host HBA
- `portWWN` WWN of the Disk subsystem port

`LUN_ID` — string of 16 hexadecimal digits (SCSI 3 address) that identifies a LUN. For DAS subsystem it must be the Subsystem LUN ID, and not the Host LUN ID.

`LUN_GROUP` name of a group of LUNs. Groups of LUNs can be defined through the S@N.IT! GUI; see *LUN Group Management*, on page 6-39.

The execution of the command launches the **UserPreMappingCommand** and **UserPostMappingCommand** commands on the S@N.IT! Agent if they have been configured in its **sanit.cfg** configuration file.

The output of the command contains a diagnostic of the execution on the S@N.IT! Agent and recommendations for subsequent operations to be performed.

- Example:

```
#sanit –s 49584 –c allowaccess csmgt14 DAS_3500 9646169 –m 10000000C920C1EF
200000601627D8FC –l 000400000000000 0003000000000000
```

CreateComponent [–H | –f family –i serialnb [–n LunSetName] [–N] [–a<property>=<value> ...]]

Creates a new SAN component and set its properties. The **serialnb** must be unique for the S@N.IT! Server. The values allowed for the **family** parameter and the **property** names corresponding to the selected family, can be listed using the **–H** option.

If the **–N** option is present, the command doesn't check that the property names are allowed for the component family.

This command is not allowed if the S@N.IT! CLI session has been opened for **common** user.

- Example:

```
#sanit –s 123 –c CreateComponent –f Library –i 45678
–aCompoModel=”STK_L700” –aCompoIpAddress
```

CreateLunGroup –i serialnb –n LunSetName

Creates a set of LUNs, whose name is defined by **–n** parameter, for the disk array specified by its serial number (**–i** parameter). The **LunSetName** syntax depends on the subsystem model (see Chapter 11. *Supported SAN Components*).

This command is not allowed if the S@N.IT! CLI session has been opened for **common** user.

- Example:

```
# sanit –s 123 –c createlungroup –i 0000000927909999 –n AX:Server_c
```

CreatePort [–H|–w wwn [–i serialnb] [–a< property >=<value> ...]]

Creates a Fibre Channel port and set its initial properties. The **wwn** parameter (world wide name) must be unique for the S@N.IT! Server. The **serialnb** parameter refers to the SAN component that contains the new port. The values allowed for the **property** names corresponding to the selected family can be listed using the **–H** option.

This command is not allowed if the S@N.IT! CLI session has been opened for **common** user.

- Examples:

```
# sanit –s 123 –c CreateComponent –f Host –n manhost –i 789
# sanit –s 123 –c createport –w 100000092183C –i789 –aPortName=”adapter 1”
```

Deactivate hostname

De–activates the LUN Access Control mechanism on the given host. This command is not allowed if the S@N.IT! CLI session has been opened for user **common**.

The output of the command contains a diagnostic of the execution on the S@N.IT! Agent and recommendations for subsequent operations to be performed.

DeleteComponent [–H | –f family –i serialnb [–n LunSetName] [–a< property >=<value> ...]]

Deletes one SAN component from the S@N.IT! Server data. This component is specified by its **family** and **serialnb** properties. The name and some other properties can also be provided as complementary filtering criteria. The command will fail if several components match the specifications. The values allowed for the **family** parameter can be listed using the **–H** option.

This command is not allowed if the S@N.IT! CLI session has been opened for common user.

Note that the component will reappear if it is automatically discovered by S@N.IT!

- Example:

```
#sanit –s 123 –c DeleteComponent –f Library –i 45678
```

DeletePort –w wwn

Deletes a fibre channel port, specified by its wwn (world wide name). This command is not allowed if the S@N.IT! CLI session has been opened for **common** user.

Note that the port will reappear if it is automatically discovered by S@N.IT!

- Example:

```
# sanit –s 123 –c DeletePort –w 100000092183C
```

DenyAccess hostname subsys_model subsys_physical_id –m adapterWWN portWWN [–m adapterWWN portWWN ...] –l LUN_ID [LUN_ID ...] [–g LUN_GROUP] [LUN_GROUP ...]

Unmaps LUNs of a Disk subsystem on the given host. This command is not allowed if the S@N.IT! CLI session has been opened for user **common**.

`hostname` — name of the host where the unmapping operation must be performed.

`subsystem_model` — model of the Disk subsystem that contains the LUNs.

`subsystem_physical_id` — physical identifier of the Disk subsystem that contains the LUNs.

`-m` flags — are followed by a description of a path for which the unmappings are to be performed (for the whole list of LUNs):

- `adapterWWN` — WWN of the host HBA.
- `portWWN` — WWN of the Disk subsystem port.

`LUN_ID` — string of 16 hexadecimal digits (SCSI 3 address) that identifies a LUN. For DAS subsystem it must be the Subsystem LUN ID, and not the Host LUN ID.

`LUN_GROUP` — name of a group of LUN's. Groups of LUN's can be defined through the S@N.IT! GUI, see *LUN Group Management*, on page 6-39.

The execution of the command launches the **UserPreUnmappingCommand** and **UserPostUnmappingCommand** commands on the S@N.IT! Agent if they have been configured in its **sanit.cfg** configuration file.

The output of the command contains a diagnostic of the execution on the S@N.IT! Agent and recommendations for subsequent operations to be performed.

- Example:

```
#sanit –s 49584 –c denyaccess csmgt14 DAS_3500 9646169 –m 10000000C920C1EF
200000601627D8FC –l 000400000000000 0003000000000000
```

DisplayComponent [–H] [–P] [–w portWWN] [–i serialnb]

Displays information about SAN Components.

If **–w** option is specified, it displays SAN component that owns the port whose World Wide Name is equal to *portWWN.*

If **–i** option is specified, it displays the SAN component whose serial number is equal to **serialnb**.

If neither **–w** nor **–i** option are specified, all discovered SAN components are displayed.

If the **–H** option is specified, the first line of the command output is a header.

The command displays one line per SAN component. If **–P** option is specified, one line per port attached to this component follows.

- SAN component information has the following general format:

```
Family|Name|Model|Physical ID|IP Address|Monitoring Status|
Parent physical Id|HW/FW level
```

Note: the HW/FW level information is not present for hosts

Family–specific information is eventually added:

– for hosts:

```
HOST|host name |host model|host physicalID|IP Address|Monitoring Status|
Parent physical Id |Lun access control state |operating system|
operating system level|S@N.IT! agent interface version
```

– for switches:

```
SWITCH|switch name|switch model|switch WWN| IP Address |Monitoring Status |
Parent physical Id (Fabric  WWN )|FW level|switch Role|switch Domain ID
```

- Each port descriptor has the following format:

```
PORT|portWWN|portName|portID|portLocation
```

- Examples:

```
#sanit -s 123 -c DisplayComponent -H

Family|Name|Model|Physical ID|IP Address|Status|Parent Container
HOSTGROUP||GCOS8|C#1||NOT_MONITORED||
HOST|agent3|Windows 2000 Server
(SP2)|6CFCCFC0-0A74-11D6-9365-806D6172696E|172.16.101.3|FAULTY||ACTIVE|Wind
ows 2000|Windows 2000|2.0
HOST|agent1|Windows 2000 Server
(SP2)|6CFCCFC0-0A74-11D6-9365-806D6172696F|172.16.101.1|FAULTY||ACTIVE|Wind
ows 2000|Windows 2000|2.0
HOST|agent4|Bull ESCALA T
(e8ED)|00000000E804|172.16.101.4|FAULTY|C#1|ACTIVE|AIX|AIX 4.3.3|2.0
HOST|agent2|Bull ESCALA T (e8ED)
|00000000E802|172.16.101.2|FAULTY||ACTIVE|AIX|AIX 5.2|2.0
SWITCH|switchX|BROCADE|1000007069100222|172.16.101.4|UNKNOWN||fw level|1|2
BRIDGE||Crossrds_4150 Router|R1A00ELW||NORMAL||fw level
SUBSYSTEM||NDAS_CX600|d33333333333|172.16.101.5|NORMAL||fw level
SUBSYSTEM||DAS_4700|d22222222222|172.16.101.6|UNKNOWN||fw level
SUBSYSTEM||DAS_5700|d11111111111||FAULTY||fw level
SUBSYSTEM||NDAS_CX400|d44444444444|172.16.101.7|NORMAL||fw level
LIBRARY||STK_L80|C#B111700363|172.16.101.8|NORMAL||
MEDIUMCHANGER||MCHG_STK_L80|B111700363||NOT_MONITORED|C#B111700363|fw level
TAPEDRIVE||TAPE_STK_9840|B16355365||NOT_MONITORED|C#B111700363|fw level
TAPEDRIVE||TAPE_QUANTUM_SuperDLT1|B16338286||NOT_MONITORED|C#B111700363|fw
level
UNKNOWN|||10000000CC999999||NOT_MONITORED||
```

Displaystatus [–R requestID]

Displays the status of the S@N.IT! CLI session, or if the **–R** option is specified, displays the status of the **ForceDiscovery** request identified by the `RequestID` parameter.

- Examples:

```
#sanit –s 123 –c displaystatus
S@N.IT! CLI session [123]:
User=sanadmin Hostname=myhost Server=myserver (Port=38000)
Connection Status=1 Timeout=300

#sanit –s 123 –c displaystatus –R 1
S@N.IT! CLI session [123]: RequestID [1]: [COMPLETED].

#sanit –s 123 –c displaystatus –R 2
S@N.IT! CLI session [123]: RequestID [2]: [INPROGRESS].
```

ForceDiscovery [–N] [–f fabricWWN] [–h hostname] [–i switchPhysicalID]

Forces the S@N.IT! discovery process to be restarted. If no SAN component is specified (no **–f ,–i** and **–h** option provided), this restarts the exploration of the range of IP addresses specified in the configuration of the S@N.IT! server, and refreshes the information provided by the SAN components (mainly S@N.IT! Agents and switches) already discovered.

Else the specified component(s) are polled for new topology information.

If **–f** option is specified, all switches of the fabric specified by the **fabricWWN** parameter are polled for SAN topology information.

If **–h** option is specified, the S@N.IT! Agent specified by the **hostname** parameter is polled for SAN topology information.

If **–i** option is specified, the switch specified by the **switchPhysicalID** parameter is polled for SAN topology information.

The command may be launched in synchronous (default) or asynchronous mode (**–N** option). When launched in asynchronous mode, a request identifier is displayed on output. The status of the request can then be checked using the **DisplayStatus** S@N.IT! CLI command (see above).

- Examples:

```
#sanit –s 123 –c forcediscovery –N
S@N.IT! CLI session [123]: RequestID [2].

#sanit –s 123 –c forcediscovery –h agent2
S@N.IT! CLI session [123]: Discovery done.
```

GenerateReport [–t template] [–f format] –o outputfile

Generates a report in the local file whose path is specified in the **outputfile** parameter, using the template specified in the **template** parameter. The default value for template is **SanFullReport**. The default format is HTML.

HostLUNs [–H] [–R] hostname

Displays the list of LUN's that are reachable from the host given as parameter.

If the **–H** option is present, the first line of the command output is a header.

The **–R** option forces a Refresh.

The command displays one line per path between the host and the subsystem port that owns the LUN. The format is as follows:

```
Adapter|adapter WWN|Subsystem Model|Subsystem Id|Port Name|
Port WWN|LUN Id|Type|Capacity|Disk|Volume|Map Flag|
Reachable flag|Path Flag|Disk State
```

where:

`Adapter`	HBA of the host that allows the LUN to be reached.
`Subsystem`	Disk subsystem that contains the LUN.

`Port`	Disk subsystem port that owns the LUN.
`LUN Id`	format: `Host LUN id: Subsystem LUN id` where both `id` are 16 hexadecimal digits strings .
`Type` / `Capacity` / `Disk`	AIX hdisk or Windows LUN corresponding to the LUN if any.
`Volume`	name of the AIX volume group or Windows drive corresponding to the LUN if any.
`Map flag`	set to 1 if the LUN is mapped on the host, 0 otherwise.
`Reachable flag`	set to 1 if the LUN is currently owned by the subsystem port, 0 otherwise.
`Path Flag`	set to 1 if there is a path between the host fibre channel adapter and the subsystem port.
`Disk State`	`A` = Available, `D` = Define.

- Example:

```
#sanit –s 1235 –c hostluns tau4
```

LsDeviceSubsystem [–H] –i serialnb

Displays the properties of a disk array specified by its serial number (**–i** parameter).

When the **–H** option is present, the first line of the command output is a header.

The command displays one line with the following format:

```
DEVICESUBSYSTEM|DISKARRAYNAME|VENDORID|PRODUCTID|CAPACITY|FWREVISION|WWN|
CROSSCALLMODE|ACCESSCONTROLMODE
```

where:

`DEVICESUBSYSTEM`	header of the command
`DISKARRAYNAME`	Name of the disk array
`VENDORID`	vendor id of the disk array
`PRODUCTID`	Product Id of the disk array
`CAPACITY`	Total capacity of the Disk array
`FWREVISION`	Product Firmware revision
`WWN`	World wide name of the disk array
`CROSSCALLMODE`	See specific component
`ACCESSCONTROLMODE`	state of the access control mode

- Example:

```
# sanit –s 123 –c LsDevicesSubsystem –i 0000000927909999
DEVICESUBSYSTEM|StorewayFDA2300|NEC|S2300 Disk Array|
Y123|200000004C7F0445|on|on
```

LsFabric [–H]

Displays the fabrics discovered in the SAN configuration.

If the **–H** option is present, the first line of the command output is a header.

The command displays one line per fabric with the following format:

```
FABRIC|fabricWWN
```

where:

`fabricWWN`	World Wide Name of the fabric

- Example:

```
#sanit -s 123 -c LsFabric -H
FABRIC|Fabric WWN
FABRIC|1000007069100222
FABRIC|1000006069100212
```

LsFabricConnectedPorts [–H] –f fabricWWN

Displays the ports (fibre channel attachments) connected to those of the fabric whose WWN (World Wide Name) is given as parameter (**–f** option).

If the **–H** option is present, the first line of the command output is a header.

The command displays one line per port with the following format:

```
FABRICCONNECTEDPORT|portWWN|portName|portID|portLocation|componentFamily|co
mponentName|componentModel|componentPhysicalID
```

where:

`portWWN`	WWN of the connected port.
`portName`	name of the connected port.
`portID`	fibre channel address of the connected port.
`portLocation`	location of the connected port.
`componentFamily`	type of the component to which the connected port belongs.
`componentName`	name of the component to which the connected port belongs.
`componentModel`	model of the component to which the connected port belongs.
`componentPhysicalID`	serial number of the component to which the connected port belongs.

- Example:

```
# sanit -s 52472 -c lsfabricconnectedports -H -f 1000006069100212
FABRICCONNECTEDPORT|Port WWN|Port Name|Port ID|Location Code|Family|Logical
Name|Model|Physical ID
FABRICCONNECTEDPORT|10000000C9211833|fchan0|021A00|04-06|HOST|csmgt08|Bull
ESCALA T (e8ED)           |00000000E800
FABRICCONNECTEDPORT|500601600601A09E|SPA|021C00||SUBSYSTEM|MY_DAS|DAS_5300|
f10002600158
```

LsFabricConnections [–H] [–E] [–I] –f fabricWWN

Displays the list of connections from/to the fabric specified by its WorldWideName (**–f**). Each connection is a pair {fabricPortWWN, connectedPortWWN}.

If the **–E** option is present, only the connections to ports that do not belong to the fabric are displayed.

If the **–I** option is present, only the connections between ports of the fabric are displayed.

If neither **–E**, not **–I** option is present, all connections are displayed.

If the **–H** option is present, the first line of the command output is a header.

The command output displays one line per connection with the following format:

```
FABRICCONNECTION|fabric PortWWN|connected PortWWN
```

- Example:

```
# sanit -s 52472 -c lsfabricconnections -H -f 1000006069100212
FABRICCONNECTION|Fabric Port WWN|Connected Port WWN
FABRICCONNECTION|2001006069100212|10000000C920B29A
FABRICCONNECTION|200A006069100212|10000000C9211833
FABRICCONNECTION|2002006069100212|10000000C9212B5D
FABRICCONNECTION|200C006069100212|500601600601A09E
FABRICCONNECTION|2004006069100212|10000000C9216CBD
FABRICCONNECTION|2005006069100212|10000000C920C1EF
FABRICCONNECTION|2006006069100212|500104F00045CD06
FABRICCONNECTION|2008006069100212|10000000C920A7E8
FABRICCONNECTION|2009006069100212|2100002037184868
```

LsFabricPorts [–H] [–C] –f fabricWWN

Displays the ports (fibre channel attachments) of the fabric whose WWN (World Wide Name) is given as (**–f**) parameter.

If the **–H** option is present, the first line of the command output is a header.

If the **–C** option is provided, only the fabric ports that are connected to an external port are displayed

The command output displays one line per port with the following format:

```
FABRICPORT|portWWN|portName|portID|portLocation|SWITCH|switchName|switchMod
el|switchPhysicalID|switchDomainID
```

where:

`portWWN`	WWN of the switch port.
`portName`	number of the switch port.
`portID`	fibre channel address of the switch port.
`portLocation`	location of the port within the switch.
`switchName`	IP name of the switch to which the port belongs.
`switchModel`	model of the switch to which the port belongs.
`switchPhysicalID`	identifier of the switch to which the port belongs.
`switchDomainID`	fibre channel identifier of the switch to which the port belongs.

- Example:

```
# sanit -s 52472 -c lsfabricports -H -C -f 1000006069100212
FABRICPORT|Port WWN|Port Name|Port ID|Location Code|Family|Logical
Name|Model|Physical ID|Domain ID
FABRICPORT|2001006069100212|1|021100|module 0/port 1|
SWITCH|switch5|BROCADE|1000006069100212|2
FABRICPORT|200A006069100212|10|021A00|module 0/port 10|
SWITCH|switch5|BROCADE|1000006069100212|2
FABRICPORT|2002006069100212|2|021201|module 0/port 2|
SWITCH|switch5|BROCADE|1000006069100212|2
FABRICPORT|200C006069100212|12|021C00|module 0/port 12|
SWITCH|switch5|BROCADE|1000006069100212|2
FABRICPORT|2004006069100212|4|021400|module 0/port 4|
SWITCH|switch5|BROCADE|1000006069100212|2
FABRICPORT|2005006069100212|5|021500|module 0/port 5|
SWITCH|switch5|BROCADE|1000006069100212|2
FABRICPORT|2006006069100212|6|0216E1|module 0/port 6|
SWITCH|switch5|BROCADE|1000006069100212|2
FABRICPORT|2008006069100212|8|021801|module 0/port 8|
SWITCH|switch5|BROCADE|1000006069100212|2
FABRICPORT|2009006069100212|9|0219EF|module 0/port 9|
SWITCH|switch5|BROCADE|1000006069100212|2
```

LsHost [–H]

Displays the list of hosts with S@N.IT! Agents.

If the **–H** option is present, the first line of the command output is a header.

The command output displays one line per S@N.IT! Agent with the following format:

```
HOST|host name|model|physical identifier|IP address|monitoring status|LUN
access control state|operating system|SAN Agent version
```

- Example:

```
#sanit –s 49584 –c LsHost

Host|csmgt14|Bull ESCALA T (e8ED)|00400035E800|172.16.101.14|
NOT_MONITORED|INACTIVE|AIX|2.0
Host|csmgt10|Bull ESCALA T (e8ED) |00000080E800|172.16.101.10|
NOT_MONITORED|ACTIVE|AIX|1.0
```

LsLunGroup –i serialnb

Displays the names of the set of LUNs for the disk array specified by its serial number (**–i** parameter).

If the **–H** option is present, the first line of the command output is a header.

The command output displays one line per set of LUNs with the following format:

```
LUNGROUP|LUN set Name
```

where:

`LUNGROUP`	header of the results
`<LUN set name>`	name of the set of LUNs (see Chapter 11. *Supported SAN Components*).

- Example:

```
# sanit –s 123 –c lslungroup –i 0000000927909999
LUNGROUP|LX:poupou
LUNGROUP|LX:csmgt21
LUNGROUP|LX:tigre1_evb
LUNGROUP|LX:cli_test
LUNGROUP|LX:poupou_ev_qla
LUNGROUP|AX:BF_Mediaserveur1
LUNGROUP|AX:BF_MediaServeur2
LUNGROUP|LX:12345678
```

LsLunGroupContent [–H| –i serialnb –n LunSetName

Displays the LUNs of a set of LUNs specified by its name (**–n** parameter) and by the disk array serial number (**–i** parameter). The **LunSetName** syntax depends on the subsystem model (see Chapter 11. *Supported SAN Components*).

If the **–H** option is present, the first line of the command output is a header.

The command output displays one line per set of LUNs with the following format:

```
LUNGROUPCONTENT|LOGICALLUN:PHYSICALLUN
```

where:

`LUNGROUPCONTENT`	header of the command
`LOGICALLUN`	number of the LUN (number seen from the hosts).
`PHYSICALLUN`	physical LUN (the physical format depends on the model.)

- Example:

```
# sanit –s 123 –c lslungroupcontent –i 0000000927909999 –n AX:Server_c
LUNGROUPCONTENT|0000000000000000:000e
LUNGROUPCONTENT|0000000000000001:000f
LUNGROUPCONTENT|0000000000000002:0010
LUNGROUPCONTENT|0000000000000003:0011
```

LsLunGroupPath [–H| –i serialnb –n LunSetName

Displays the world wide names of the HBAs allowed to access the set of LUNS specified by its name (**–n** paramter) for the disk array specified by its serial number (**–i** parameter).

If the **–H** option is present, the first line of the command output is a header.

The command output displays one line per set of LUNs with the following format:

```
LUNGROUPPATH| PATH
```

where:

`LUNGROUPPATH`	header of the result
`PATH`	world wide name of the HBA accessing to this set of LUNS.

- Example:

```
# sanit –s 123 –c lslungrouppath –i 0000000927909999 –n LX:tigre1_evb
LUNGROUPPATH|210000E08B0A9715
LUNGROUPPATH|210000E08B08ECE1
LUNGROUPPATH|10000000C928C690
LUNGROUPPATH|10000000C9216F01
```

LsLUNs [–H] –w portWWN

Displays the list of LUNs behind a subsystem port represented by its World Wide Name (**–w**).

If the **–H** option is present, the first line of the command output is a header.

The command output displays one line per LUN with the following format:

```
LU|portWWN|LUN|LU's SCSI 3 ID|LU Type|LU Capacity|LU reachability
```

where:

`portWWN`	WWN of the subsystem port.
`LUN`	decimal value representing the logical unit number within the subsystem.
`LU's SCSI 3 ID`	hexadecimal digit strings representing the subsystem LUN.
`LU Type`	type of the logical unit (individual disk, RAID1, ...).
`LU Capacity`	size of the logical unit.
`LU reachability`	`1` indicates that the logical unit is reachable through the subsystem port. `0` indicates that the LUN is not reachable through the subsystem port.

- Example:

```
# sanit –s 52472 –c lsluns –w 500601600601A09E
LU|500601600601A09E|4|0004000000000000|RAID+5|104.9MB|1
LU|500601600601A09E|6|0006000000000000|RAID+5|104.9MB|1
LU|500601600601A09E|12|000C000000000000|RAID+5|104.9MB|1
LU|500601600601A09E|14|000E000000000000|RAID+5|104.9MB|1
LU|500601600601A09E|8|0008000000000000|RAID+5|104.9MB|1
LU|500601600601A09E|0|0000000000000000|RAID+5|104.9MB|1
LU|500601600601A09E|11|000B000000000000|RAID+5|104.9MB|1
LU|500601600601A09E|2|0002000000000000|RAID+5|104.9MB|1
LU|500601600601A09E|13|000D000000000000|RAID+5|104.9MB|1
LU|500601600601A09E|15|000F000000000000|RAID 5|104.9MB|1
LU|500601600601A09E|10|000A000000000000|RAID+5|104.9MB|1
```

LsPort [–H] [–h hostname [hostname..] [–i serialnb [serialnb...]]

Displays the list of ports (fibre channel attachment).

If no parameter is provided, all known ports are displayed.

If the **–H** option is present, the first line of the command output is a header.

If **–h** option is present, the ports of the hosts whose names are given as parameters are displayed.

If **–i** option is present, the ports of the SAN components whose serial numbers are given as parameters are displayed.

A command can contain both **–h** and **–i** options.

The command output displays one line per port with the following format:

```
component family|component name|component model|component physical
identifier|port name|port WWN|port ID|port location
```

where:

component	SAN component that owns the port.

- Example:

```
#sanit -s 49584 -c LsPort -i 9264297 -h csmgt14

Subsystem||DAS_5700|9264297|SPA|200000601631A2FE|0000EF|
Subsystem||DAS_5700|9264297|SPB|200000601631A304|0000E8|
Host|csmgt14|Bull ESCALA T (e8ED)|00400035E800|fchan0|10000000C920D201|021500|04-01
Host|csmgt14|Bull ESCALA T (e8ED)|00400035E800|fchan1|10000000C920C1EF|000001|04-03
```

LsPortLuns [–H] –i serialnb –p portname

Displays the physical LUNs for the disk array specified by its serial number (**–i** parameter) and seen from the port defined by the **–p** parameter (**portname** is the connected port).

If the **–H** option is present, the first line of the command output is a header.

The command output displays one line per set of LUNs with the following format:

```
PORTLUNS|FLUN|TYPE|CAPACITY|LDSTATE
```

where:

PORTLUNS	header of the results
TYPE	type of LUNs(RAID 0 for example)
FLUN	number of the physical LUN seen by this port.
CAPACITY	capacity of the LUN
LDSTATE	state of the LUN.

- Example:

```
# sanit -s 123 -c lsportluns -i 0000000927900358 -p 00-01
PORTLUNS|003c|RAID0|4.0GB|ready
PORTLUNS|003d|RAID0|4.0GB|ready
PORTLUNS|003e|RAID0|4.0GB|ready
PORTLUNS|003f|RAID0|4.0GB|ready
PORTLUNS|0040|RAID0|4.0GB|ready
PORTLUNS|0041|RAID0|4.0GB|ready
```

LsSubSystem [–H] [hostname]

Displays the list of Disks subsystems. If `hostname` is provided, the list is restricted to the Disk subsystems that have a path to the given host.

If the **–H** option is present, the first line of the command output is a header.

The command output displays one line per subsystem with the following format:

```
SUBSYSTEM|subsytem name|model|physical identifier|IP address|monitoring
status
```

- Example:

```
#sanit -s 49584 -c LsSubSystem csmgt14
Subsystem|EMC|EMC_3300|182600940||NOT_MONITORED
Subsystem||DAS_5720|9264297||NOT_MONITORED
Subsystem||DAS_3500|9646169||NOT_MONITORED

fchan0|10000000C920D26E|EMC_3300|182600940|15b|50060482B8913B1E|00FF000000000000:00FF000000000000|
          VCM Base|7.9MB|||0|1|1|
fchan1|10000000C920B29A|EMC_3300|182600940|15a|50060482B8913B0E|0010000000000000:0010000000000000|
          RAID-S(local)|4.5GB|||0|1|1|
fchan1|10000000C920B29A|EMC_3300|182600940|15a|50060482B8913B0E|0021000000000000:0021000000000000|
          Disk Drive|4.5GB|hdisk255||1|1|1|A
fchan1|10000000C920B29A|EMC_3300|182600940|15a|50060482B8913B0E|0001000000000000:0001000000000000|
          RAID-1(local)|Unknown|hdisk2||1|1|1|A
fchan1|10000000C920B29A|EMC_3300|182600940|15a|50060482B8913B0E|0012000000000000:0012000000000000|
          RAID-S(local)|4.5GB|||0|1|1|
fchan1|10000000C920B29A|EMC_3300|182600940|15a|50060482B8913B0E|00A0000000000000:00A0000000000000|
          Disk Drive|2.9MB|||0|1|1|
fchan1|10000000C920B29A|EMC_3300|182600940|15a|50060482B8913B0E|0002000000000000:0002000000000000|
          RAID-1(local)|4.5GB|||0|1|1|
fchan1|10000000C920B29A|DAS_5700|9264297|SPA|200000601631A2FE|000C000000000000:000C000000000000|
          RAID+5|104.9MB|hdisk273||1|1|1|A
fchan1|10000000C920B29A|DAS_5700|9264297|SPA|200000601631A2FE|0015000000000000:0015000000000000||||
          |1|0|1|B
fchan1|10000000C920B29A|DAS_5700|9264297|SPA|200000601631A2FE|0000000000000000:0000000000000000|
          DISK+|8.7GB|hdisk270||1|1|1|A
fchan1|10000000C920B29A|DAS_5700|9264297|SPA|200000601631A2FE|0001000000000000:0001000000000000||||
          |1|0|1|B
fchan1|10000000C920B29A|DAS_5700|9264297|SPA|200000601631A2FE|0002000000000000:0002000000000000|
          RAID 5|17.5GB|||0|1|1|
fchan1|10000000C920B29A|DAS_5700|9264297|SPA|200000601631A2FE|0042000000000000:0042000000000000|
          RAID 5|104.9MB|||0|1|1|
fchan1|10000000C920B29A|DAS_5700|9264297|SPA|200000601631A2FE|0043000000000000:0043000000000000|
          RAID+5|104.9MB|hdisk275||1|1|1|A
fchan1|10000000C920B29A|DAS_5700|9264297|SPA|200000601631A2FE|0064000000000000:0064000000000000|
          RAID+5|104.9MB|hdisk276||1|1|1|A
fchan1|10000000C920B29A|DAS_5700|9264297|SPA|200000601631A2FE|000B000000000000:000B000000000000|
          RAID+5|104.9MB|hdisk272||1|1|1|A
fchan0|10000000C920D26E|DAS_5700|9264297|SPB|200000601631A304|0000000000000000:0000000000000000||||
          |1|0|1|B
fchan0|10000000C920D26E|DAS_5700|9264297|SPB|200000601631A304|0001000000000000:0001000000000000|
          DISK+|Unknown|hdisk277||1|1|1|A
fchan0|10000000C920D26E|DAS_5700|9264297|SPB|200000601631A304|0043000000000000:0043000000000000||||
          |1|0|1|B
fchan0|10000000C920D26E|DAS_5700|9264297|SPB|200000601631A304|000B000000000000:000B000000000000||||
          |1|0|1|B
fchan0|10000000C920D26E|DAS_5700|9264297|SPB|200000601631A304|0064000000000000:0064000000000000||||
          |1|0|1|B
```

RemLunGroup –i serialnb –n LunSetName

Removes the set of LUNs specified by its name (**–n** parameter), for the disk array specified by its serial number (**–i** parameter). The **LunSetName** syntax depends on the subsystem model (see Chapter 11. *Supported SAN Components*).

This command is not allowed if the S@N.IT! CLI session has been opened for **common** user.

- Example:

```
# sanit -s 123 -c remlungroup -i 0000000927909999 -n AX:Server_c
```

RemLunGroupContent –i serialnb –n LunSetName –l <lun_id> [lun_id*n*]

Removes LUNs from the set of LUNs specified by its name (**–n** parameter), for the disk array specified by its serial number (**–i** parameter). The LUN numbers are specified by their **lun_id** (**–l** parameter). Each **lun_id** is a number, separated by a space. The **LunSetName** syntax depends on the subsystem model (see Chapter 11. *Supported SAN Components*).

This command is not allowed if the S@N.IT! CLI session has been opened for **common** user.

- Example:

```
# sanit –s 123 –c remlungroupcontent –i 10000000CC99999 –n AX:Server_c –l 0 2
```

RemLunGroupPath –i serialnb –n LunSetName –w [wwn_id wwn_n1 ...]

Removes the world wide name(s) of a HBA from a set of LUNs specified its name (**–n** parameter), for the disk array specified by its serial number (**–i** parameter). Each wwn is separated by a space. The **LunSetName** syntax depends on the subsystem model (see Chapter 11. *Supported SAN Components*).

This command is not allowed if the S@N.IT! CLI session has been opened for **common** user.

- Example:

```
# sanit –s 123 –c remlungrouppath –i 10000000000000927909999
–n AX:Server_c –w 10000000CC99900
```

SetAclMode [–i serialnb] –s [on | off]

Activates / De–activates the LUN masking feature of a disk array specified by its serial number (**–i** parameter).

This command is not allowed if the S@N.IT! CLI session has been opened for **common** user.

- Example:

```
# sanit –s 123 –c setaclmode –i 0000000927909999 –s on
```

SetPassword <common|sanadmin> oldpassword newpassword

Modifies the password of the S@N.IT! user specified in the command. This command is not allowed if the S@N.IT! CLI session has been opened for user **common**.

`user` S@N.IT! user (either `sanadmin` or `common`).

`oldpassword` previous password for the selected S@N.IT! user.

`newpassword` new password for the selected S@N.IT! user.

- Example:

```
#sanit –s 123 –c SetPassword sanadmin admin newpass
S@N.IT! CLI session [123]: Password set.
```

SetUser <common/sanadmin> password

Changes the S@N.IT! user for the current S@N.IT! CLI session. The password of the new user must be specified.

- Example:

```
#sanit –s 123 –c SetUser sanadmin admin
```

UpdateComponent [–H | –f family –i serialnb [–n LunSetName] [–N] [–a <property >=<value> ...]]

Modifies the properties of one SAN component. The **family**, **serialnb** and **name** properties must define a unique component. The values allowed for **family**, and the **name** corresponding to the selected family can be listed using the **–H** option.

This command is not allowed if the S@N.IT! CLI session has been opened for **common** user.

- Examples:

```
# sanit –s 123 –c UpdateComponent –f Library –i 45678 –aCompoName=”my
library”
```

UpdatePort [–H|–w wwn [–i serialnb] [–a< property >=<value> ...]]

Modifies the properties of a fibre channel port specified by the **wwn** (world wide name) parameter. The owner of the port can be modified using **–i serialnb** to specify the SAN component to which the port must be attached.

The values allowed for the **property** names corresponding to the selected **family** can be listed using the **–H** option.

This command is not allowed if the S@N.IT! CLI session has been opened for **common** user.

- Example:

```
# sanit –s 123 –c updatePort –w 10000000CC99999 –i 789 –aPortName=”adapter
nb 2”
```

Chapter 8. S@N.IT! SNMP Agent

S@N.IT! SNMP agent is launched as part of the S@N.IT! Server. It manages two MIBs:

- A Fibre Alliance MIB, which provides SAN topology and status information.
- A proprietary MIB, which provides mainly LUN mapping information (see S@N.IT! *Proprietary MIB*, on page 8-8).

S@N.IT! SNMP agent generates traps conformant to Fibre Alliance MIB specifications (Refer to ASN.1 definition of trap PDUs in RFC 1157).

To avoid conflict with other SNMP agents on the S@N.IT! Server, the S@N.IT! SNMP agent uses a dedicated port, defined in **SnmpAgentPortNumber** parameter of S@N.IT! Server configuration.

SNMP manager registration can be performed statically or dynamically:

- Statically, using the following configuration parameters of the S@N.IT! Server:
  - **SnmpManagerIp**
  - **SnmpManagerPort**
  - **SnmpManagerFilter** (must be set to **8** to receive all traps)
- Dynamically, using a SET request on a new **trapRegRowState** variable in the **trapRegTable** of the fibre alliance MIB.

The list of registered SNMP managers is saved on the S@N.IT! Server. Thus, the registration of an SNMP manager is kept even if S@N.IT! services are stopped and restarted on the S@N.IT! Server.

SNMP Traps

S@N.IT! SNMP agent can send three types of traps:

1. AUTHENTICATION_FAILURE (4) generic trap if bad *community* field received. The *community* must be:
   - *public* or the *sandamin* password for a GET/NEXT request,
   - the *sandamin* password for a SET request.
2. COLD_START (0) generic trap when the S@N.IT! SNMP agent is started.
3. ENTERPRISE_SPECIFIC (6) Fibre Alliance trap with the following specific trap values:
   - connUnitStatusChange (1)
   - connUnitDeletedTrap (3)
   - connUnitEventTrap (4)

connUnitStatusChange (1)

This trap is sent to SNMP managers registered with a trap filter greater or equal to 3 (alert), when the monitoring status of a SAN component changes. Its ASN.1 definition is specified as follows in the Fibre Alliance MIB:

```
connUnitStatusChange TRAP-TYPE
ENTERPRISE fcmgmt
VARIABLES { connUnitStatus, connUnitState }
DESCRIPTION
"The overall status of the connectivity unit has changed.
Recommended severity level (for filtering): alert"
::= 1
```

The variables used in the **connUnitStatusChange** trap are **connUnitStatus** and **connUnitState**, defined as follows:

connUnitStatus

- ASN.1 definition

```
connUnitStatus OBJECT-TYPE
SYNTAX INTEGER {
         unknown(1),
         unused(2), -- cannot report status
         ok(3), -- available for meaningful work
         warning(4), -- something needs attention
         failed(5) -- something has failed
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Overall status of the connectivity unit. The goal of this object is to be
the single poll point to check the status of the connunit. If there is any
other component that has warning, then this should be set to warning, etc."
::= { connUnitEntry 6 }
```

- OID

  This variable is the 6th member of the **connUnitEntry** which is an entry in the **connUnitTable**. This table is indexed by the **connUnitId** *(*see *connUnitDeletedTrap (3)*, on page 8-3), which is the 1st member of **connUnitEntry**, whilst the serial number is stored in **connUnitSn** (8th member). So its OID is:

```
(iso)1.(org)3.(dod)6.(internet)1.(experimental)3.(fcmgt)94.(connSet)1.(conn
UniTable)6.(connUnitEntry)1.(connUnitStatus)6.<index = 16 hexadecimal chars
representing the connUnitId value>
```

  Example:

```
.iso.org.dod.internet.experimental.94.1.6.1.6.57.54.52.54.49.54.57.0.0.0.0.
0.0.0.0.0.0
                     [9]  [6] [4] [6] [1] [6] [9] <  padding    >
```

- Value

S@N.IT! Monitoring Status	ConnUnitStatus value
UNKNOWN	unknown(1)
NOT_MONITORED	unused(2)
NORMAL	ok(3)
FAULTY	failed(5)

connUnitState

- ASN.1 definition

```
connUnitState OBJECT-TYPE
SYNTAX INTEGER {
         unknown(1),
         online(2), -      available for meaningful work
         offline(3) -      unavailable for meaningful work, for
                     example in self-test mode, configuration, etc.
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
         "Overall state of the connectivity unit."
::= { connUnitEntry 5 }
```

- OID

  This variable is the 5th member of the **connUnitEntry** which is an entry in the **connUnitTable**. This table is indexed by the **connUnitId** *(*see *connUnitDeletedTrap (3)*, on page 8-3), which is the 1st member of **connUnitEntry**, whilst the serial number is stored in **connUnitSn** (8th member). So its OID is:

```
(iso)1.(org)3.(dod)6.(internet)1.(experimental)3.(fcmgt)94.(connSet)1.(conn
UniTable)6.(connUnitEntry)1.(connUnitState)5.<index = 16 hexa chars
representing the connUnitId value>
```

  Example:

```
.iso.org.dod.internet.experimental.94.1.6.1.5.57.54.52.54.49.54.57.0.0.0.0.
0.0.0.0.0.0
                            [9] [6] [4] [6] [1] [6] [9] <  padding       >
```

- Value

  S@N.IT! SNMP agent always set the value to unknown(1).

connUnitDeletedTrap (3)

This trap is sent to SNMP managers registered with a trap filter greater or equal to 6 (warning), when a SAN component is deleted .

Its ASN.1 definition is specified as follows in the Fibre Alliance MIB:

```
connUnitDeletedTrap TRAP-TYPE
ENTERPRISE fcmgmt
VARIABLES { connUnitId }
DESCRIPTION
         "A connUnit has been deleted from this agent.
         Recommended severity level (for filtering): warning"
::= 3
```

The variable used in the **connUnitDeletedTrap** trap is **connUnitId**, defined as follows:

connUnitId

- ASN.1 definition

```
connUnitId OBJECT-TYPE
SYNTAX FcGlobalId
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The unique identification for this connectivity unit among those within
this proxy domain. The value MUST be unique within the proxy domain because
it is the index variable for connUnitTable. The value assigned to a given
conectivity unit SHOULD be persistent across agent and unit resets. It
SHOULD be the same as connUnitGlobalId if connUnitGlobalId is known and
stable."
::= { connUnitEntry 1 }
```

This variable is used as an index for **connUnitTable**. Its type is **FcGlobalId**, which is defined as:

```
(FcGlobalId ::= OCTET STRING (SIZE(16))
```

- OID

  This variable is the 1st member of the **connUnitEntry**, which is an entry in the **connUnitTable** and it is the index of this table whilst the serial number is stored in **connUnitSn** (8th member). So its OID is:

```
(iso)1.(org)3.(dod)6.(internet)1.(experimental)3.(fcmgt)94.(connSet)1.(conn
UniTable)6.(connUnitEntry)1.(connUnitId)1.<index = 16 hexa chars
representing the connUnitId value>
```

  Example:

```
.iso.org.dod.internet.experimental.94.1.6.1.1.100.52.52.52.52.52.52.52.52.5
2.52.52.0.0.0.0
    hexa          64 34 34 34 34 34 34 34 34 34 34 34 34 <padding>
    ascii         [d] [4] [4] [4] [4] [4] [4] [4] [4] [4] [4] [4] [4]
```

- Value

  The **connUnitId** is build from the physical Id (serial number) of the object but with possible translation (e.g. ASCII to HEXA translation): The Physical Ids must be constituted of hexadecimal chars, all other chars are removed from the String, and they are converted into an array of 16 bytes (if the Id is longer, it is truncated).

  The physical ID of a S@N.IT! Component is stored in the **connUnitSn** MIB object which can be retrieved using the OID instance of the object.

  In the example mentioned above the OID of the serial number is:

```
.iso.org.dod.internet.experimental.94.1.6.1.8.100.52.52.52.52.52.52.52.52.5
2.52.52.0.0.0.0
```

connUnitEventTrap (4)

This trap is sent to SNMP managers when one of the following events occurs, and the trap filter of the SNMP manager is set to a level greater or equal to the event severity:

New Connection added	(**severity = 7 – notify**)
Connection deleted	(**severity = 6 – warning**)
SAN Component added	(**severity = 7 – notify**)
SAN Component deleted	(**severity = 6 – warning**)
SAN Component status changed	(**severity = 3 – alert**)
Activation or Deactivation of the LUN access control mechanism for a SAN agent	(**severity = 6 – warning**)
LUN mapping	(**severity = 8 – info**)
LUN un–mapping	(**severity = 8 – info**)

Its ASN.1 definition is specified as follows in the Fibre Alliance MIB:

```
connUnitEventTrap TRAP-TYPE
ENTERPRISE fcmgmt
VARIABLES { connUnitEventId,
         connUnitEventType,
         connUnitEventObject,
         connUnitEventDescr }
DESCRIPTION
         ”An event has been generated by the connectivity unit.
         Recommended severity level (for filtering): info”
::= 4
```

The variables used in the **connUnitEventTrap** trap are **connUnitEventId**, **connUnitEventType**, **connUnitEventObject**, and **connUnitEventDescr**, defined as follows:

connUnitEventId

- ASN.1 definition

```
connUnitEventId OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS deprecated
DESCRIPTION
"The internal event Id. Incremented for each event, ranging between 1 and
connUnitMaxEvents. Not used as table index to simplify the agent
implementation. When this reaches the end of the range specified by
connUnitMaxEvents, the Id will roll over to start at one. This value will
be set back to one at reset. The relationship of this value to the index is
that internal event id may represent a smaller number than a 32 bit integer
(eg max 100 entries) and would only have a value range up to
connUnitMaxEvents."
::= { connUnitEventEntry 3 }
```

- OID

  This variable is the 3rd member of the **ConnUnitEventEntry**, which is an entry in the **connUnitEventTable**. This table is indexed by the **connUnitEventUnitId** (1st member and equals to the **connUnitId** of the component concerned by the event) and **connUnitEventIndex** (2nd member). So its OID is:

```
(iso)1.(org)3.(dod)6.(internet)1.(experimental)3.(fcmgt)94.(connSet)1.(conn
UnitEventTable)11.(connUnitEventEntry)1.(connUnitEventId)3.<index = 16 hexa
chars representing the connUnitEventId value>< connUnitEventIndex>
```

  Example:

```
iso.org.dod.internet.experimental.94.1.11.1.3.211.51.51.51.51.51.0.0.0.0.0.
0.0.0.0.0.0.22
         ascii        [0]  [3] [3] [3] [3] [3] < padding  > <idx>
```

- Value

  S@N.IT! SNMP agent sets this variable to an internal counter.

connUnitEventType

- ASN.1 definition

```
connUnitEventType OBJECT-TYPE
SYNTAX INTEGER {
         unknown(1),
         other(2),
         status(3),
         configuration(4),
         topology(5)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
         "The type of this event."
::= { connUnitEventEntry 7 }
```

- OID

  This variable is the 7th member of the **ConnUnitEventEntry**, which is an entry in the **connUnitEventTable**. This table is indexed by the **connUnitEventUnitId** (1st member and equals to the **connUnitId** of the component concerned by the event) and **connUnitEventIndex** (2nd member). So its OID is:

```
(iso)1.(org)3.(dod)6.(internet)1.(experimental)3.(fcmgt)94.(connSet)1.(conn
UnitEventTable)11.(connUnitEventEntry)1.(connUnitEventType)7.<index = 16
hexa chars representing the connUnitEventId value>< connUnitEventIndex>
```

Example:

```
iso.org.dod.internet.experimental.94.1.11.1.7.211.51.51.51.51.51.0.0.0.0.0.
0.0.0.0.0.22
          ascii        [0]  [3] [3] [3] [3] [3] < padding       > <idx>
```

- Value

S@N.IT! event	ConnUnitEventType value
New connection added	topology(5)
Connection deleted	topology(5)
SAN Component added	topology(5)
SAN Component deleted	topology(5)
SAN Component status changed	status(3)
Activation or Deactivation of the LUN access control mechanism for a SAN agent	configuration(4)
LUN mapping	configuration(4)
LUN un–mapping	configuration(4)

connUnitEventObject

- ASN.1 definition

```
connUnitEventObject OBJECT-TYPE
SYNTAX OBJECT IDENTIFIER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This is used with the connUnitEventType to identify which object the event
refers to. Examples are connUnitPortStatus.connUnitId.connUnitPortIndex,
connUnitStatus.connUnitId, etc."
::= { connUnitEventEntry 8 }
```

- OID

This variable is the 8th member of the **ConnUnitEventEntry** which is an entry in the **connUnitEventTable**. This table is indexed by the **connUnitEventUnitId** (1st member and equals to the **connUnitId** of the component concerned by the event) and **connUnitEventIndex** (2nd member). So its OID is:

```
(iso)1.(org)3.(dod)6.(internet)1.(experimental)3.(fcmgt)94.(connSet)1.(conn
UnitEventTable)11.(connUnitEventEntry)1.(connUnitEventObject)8.<index = 16
hexa chars representing the connUnitEventId value>< connUnitEventIndex>
```

Example:

```
iso.org.dod.internet.experimental.94.1.11.1.8.211.51.51.51.51.51.0.0.0.0.0.
0.0.0.0.0.22
          ascii        [0]  [3] [3] [3] [3] [3] < padding    > <idx>
```

- Value

The value is set to the OID of the object that is involved in the event

S@N.IT! event	Object which OID is set as connUnitEventObject value
New connection added	connUnitLinkUnitId
Connection deleted	connUnitLinkUnitId
SAN component added	connUnitId
SAN component deleted	connUnitId
SAN component status changed	connUnitStatus
Activation or Deactivation of the LUN access control mechanism for a SAN agent	SanLbhLacState (from sanit.MIB)
LUN mapping	SanLbhMapFlag (from sanit.MIB)
LUN un–mapping	SanLbhMapFlag (from sanit.MIB)

connUnitEventDescr

- ASN.1 definition

```
connUnitEventDescr OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-only
STATUS mandatory
DESCRIPTION
        "The description of the event."
::= { connUnitEventEntry 9 }
```

- OID

  This variable is the 9th member of the **ConnUnitEventEntry** which is an entry in the **connUnitEventTable**. This table is indexed by the **connUnitEventUnitId** (1st member and equals to the **connUnitId** of the component concerned by the event) and **connUnitEventIndex** (2nd member). So its OID is:

```
(iso)1.(org)3.(dod)6.(internet)1.(experimental)3.(fcmgt)94.(connSet)1.(conn
UnitEventTable)11.(connUnitEventEntry)1.( connUnitEventDescr)9.<index = 16
hexa chars representing the connUnitEventId value>< connUnitEventIndex>
```

  Example:

```
.iso.org.dod.internet.experimental.94.1.11.1.9.211.51.51.51.51.51.0.0.0.0.0
.0.0.0.0.0.22
        ascii        [0]  [3] [3] [3] [3] [3] < padding  > <idx>
```

- Value

S@N.IT! Event	ConnUnitEventDescr value
New connection added	"*Connection added between [<a> (<b>)] port [<c>] and [<d> (<e>)] port [<f>].*" <*a*>: physical id of component 1 <*b*>: <model – name>(*) of component 1 <*c*>: WWN of port 1 <*d*>: physical id of component 2 <*e*>: <model – name>(*) of component 2 <*f*>: WWN of port 2
Connection deleted	"*Connection deleted between [<a> (<b>)] port [<c>] and [<d> (<e>)] port [<f>].*" <*a*>: physical id of component 1 <*b*>: <model – name>(*) of component 1 <*c*>: WWN of port 1 <*d*>: physical id of component 2 <*e*>: <model – name>(*) of component 2 <*f*>: WWN of port 2
SAN Component added	"*Component [<a> (<b>)] added.*" With: <*a*>: physical id of component <*b*>: <model – name>(*) of component
SAN Component deleted	"*Component [<a> (<b>)] deleted.*" With: <*a*>: physical id of component <*b*>: <model – name>(*) of component
SAN Component status changed	"*Status of component [<a> (<b>)] changed from [<c>] to [<d>].*" <*a*> : physical id of component <*b*> : <model – name>(*) of component <*c*> : previous status of component <*d*> : new status of component
Activation or Deactivation of the LUN access control mechanism for a SAN agent	"*Lun access control of component [<a> (<b>)] changed from [<c>] to [<d>].*" <*a*>: physical id of SAN agent <*b*>: <model – name>(*) of SAN agent <*c*>: previous LUN access control state of SAN agent <*d*>: new LUN access control state of SAN agent

LUN mapping	"*Mapping added for LUN [<a>] for subsystem [<b> (<c>)]/port [<d> (<e>)] and host [<f> (<g>)]/adapter [<h> (<i>)].*" <*a*>: LUN <*b*>: physical id of subsystem <*c*>: <model – name>(*) of subsystem <*d*>: WWN of subsystem port <*e*>: <model – name>(*) of subsystem port <*f*>: physical id of SAN agent host <*g*>: <model – name>(*) of host <*h*>: WWN of host adapter <*i*>: <model – name>(*) of host adapter
LUN un–mapping	"*Mapping deleted for LUN [<a>] for subsystem [<b> (<c>)]/port [<d> (<e>)] and host [<f> (<g>)]/adapter [<h> (<i>)].*" <*a*>: LUN <*b*>: physical id of subsystem <*c*>: <model – name>(*) of subsystem <*d*>: WWN of subsystem port <*e*>: name of subsystem port <*f*>: physical id of S@N.IT! Agent host <*g*>: <model – name>(*) of host <*h*>: WWN of host adapter <*i*>: name of host adapter

(*)<*model – name*>: *model* represents the model of the component involved in the event (example : DAS_4700), *name* is the logical name of this object: for a host it is the value returned by the **hostname** command; for a switch it is the DNS name matching the IP address of the switch, for a storage device, it can be set by the S@N.IT! administrator via the S@N.IT! GUI. This field can be empty if no value has been set.

S@N.IT! Proprietary MIB

The S@N.IT! proprietary MIB provides mainly LUN mapping information. Its ASN.1 formal description is delivered in the **sanit.mib** file located in the /**etc**/**sanit**/**mibs**/ directory on AIX and Linux hosts, and in the **<%installation directory>****mibs** directory on Windows hosts.

Chapter 9. Best Practices

This chapter describes how to manage the changes that occur in a SAN configuration. The following tasks are detailed:

Adding a Host to the SAN

The S@N.IT! software must be installed and the host rebooted before the host is connected to the SAN to ensure that the new host will have an active LUN Access Control mechanism.

Once the host is configured as a S@N.IT! Agent, it will automatically register to the S@N.IT! Server and appear on running instances of the S@N.IT! GUI within a few minutes.

Adding a SAN Component to the SAN

Adding a Switch or a Hub

1. Configure the IP address of the switch or managed hub.
2. Connect the switch or managed hub to the TCP/IP network (it must be reachable from the host on which the S@N.IT! Server runs).
3. Update the configuration of the fibre channel adapters on the host if needed (for example: change from “point–to–point” to “loop” or vice versa).
4. Refresh the configuration of fibre channel drivers on all S@N.IT! Agent hosts (depending on the operating system, this may require to launch a configuration command or to reboot the host – see details in Chapter 12. *Supported Platforms*.
5. If the new component is a switch, it will be discovered and will appear within running S@N.IT! GUI instances.
6. If the new component is a hub, it will not be discovered automatically. Use the Edit/Create Component menu from S@N.IT! GUI (see *Edit /Create Component*, on page 6-26).

Adding a Supported Subsystem

The subystem preparation must be perfomed (LUN bindings) out of the SAN (see *Disk subsystems*, on page 11-3 for details about DAS and Symmetrix subsystems).

The following steps must then be performed:

1. Connect the subsystem.
2. Refresh the configuration of fibre channel drivers on all S@N.IT! Agent hosts (depending on the operating system, this may require to launch a configuration command or to reboot the host – see details in Chapter 12. *Supported Platforms*.

The subsystem will appear within one minute on running S@N.IT! GUI instances. Mapping operations can then be started to allocate the LUNs of the subsystems to the S@N.IT! Agents.

Note that the software associated to the disk subsystems (Navisphere for DAS, ECC for Symmetrix) usually needs some LUNs to be available on the host(s) on which this software runs, thus these LUNs (such as Symmetrix gatekeepers) must be mapped quickly for these hosts, before the management software is (re–) configured.

Adding a Non Supported Subsystem or Library

1. Connect the subsystem.
2. Refresh the configuration of fibre channel drivers on all S@N.IT! Agent hosts (depending on the operating system, this may require to launch a configuration command or to reboot the host – see details in Chapter 12. *Supported Platforms*.
3. The new SAN device will appear within one minute on running S@N.IT! GUI instances as a library or subsystem SAN component (or several SAN Components if it has several fibre channel ports).

Important: on Windows hosts it is necessary to explicitely map the LUNs of the new devices – see details in *Windows Platforms*, from page 12-5.

Replacing / Moving a Fibre Channel Adapter in a S@N.IT! Agent Host

The LUN Access Control information is attached to the adapter location (PCI bus and slot) within the S@N.IT! Agent host. Thus if the replacement occurs in the same PCI slot, the LUN Access Control is preserved.

In other cases (fibre channel adapter moved from a PCI slot to another, Fibre channel adapter replaced by another one but in another PCI slot):

- the LUN mappings for the initial slot are not visible through S@N.IT! but remain on the S@N.IT! Agent, thus if an adapter is re–plugged into this slot, the LUN mappings will become effective again (unmap all LUNs for the adapter's original location before removing it, unless the location will be used again with the same mapping),
- the LUN mapping operations must be re–run for the new location.

Adding an Adapter to a S@N.IT! Agent

If the new adapter is installed to bring a new path between a S@N.IT! Agent host and a storage device, this is usually to bring redundancy in conjunction with a failover software (such as Powerpath).

LUNs that were mapped on the already available paths between the host and the disk subsystem need to be mapped for the new path, and the failover software needs to be reconfigured to take the new path into account.

The operation to be performed depends on:

- The ability of the failover software to support dynamic addition of paths
- The way the subsystem shows the same LUN through different ports.

See *Disk subsystems*, on page 11-3 for details about Powerpath.

Replacing a Subsystem Port

The LUN Access Control information is attached to the subsystem port WWN within the S@N.IT! Agent host. Thus if the replacement of the subsystem port does not modify its WWN, the LUN Access Control is preserved.

If this is not the case:

- All LUN mappings that were done for this subsytem port must be removed (otherwise, they would be kept on the S@N.IT! Agents and would become effective again if the subsystem port is replugged).
- All LUN mapping operations that were done with the old subsystem port must be rerun.

If a failover software is used on S@N.IT! Agent host connected to the subystem, it usually needs to be reconfigured (see *Disk subsystems*, on page 11-3 for details about Powerpath).

Removing a SAN Component

If a SAN component is unplugged from the SAN, it will appear with the monitoring status UNKNOWN within the S@N.IT! GUI and can be deleted using the Edit/Delete menu (see *Edit /Delete*, on page 6-27).

Chapter 10. Troubleshooting

This chapter describes the tools available to perform troubleshooting operations. The following information is provided:

Traces

S@N.IT! manages two levels of traces:

- tracing of errors, which is always activated,
- internal traces, for maintenance purpose. The internal traces are activated only when the **DebugLabels** field is set in the **sanit.cfg** configuration file. Set this field to `DebugLabels=*` before running a `san_snap` command to gather as much information as possible, but leave it empty in normal use to improve performance.

On AIX, Linux and Solaris systems, traces are displayed on the screen where S@N.IT! process (either S@N.IT! services, GUI or CLI) has been launched, if `TraceOnScreen=true` in the **sanit.cfg** configuration file. Note that traces of errors will always be displayed, even if `TraceOnScreen=false`.

Traces are stored into circular files if `TraceOnFile=true`. These files are located in the directory defined by the **TraceDirectory** parameter of the configuration file, their maximum size (bytes) is determined by the **TraceSize** parameter.

A new trace file is created each time a S@N.IT! process is started (either S@N.IT! services, GUI or CLI). But the maximum number of trace files of each type (trace of service, trace of GUI, trace of CLI) is limited by the **MaxTraceFiles** parameter on each machine.

S@N.IT! Agent Local Commands

S@N.IT! Agent local commands must be launched on each host separately. They only act on the host where they are used and do not need the S@N.IT! application to be running or the link to the S@N.IT! Server to be operational.

Command path

AIX, Linux or Solaris path: **/usr/sanit/bin**

Windows path: **<%installation directory%>\bin**

san_info

This command displays information about the LUN access control state on the local host.

Command syntax

```
san_info
```

Example on AIX

```
# /usr/sanit/bin/san_info
```

```
HOST tau4 - LUN access control : ACTIVE
----------------------
Fiber channel adapters
----------------------
ADAPTER    - LOCATION   - FC Addr    - WWN
fcs2       - 14-08      - 011000     - 10000000C9211833
fchan1     - 2A-08      -            - 10000000C92153EB
fcs0       - 1A-08      - 021400     - 10000000C920D26E
fchan0     - 17-08      -            - 10000000C9212B5D
------------------------
SAN Subsystems discovered
------------------------
IDENTIFIER    - TYPE                  - PORT     - FC Addr  - WWN
APM000232006  - NDAS_CX400            - SPB      - 011A00   - 5006016808202136 (fscsi2) (fcs2)
WRE000217006  - NDAS_CX600            - SPB      - 011C00   - 5006016900600277 (fscsi2) (fcs2)
WRE000217006  - NDAS_CX600            - SPB      - 011E00   - 5006016800600277 (fscsi2) (fcs2)
A00ELW        - BRIDGE_Crossrds_4150  -          -          - 100000E00221BF10:SCSI (fscsi0) (fcs0)
11700363      - MCHG_EXABYTE_EXB-210  -          -          - 100000E00221BF10:0000 (fscsi0) (fcs0)
6338286       - TAPE_EXABYTE_EXB-85058SQANXR1  - -          - 100000E00221BF10:0001 (fscsi0) (fcs0)
6355365       - TAPE_EXABYTE_EXB-85058SQANXR1  - -          - 100000E00221BF10:0002 (fscsi0) (fcs0)
A00ELW        - BRIDGE_Crossrds_4150  -          - 021100   - 100000E00221BF10 (fscsi0) (fcs0)
-----------------------------
LUN mapping (ACTIVE)
-----------------------------
Subsystem Identifier - Port WWN             - LUN              - Adapter WWN      - Disk
WRE000217006         - 5006016800600277     - 0003000000000000 - 10000000C9211833 - hdisk28
WRE000217006         - 5006016800600277     - 0003000000000000 - 10000000C9211833 - hdisk28
F41998431924F41998431964 - 21000020370082F2 - 0000000000000000 - 10000000C920D26E - hdisk10
```

Example on Windows

```
C:\Program Files\Bull\S@N.IT\bin>san_info
```

```
**
** HOST CSMGT29 - LUN access control : ACTIVE**
** -------------------------
** SAN Subsystems discovered
** -------------------------**
** IDENTIFIER  - TYPE        - PORT      - WWN
** 182600940   - EMC_3300    - 15a       - 50060482B8913B0E
** 182600940   - EMC_3300    - 15b       - 50060482B8913B1E
** 9264297     - DAS_5700    - SPA       - 200000601631A2FE
** 9264297     - DAS_5700    - SPB       - 200000601631A304
** 9646169     - DAS_3500    - SPB       - 200000601627D8FC
** 9646169     - DAS_3500    - SPA       - 200000601627D862
**
**
** ----------------------
** LUN mapping (ACTIVE)
** ----------------------
**
9646169^200000601627D8FC^0000000000000000^10000000C9212CA2^NT LUN
09646169^200000601627D8FC^0001000000000000^10000000C9212CA2^NT LUN
19646169^200000601627D8FC^0003000000000000^10000000C9212CA2^Disk
19646169^200000601627D862^0000000000000000^10000000C9212CA2^Disk
29646169^200000601627D862^0001000000000000^10000000C9212CA2^Disk
39646169^200000601627D862^0003000000000000^10000000C9212CA2^NT LUN 3
```

san_snap

This command gathers information required for maintenance purpose.

AIX, Linux and Solaris command syntax

```
/usr/sanit/bin/san_snap [-o outputdevice] [ -d dir] [-w "problem description"]
/usr/sanit/bin/san_snap [-c ] [ -d dir] [-w "problem description"]
/usr/sanit/bin/san_snap [-r ] [ -d dir]
/usr/sanit/bin/san_snap [-v component]
```

The **san_snap** command gathers information about SAN configuration (software level for SAN related software such as S@N.IT! software, Navisphere, fibre channel driver, device configuration, errlogs, traces, LUN access control, dump, Unix) and compresses the information into a tar file. The file can then be downloaded to disk or tape, or transmitted to a remote system.

The information gathered with the **san_snap** command may be required to identify and resolve problems. The **san_snap** command requires **root** user authority.

Use the `san_snap -o/dev/rfd0` command to copy the compressed image to diskette.

Use the `san_snap -o/dev/rmt0` command to copy the compressed image to tape.

The output of the **san_snap** command is written to the **/tmp/san_it.snap** directory, unless the **–d** option is used to specify another directory. The **san_snap** command checks for available space in the directory. If there is not enough space to hold the **san_snap** command output, you must expand the file system.

Each execution of **san_snap** command appends information to previously created files. Use the **–r** flag to remove previously gathered and saved information.

Parameters

`-c`	Creates the **sanit_snap.tar.Z** file in the directory specified by `-d` option.
`-o outputdevice`	Sends information to removable output device (**/dev/rfd0**).
`-d dir`	Specifies the directory where to store information (default **./tmp/san_it.snap**).
`-r`	Remove directory (**/tmp/san_it.snap**).
`-v component`	Output component snap file to stdout. Current component choices: `dump filesys general san_it navisphere symmetrix install`.
`-w problem description`	Create README file from command line.

Windows command syntax

```
<%S@N.IT! installation directory>\bin\san_snap.bat
```

This command gathers information (such as traces, version of software) required for problem analysis into the directory: **<%S@N.IT! installation directory>\snap<hostname>**.

san_activate

This command activates the LUN access control mechanism on the local host. It requires **root** authority on AIX, Linux and Solaris hosts, **administrator** authority on Windows hosts.

Note: This command can be launched even if the S@N.IT! application is not running or if the S@N.IT! server is not reachable. If S@N.IT! GUI or CLI sessions are running on other systems they will not be notified.

The actions to be performed afterwards depend on the type of host (see *LUN Access Control*, on page 1-8 for details).

Command syntax

```
san_activate
```

san_deactivate

This command de–activates the LUN access control mechanism on the local host.

It requires **root** authority on AIX, Linux and Solaris hosts, **administrator** authority on Windows hosts.

Note: This command can be launched even if the S@N.IT! application is not running or if the S@N.IT! server is not reachable. If S@N.IT! GUI or CLI sessions are running on other systems they will not be notified.

The actions to be performed afterwards depend on the type of host (see *LUN Access Control*, on page 1-8 for details).

Command syntax

```
san_deactivate
```

san_map_lun

This command maps the specified LUN's for the specified paths (subsystem port – host HBA).

It requires **root** authority on AIX, Linux and Solaris hosts, **administrator** authority on Windows hosts.

It is part of the S@N.IT! application, thus can be launched when the application is not running or the S@N.IT! server, but:

Notes:

1. This command can be launched even if the S@N.IT! application is not running or if the S@N.IT! server is not reachable. If S@N.IT! GUI or CLI sessions are running on other systems they will not be notified.
2. The **UserPreMappingCommand** and **UserPostMappingCommand** that may be specified into the **sanit.cfg** file of the local host are not taken into account and not launched

The actions to be performed afterwards depend on the type of host (see *LUN Access Control*, on page 1-8 for details).

Command syntax

```
san_map_lun [-T] -l "LUN_ID,LUN_ID" -m portWWN adapterWWN> [-m portWWN
adapterWWN]
```

Parameters:

`-T`	This flag launches the command in trace mode.
`LUN_ID`	String of 16 hexadecimal digits (SCSI 3 address) that identifies a LUN to be mapped. For DAS subsystem with Access Logix, the LUN identifier must be the host LUN identifier (Virtual LUN).
`-m`	This flag specifies a path (subsystem port + host HBA) for which the mapping must be performed. If several paths are provided, they must all refer to the same subsystem.
`portWWN`	WWN of the port of the subsystem to which all the specified LUN's belong.
`adapterWWN`	WWN of the host HBA.

Example:

```
san_map_lun -l "0002000000000000,0003000000000000" -m  5006016800600277 10000000C9211833
```

san_unmap_lun

This command unmaps the specified LUN's for the specified paths (subsystem port – host HBA).

It requires **root** authority on AIX, Linux and Solaris hosts, **administrator** authority on Windows hosts.

It is part of the S@N.IT! application, thus can be launched when the application is not running on the S@N.IT! server, but:

Notes:

1. This command can be launched even if the S@N.IT! application is not running or if the S@N.IT! server is not reachable. If S@N.IT! GUI or CLI sessions are running on other systems they will not be notified.
2. The **UserPreUnmappingCommand** and **UserPostUnmappingCommand** that may be specified into the **sanit.cfg** file of the local host are not taken into account and not launched.

The actions to be performed afterwards depend on the type of host (see *LUN Access Control*, on page 1-8 for details).

Command syntax

```
san_unmap_lun [-T] -l "LUN_ID,LUN_ID" -m portWWN adapterWWN> [-m portWWN
adapterWWN]
```

Parameters:

`-T`	This flag launches the command in trace mode.
`LUN_ID`	string of 16 hexadecimal digits (SCSI 3 address) that identifies a LUN to be mapped. For DAS subsystem with Access Logix, the LUN identifier must be the host LUN identifier (Virtual LUN).
`-m`	This flag specifies a path (subsystem port + host HBA) for which the unmapping must be performed. If several paths are provided, they must all refer to the same subsystem.
`portWWN`	WWN of the port of the subsystem to which all the specified LUN's belong.
`adapterWWN`	WWN of the host HBA.

Example

```
san_unmap_lun -l "0002000000000000,0003000000000000" -m 5006016100600277 10000000C928C6B1
```

Chapter 11. Supported SAN Components

This chapter provides an overview of the SAN components supported by S@N.IT!. The following families of components are described:

Notes:

1. As this list may increase in further delivery please refer to the Software Release Bulletin for up–to–date information.
2. The information provided here does not supersede the compatibility matrixes of each server or storage device. Please consult the appropriate documents or contact Bull for up–to–date information.

Hosts and Adapters

Escala Servers

S@N.IT! software can be installed on Escala platforms running AIX 5.1 or later. Escala platforms with Bull PCI fibre channel adapters and drivers can be configured as SAN agents. Compatibility is ensured with clustering configuration (HA-CMP).

Monitoring of Escala S@N.IT! Agents includes:

- Control of the connection between the SAN agent and the S@N.IT! Server
- Control of the paths involved in LUN mapping
- Control of the file systems usage rate

NovaScale Linux Servers

S@N.IT! software can be installed on NovaScale servers running Linux RedHat AS 2.1 or RedHat AS 3 distributions. NovaScale servers with Bull PCI Fibre channel adapters and the appropriate drivers can be configured as S@N.IT! Agent, without host based LUN mapping mechanism.

Monitoring of NovaScale Agents includes:

- Control of the connection between the S@N.IT! Agent and the S@N.IT! Server
- Control of the file systems usage rate.

NovaScale Blade Windows Servers

S@N.IT! software can be installed on NovaScale Blade servers running Windows 2003. NovaScale Blade servers with embedded Fibre channel attachment and the appropriate driver can be configured as S@N.IT! Agent.

Monitoring of NovaScale Agents includes:

- Control of the connection between the S@N.IT! Agent and the S@N.IT! Server
- Control of the paths involved in LUN mappings
- Control of the file systems usage rate.

NovaScale Blade Linux Servers

S@N.IT! software can be installed on NovaScale Blade servers running Linux RedHat distribution. NovaScale Blade servers with embedded Fibre channel attachment and the appropriate driver can be configured as S@N.IT! Agent, without host based LUN mapping mechanism.

Monitoring of NovaScale Agents includes:

- Control of the connection between the S@N.IT! Agent and the S@N.IT! Server
- Control of the file systems usage rate.

Express 5800 Windows Server

S@N.IT! software can be installed on Express 5800 servers running Windows 2000 and Windows2003.

- Windows 2000/2003 Express 5800 platforms with Bull / Emulex PCI fibre channel drivers and the appropriate version of Emulex SCSI Port driver can be configured as S@N.IT! Agents.
- Windows 2003 Express 5800 platforms with Bull / Emulex PCI fibre channel drivers and the appropriate version of Emulex mini Port driver can be configured as S@N.IT! Agents, without host based LUN mapping mechanism.

Monitoring of Express 5800 S@N.IT! Agents includes:

- Control of the connection between the SAN agent and the S@N.IT! Server
- Control of the paths involved in LUN mapping (if LUN mapping is supported)
- Control of the file systems usage rate.

Express 5800 Linux Servers

S@N.IT! software can be installed on Express 5800 running Linux RedHat, Advanced Server or SuSe distribution. Express 5800 platforms with Bull/Emulex PCI fibre channel adapters and drivers can be configured as S@N.IT! Agents, without host based LUN mapping mechanism.

Monitoring of Linux S@N.IT! Agents includes:

- Control of the connection between the S@N.IT! Agent and the S@N.IT! Server
- Control of the file systems usage rate.

Solaris Sparc Servers

S@N.IT! software can be installed on SPARC servers running Solaris SunOS. Solaris platforms with Bull/Emulex PCI fibre channel adapters and drivers can be configured as S@N.IT! Agents, without host based LUN mapping mechanism.

Monitoring of Solaris S@N.IT! Agents includes:

- Control of the connection between the S@N.IT! Agent and the S@N.IT! Server
- Control of the file systems usage rate.

EMC DAS / NDAS Disk Subsystems

The following DAS models are managed:

- DAS 3500, 4500, 4700, 4700–2, 5300, 5400, 5700, 5720
- NDAS CX200, CX300, CX400, CX500, CX600, CX700
- NDASG10x.

This management includes:

- automatic discovery.
- LUN Access Control management for SAN agents.
- Monitoring (no specific configuration required, except for DAS 4700 and NDAS family).
- Launching of Navisphere from S@N.IT! GUI (if Navisphere Manager or Supervisor is installed on the platform where the S@N.IT! GUI has been launched).

Monitoring DAS 4700 and NDAS Family

The monitoring of these subsystems requires that the following conditions are met:

- The IP address of one subsystem SP (Service Processor) is manually specified in the Properties menu of the S@N.IT! GUI.
- The S@N.IT! Server can reach the Ethernet port of the subsystem SP.
- The SNMP agent is installed and configured in the subsystem.

  You can check that the FA MIB software (the SNMP Agent) is installed on the subsystem, and install it if necessary, using Navisphere Manager (as outlined for SnapView or MirrorView in the *Navisphere Manager* documentation) or Navisphere CLI.

  With Navisphere CLI, logged on as **root** or **admin**, enter a command similar to the following one:

  – To check the installed software:

```
navicli -h <sp-hostname> ndu -list
```

  – To install the SNMP Agent:

```
navicli -h <sp-hostname> ndu -install <%directory>\fa_mib.ndu
```

  Please refer to the Navisphere Manager documentation for details about software installation.

Monitoring NDASG10x

Note: NDASG10x are identified as **NDAS_AX100** in S@N.IT! GUI.

The only way to monitor this subsystem from S@N.IT! is to configure the S@N.IT! Server to monitor via traps (by setting the **ActivateTrapListener** parameter to **true**) and to configure the array to send traps to the platform where the S@N.IT! Server runs (using its NaviSphere interface to specify either the IP address or the name of the platform).

Note: The subsystem sends traps only for its first error. It does not send traps when it comes back to a normal state.

Access Logix

Compatibility between S@N.IT! LUN Access Control mechanism and Access Logix is ensured, but the Information/Host LUNs tab of the S@N.IT! GUI shall be used (rather than the Information/Subsystem LUNs tab) to map/unmap LUNs to make sure that Access Logix really allows the SAN host to access LUNs.

Navisphere Agent on S@N.IT! Agents

Navisphere agent can be installed on S@N.IT! Agents, but special attention is required when using Navisphere components and S@N.IT! LUN AccessControl on Windows hosts.

Navisphere agent is configured to communicate with a disk subsystem through a path to a LUN. Once the LUN Access Control is activated on a host and no LUN is mapped, all disk subsystem LUNs are masked for this host, especially LUN(s) used by Navisphere agent.

Consequently, Navisphere can't become operational; for example this occurs after the first installation of S@N.IT!.

To prevent that problem, it is recommended to create and map one reserved LUN per host (through a single path) on each disk subsystem. These LUNs will be dedicated to Navisphere agents, they should not contain operational data, they can be very small, and they must not be denied (un–mapped).

Before using LUNs, they must have been created. This operation is generally performed once, the first time the disk subsystem is powered–on. Navisphere Supervisor is generally used to perform this initial configuration. A Navisphere agent located on the host and a direct access to the disk subsystem through the SAN are required.

S@N.IT!.provides LUN Access Control functions. These functions apply consequently when one LUN at least has been created. In addition, when the global LUN Access Control is activated, the communication between the related host and disk subsystem is closed whether LUNs exist or not.

In this context, when a new DAS subsystem is to be connected to an existing SAN, there are only two possibilities to provide a direct access to the DAS subsystem in order to prepare it:

- The first possibility – which is the recommended solution – is to perform the preparation operation outside the SAN and to connect the disk subsystem to the SAN when at least one LUN has been created.
- The second possibility is to:
  - de–activate the LUN Access Control from the related host,
  - prepare the disk subsystem ,
  - activate the LUN Access Control again,
  - map previously created LUNs,
  - reconfigure the Navisphere Agent .

  However this operation is not recommended due to the possibility that the related host will be able to access all LUNs seen from its HBAs during this phase.

Installation rules on a Windows S@N.IT! Agent connected to a DAS subsystem

Navisphere management application (NT Navisphere Manager or Windows Navisphere Supervisor or AIX Navisphere Manager) uses an agent to communicate with disk subsystems through the SAN with one or several LUNs.

But S@N.IT! LUN Access Control can deny access to LUNs, in particular LUNs used by Navisphere agent: in this case, Navisphere management application cannot be operational.

To prevent that, it is recommended that some LUNs are dedicated to Navisphere agent(s). These LUNs should contain no data and can be very small.

Access must be permanently allowed for LUNs dedicated to hosts that run Navisphere agent(s) and never be denied.

Follow these main installation steps for Windows platforms that will run a Navisphere agent:

1. Connect the hardware components.
2. If it exists, clean the Emulex driver environment and install the Emulex driver (refer to *Installing Emulex Software*, on page 0).
3. Install Navisphere agent (refer to the Navisphere documentation).
4. Create (using the Navisphere management application) as many LUNs as hosts on which Navisphere agent runs: these LUNs will be dedicated to Navisphere activity.
5. Install the S@N.IT! application
6. Start the S@N.IT! GUI and allow access to each LUN to hosts on which Navisphere agents are running
7. Reconfigure the Navisphere agent on each host running it:
   a. Start the Navisphere Agent Configurator.
   b. Clear device list.
   c. Auto Detect (the new device list). Detected host/SP link is displayed as follows in the 'Navisphere Agent Configurator' window:

```
Device Name Description
\\.\SCSIv:x:y:z SP_xxx ...
```

   d. Save.
   e. Exit the Navisphere Agent Configurator.

Replacement of a DAS SP – impact on Windows S@N.IT! Agents

This case applies when a Storage Processor (SP) is to be replaced in a disk subsystem connected to a SAN managed through S@N.IT!.

The initial state is: a SAN exists and one or several S@N.IT! agent platforms, the S@N.IT! Server and a S@N.IT! GUI are running. One or several LUNs bound on a SP are mapped to one or several hosts. This SP must be replaced. The activity from the related hosts to this SP is held.

The target state is: a new SP is plugged into the related disk subsystem, and the LUN configuration (the previous mapping of LUNs) is kept.

SPs are identified through their WWN in the system registry and LUNs are mapped regarding these SP WWNs. When a new SP, with an unknown, WWN is plugged in, LUNs that were mapped on the old SP are lost. Consequently, LUN mapping must be run again after the SP replacement.

Replacing a SP is seen as if disks were remoded from the system. Consequently, after the SP replacement, LUNs have to be re–mapped, and the system(s) must (usually and frequently) be rebooted.

EMC Symmetrix Disk Subsystems

All the Symmetrix models with front-end Fibre Channel directors are managed by S@N.IT!. This management includes:

- automatic discovery
- LUN Access Control management for S@N.IT! Agents
- monitoring (specific configuration required)
- launching of ECC from S@N.IT! GUI (if installed on the platform from which the S@N.IT! GUI has been launched).

SNMP Agent Configuration (Monitoring)

- Install and configure the Control Center Agent on a host. Refer to the *EMC Control Center Installation Guide* for details.
- Set the IP address of the Symmetrix subsystem in the S@N.IT! configuration as follows:
  - select the subsystem to be monitored in the S@N.IT! GUI,
  - From the Edit Properties menu, enter the IP address of the host where the Control Center has been configured.

Volume Logix

Compatibility between S@N.IT! LUN Access Control mechanism and Volume Logix is ensured, but the Information/Host LUNs tab of the S@N.IT! GUI shall be used (rather than the Information/Subsystem LUNs tab) to map/unmap LUNs to make sure that Volume Logix really allows the SAN host to access LUNs

ECC use

At least one gatekeeper LUN must be mapped on each host to be managed by ECC, before the poller can be launched.

Powerpath on S@N.IT! Agents

When a LUN mapping operation is performed from the GUI and if S@N.IT! discovers several paths between a S@N.IT! Agent and a Symmetrix or DMX array, then the GUI proposes to map the LUN for all paths. If it is not the expected behaviour, the administrator must explicitely perform the mapping for each concerned path.

On AIX S@N.IT! Agent hosts, the configuration method of Powerpath is automatically called after each LUN mapping/unmapping operation.

On Windows hosts, this operation has to be performed manually.

StoreWay Cost Effective Line Disk Subsystems

The Fibre Channel models are managed by S@N.IT!. This management includes:

- automatic discovery
- LUN Access Control management of S@N.IT! Agents.

FDA Disk Arrays

The following FDA arrays are supported:

- FDA 1300. 2300

This management includes:

- Automatic discovery
- Monitoring

Note: Automatic discovery and Monitoring features require that the SNMP agent of the FDA array is started and that its IP address is included into the ranges of IP addresses specified for out band discovery (see *Server Configuration* , on page 5-4) and reachable from the S@N.IT! Server.

- Launching of ISM client from S@N.IT! GUI (if ISM Client is installed on the platform where the S@N.IT! GUI has been started).
- LUN access control
- LUN masking configuration

For this feature to be operational, the following requirements must be met:

– The ISM Command Line Interface software must be installed on a S@N.IT! Agent.

– The SAN administrator must set FDA–specific properties using S@N.IT! GUI or CLI:

  . **FDAServer**: IP name of the platform where the ISM Server is installed.

  . **FDAUser**: user name to be used to connect to the ISM Server.

  . **FDAPassword**: password associated to this user within the ISM application**.**

  . **FDAClient**: name of the S@N.IT! Agent where the ISM Command Line Interface software is installed**.**

  . **FDALUNMasking**: must be set to `true`.

– The LUN access control feature for the array must be activated (using the **SetAclMode** CLI command).

– The syntax for the name of the LUNs sets is as follows: `<platform>:<set name>`, where :

  `platform = CX` (for Solaris), `LX` (for Linux), `AX` (for AIX) or `WN` (for Windows). See the *Configuration Setting Tool (GUI)* documentation (ref. 86A288EG) for more details.

  `set name` = 1 to 24 characters among the following ones: letters (A to Z or a to z), numbers (0 to 9), '–' '/'.

– The number of Physical LUN is a number between 0 and FFFF.

FRA Disk Arrays

The following FDA arrays are supported:

- FRA-1622

This management includes:

- Automatic discovery
- LUN access control
- Launching of FRA Storage Manager from S@N.IT! GUI (if FRA Storage Manager is installed on the machine where the GUI has been started).

Switches

Brocade Switches

The following models are managed by S@N.IT!:

- 2040, 2400, 2800, 2850, 3800, 3850, 3900, 12000

The management includes:

- automatic discovery,
- monitoring,
- launching of management application from S@N.IT! GUI (telnet or WebBrowser),
- discovery of zoning configuration.

For these features to be operational, the switch must have a configured IP address and name, and its ethernet port must be reachable by the S@N.IT! Server. The firmware level must be higher than 2.2.

Connectrix Switches

The following model is managed by S@N.IT!:

- MACDATA

The management includes:

- automatic discovery,
- monitoring,
- launching of management application from S@N.IT! GUI (telnet).

For these features to be operational, the switch must have a configured IP address and name, and its ethernet port must be reachable by the S@N.IT! Server.

NovaScale Blade Switches

The switches embedded in the NovaScale Blade servers are supported.

The management includes:

- automatic discovery,
- monitoring.

For these features to be operational, the SNMP agent of the switch must be configured properly and the S@N.IT! Server must have a TCP/IP access to the switch.

FC/SCSI Bridges

The following models are managed by S@N.IT!:

- Crossroads CP4150
- Crossroads CP4250

The management includes:

- automatic discovery
- monitoring of the bridge and of the SCSI devices connected to it.

Libraries

StorageTek libraries

The following models are managed by S@N.IT!:

- StorageTek L–Series

The management includes:

- Automatic discovery of drives and robot
- Monitoring
- Launching the Web Tools

For these features to be operationnal the SNMP agent of the library must be started, its IP address must be included into the ranges of IP addresses specified for out band discovery (see *Server Configuration* , on page 5-4) and it must be accessible from the S@N.IT! Server. If this is not the case discovery will be partial and monitoring will not be operational.

Overland libraries

The following models are managed by S@N.IT!

- Overland Neo Series
- Overland LXN2000, LXN4000 and LXN8000

The management includes:

- Automatic discovery of drives and robot
- Monitoring (Overland Neo Series only)
- Launching the Web Tools (Overland Neo Series only)

For these features to be operationnal the SNMP agent of the library must be started, its IP address must be included in the ranges of IP addresses specified for out band discovery (see *Server Configuration* , on page 5-4) and it must be accessible from the S@N.IT! Server. If this is not the case discovery will be partial and monitoring will not be operational.

Media Server Virtuo

S@N.IT! supports the Bull Media Server Virtuo version V4 or higher.

The management includes:

- automatic discovery of virtual tape drives and virtual librairies.

Applications

The applications must be added manually (see *Edit /Create Component*, on page 6-26).

The management includes:

- launching the configuration tools of the application.

For this feature to be operational, the Application component must be created with **Model** property set to:

- **NSMaster** for Bull NovaScale Master application
- **OpenSave** for Bull OpenSave backup application
- **NetBackup** for Veritas Netbackup application
- **NetWorker** for EMC NetWorker application.

Non Supported Subsystem or Library

A non supported disk array or library must be added to a SAN configuration after S@N.IT! software installation. The procedure to follow is described in *Adding a Non Supported Subsystem or Library*, on page 9-2.

Chapter 12. Supported Platforms

This chapter provides specific information (prerequisites, installation procedures, configuration, LUN access control operation) for the following supported platforms:

AIX Platforms

S@N.IT! Features

All S@N.IT! features are supported on AIX platforms.

AIX Installation of S@N.IT! Limited Edition

Delivery

The AIX S@N.IT! Limited Edition software is delivered in the *S@N.IT! Limited Edition* CD-ROM as two LPPs :

san_it.devices describes the supported SAN devices,

san_it.all contains binaries, commands and configuration files.

Both filesets must be installed on all AIX servers.

Prerequisites

Hardware requirements

– Escala platforms with at least:

  . 128 Mbytes memory
  . a TCP/IP connection
  . Bull Fibre Channel adapters.

Software requirements

– AIX 5.1 or later

– Java14.sdk version 1.4.0.0 or later

– Fibre Channel driver (provided by Bull) on S@N.IT! Agents.

– To manage the disk subsystems, additional software (such as PowerPath) may be required and installed at any time.

For more information about the requirements, refer to the *SRB (System Release Bulletin) for AIX and Bull Enhancement* corresponding to the version installed on the system.

Software Installation

To install S@N.IT! Limited Edition on AIX hosts, perform these steps:

1. From the shell (or dtterm window), login as **root**.
2. Start SMIT as follows:

`smit [-C] install_latest` or `smitty install_latest`

Note: `smit` starts SMIT in graphic mode while `smit -C` and `smitty` start SMIT in ASCII mode.

3. From the System Management menu, select the following sequence of sub–menus

```
Software Installation and Maintenance
Install / Update Software
Install and Update from LATEST Available Software
```

4. A window opens that displays the installation parameters; select the parameters as follows (to choose the software to install, click on F4):

```
INPUT device / directory for software                /dev/cd0
SOFTWARE to install:                                 all_latest
PREVIEW only? (install operat. will NOT occur)       no
COMMIT software updates?                             yes
SAVE replaced files?                                 no
AUTOMATICALLY install requisite software?            yes
EXTEND file systems if space needed                  yes
OVERWRITE same or newer versions?                    no
VERIFY install and check file sizes?                 yes (Note: default = no)
Include corresponding LANGUAGE filesets?             yes
Detailed output?                                     yes (Note: default = no)
Process multiple volumes?                            yes
```

Note: `Detailed output=yes` to set verbose mode for detailed audit log.

5. Validate.
6. Confirm the message "`Are you sure?`"

   The installation sequence is completed automatically. An installation log is created and displayed progressively on the local monitor screen. This installation log includes the following items:

   – list of all installed files and directory paths,

   – text of the License Agreement,

   – preliminary installation check confirmation,

   – installation summary, showing for each installed software:

   . Name (Product Name)
   . Revision Level (n.n.n.n.), where n represents any number
   . User (User or Root)
   . Event (Application)
   . Result (Success).

   An installation trace is saved in a dated log filed under /**smit.log**.

7. If the AIX platform uses SAN LUNs as boot device, the LUNs associated to rootvg have been mapped during installation. Check this mapping by running the /**usr**/**sanit**/**bin**/**san_info** command before rebooting.
8. Reboot the AIX server to make the LUN Access Control ACTIVE

   If the host was already connected to the SAN, this operation will remove all the hdisk devices corresponding to SAN LUNs except for:

   – those that had a PVID: they remain defined,

   – those belonging to storage devices not supported by S@N.IT! (see Chapter 11. *Supported SAN Components*).

Configuration files are delivered under the /**etc**/**sanit** directory. This directory will also contain the data of the S@N.IT! Server.

Executable files (commands and Jar) are delivered under the /**usr**/**sanit**/ directory.

A `san0` device entry is created in CuDv ODM database.

The default trace directory, /**var**/**tmp**/**sanit**, is created.

During a first installation, the **sanit.cfg** configuration file is created with **ServerRole** and **AgentRole** parameters set to `true` and Server parameters set to `127.0.0.1` (IP loopback address); this means that the AIX host is configured as both a S@N.IT! Agent and a S@N.IT! Server. If this is not the desired behaviour, launch the S@N.IT! GUI on the local host and update the local configuration (see Chapter 5. *Configuration and Launching*).

Software De-installation

The S@N.IT! Limited Edition software can be de–installed by removing the **san_it.all** and **san_it.devices** filesets. But this is possible only when the LUN Access Control mechanism is de–activated (see *LUN Access Control mechanism*, on page 2-6).

Warning: removing the S@N.IT! software will also remove all the LUN Access Control configuration (LUN mappings) for the current AIX host and hence this operation must not be performed while the host is connected to the SAN.

AIX Installation of S@N.IT! Data Center

Delivery

The AIX *S@N.IT! Data Center* software is delivered within the *S@N.IT! Data Center* CD-ROM as a unique LPP (**san_it.full**). This software must be installed on the S@N.IT! Server only.

Prerequisites

The AIX S@N.IT! Limited Edition software must be installed before S@N.IT! Data Center.

Software Installation

Installation is performed via SMIT and is identical to S@N.IT! Limited Edition installation, described on page 12-2.

The S@N.IT! services are automatically stopped and restarted with the current configuration. This configuration can then be updated if required, by launching the S@N.IT! GUI on the local AIX server (see Chapter 5. *Configuration and Launching*).

Software De-installation

The S@N.IT! Data Center software can be de–installed by removing the **san_it.full** LPP. This operation does not de–install the S@N.IT! Limited Edition software, but it stops the S@N.IT! services, which must be restarted using the /**etc**/**rc.sanit** command.

Configuration

Selecting the Java Version

S@N.IT! software requires Java run time 1.4. If this is not the default Java version for a host, change it using the **JavaPath** configuration parameter.

Example:

```
JavaPath=/usr/java14/jre/bin/java
```

AIX User Provided Commands

The commands provided by the S@N.IT! administrator can be:

- Executable files(binaries)
- Shell scripts: in that case the name of the shell script must be preceded by the path of the script itself, unless the first line of the script contains the shell path (“#!/bin/sh”).

The S@N.IT! services launch these commands as daemons so they cannot be interactive.

Examples:

```
UserNotificationCommand=/bin/ksh /etc/sanit/NotifyCommand.template
UserPostMappingCommand=/mypath/myprogram
```

Error logging on AIX S@N.IT! Server

If the **EnableSystemErrorLog** configuration parameter of the AIX S@N.IT! Server is set to true, a message will be recorded in the errlog each time a component is detected as faulty.

LUN Access Control

Mechanism

- After initial installation of the S@N.IT! software, the LUN Access Control is set to ACTIVATED state. After a reboot, the LUN Access Control turns to ACTIVE. This means that AIX sees only the LUNs that are explicitly mapped: hdisk devices corresponding to non-mapped LUNs are removed, unless a physical volume was configured for them (in that case, the hdisk remains in `defined` state).
- If the LUN Access Control is de–activated for some reason, it turns to INACTIVE state. This operation must be followed by a **cfgmgr** command allowing AIX to create hdisk devices for all SAN LUNs.
- The mapping/unmapping operations must be performed while the LUN Access Control is ACTIVE. If mapping/unmapping is performed while LUN Access Control is INACTIVE or ACTIVABLE, there will be no impact on the system behaviour until LUN Access Control is turned to ACTIVE.

Important: the access to storage devices that are not supported by S@N.IT! is not modified by the use of S@N.IT! LUN Access Control mechanism: the LUNs of these devices remain available for the operating system and cannot be hidden by S@N.IT!. Refer to Chapter 11. *Supported SAN components* for adding a storage device not supported by S@N.IT!.

Mapping LUNs

After a LUN mapping operation a hdisk device is created for each newly mapped LUN.

Un-mapping LUNs

The LUNs to be un-mapped must not be used by AIX. So before un-mapping a LUN the volume groups that use this LUN must be un-mounted.

The un-mapping operation removes the corresponding hdisk device, unless a physical volume was configured on it (in that case the hdisk device remains in **defined** state).

Miscellaneous

S@N.IT! commands path

command	directory
sanit	/usr/bin
rc.sanit	/etc
san_info, san_snap, san_activate, san_deactivate, san_map_lun,san_unmap_lun	/usr/sanit/bin

Management of SAN configuration changes

When a new SAN component (storage device, switch) is connected to the SAN, or when a SAN component is removed (disconnected) from the SAN, it is necessary to restart the fibre channel driver configuration. This requires to run the following command:

```
cfgmgr
```

Windows Platforms

S@N.IT! Features

All S@N.IT! features are supported on Windows 2003 and Windows 2000 IA32 platforms.

Depending on the platform and fibre channel driver, all S@N.IT! Agent features may not be available. The current status is summarized in the table below (check for updates with your Bull representative):

Platform	Windows	FC HBA	Driver	Discovery	Host based LUN mapping
NovaScale Blade	2003 IA32	Qlogic	mini port driver	Supported	Supported
Express 5800	2000	Emulex	SCSI port driver	Supported	Supported
Express 5800	2003 IA32	Emulex	SCSI port driver	Supported	Supported
Express 5800	2003 IA32	Emulex	mini port driver	Supported	Not supported

Notes:

1. StorPort driver types are not supported in this S@N.IT! delivery.
2. Windows 2003 Enterprise Edition IA64 is not supported in this S@N.IT! delivery.

Windows Installation of S@N.IT! Limited Edition

Delivery

The Windows S@N.IT! Limited Edition software is delivered in the *S@N.IT! Limited Edition* CD-ROM as a single self–installable binary (**SanITSetup.exe**).

Prerequisites

Hardware requirements

– NovaScale Blade (IA32) or Express 5800 with at least:
  . 548 MHz processor,
  . 256 Mbytes memory,
  . a TCP/IP connection.

– For S@N.IT! Agents:

  Important: all HBAs must have the same driver type. It means that all drivers for HBA fibre installed on the Agent Host must be either Port driver or miniPort driver. When mixed drivers are found, both S@N.IT! discovery and LUN Access Control functions are not operational.

  . Emulex Port driver version 5.2.21a8 for all HBA models.

    Warning: Do not install Emulex HBAs Port driver version 5.2.22a8, which leads to a blue screen in some cases.

  . Emulex MiniPort driver version 5.5.10a10 (with adjunct driver) for all HBA models.

  . QLogic MiniPort driver 9.0.1.10 for HBA 23xx family only.

  Note: StorPort driver types are not supported in this S@N.IT! delivery.

Software requirements

– Windows 2000 service pack 4 or higher,

– Windows 2003 Enterprise Edition IA32 service pack 1 build 1218 or higher,

– Windows TCP/IP services,

– Java Runtime version 1.4.2 or higher ,

Note: Windows 2003 Enterprise Edition IA64 is not supported in this S@N.IT! delivery.

Prerequisite Software Installation

The Emulex or Qlogic drivers must be installed prior to S@N.IT! software.

Important: S@N.IT! LUN Access Control (and LUN Mapping) is adapter–based through Emulex or QLogic internal APIs and is global to all adapters. Consequently, once S@N.IT! software is installed, the configuration tools for Emulex (elxcfg and ntutilnt) or QLogic (SANsurfer FC HBA Manager or scli) must not be used for configuring both LUN Access Control and LUN Mapping operations. Impredictable results may occur if one of these tools is used.

Special notes regarding Emulex Port Priver Software Installation

– **Important:** Be especially careful when setting the driver option:

  – Select the Arbitrated Loop or Fabric entry according to the Fibre Channel connection mode of the component physically connected to the related HBA.

  – With the previous selection, choose the Automap option, if it exists.

– The “with LUN mapping“ option of the Emulex driver from the Express 5800 “Storage Fibre Host Connection Kits for Windows“ is compatible with the LUN Access Control managed by S@N.IT!. For more information, please refer to the documentation provided with this Connection kit.

Java Runtime Environment Installation

Java Runtime Environment must be installed prior to S@N.IT! software. If it is not already installed, perform the procedure below:

1. Log on the Windows host using an **administrator** account.
2. Insert the *S@N.IT! Limited Edition* CD-ROM.
3. Start Windows Explorer and select the **Windows32\java** directory from the CD-ROM drive.
4. Double click on the **j2re–1_x_y_zz–win–i.exe** file and follow the instructions.

Software Installation

Warning: for Windows hosts that connect DAS subsystems, refer to the general rules given in *Installation rules on a Windows S@N.IT! Agent connected to a DAS subsystem*, on page 11-4.

To install S@N.IT! Limited Edition on Windows hosts, perform these steps:

1. Log in the Windows host using an **administrator** account.
2. Insert the *S@N.IT! Limited Edition* CD-ROM.
3. Start Windows Explorer and select the Windows32 directory from the CD-ROM drive.
4. Double click on the **SanITSetup.exe** file and follow the provided instructions. In particular take care to the recommendations that follow.

   – Correctly select the role of the platform: “S@N.IT! Server” is the role for the S@N.IT! Server platform, and “S@N.IT! Agent” is the role for a S@N.IT! Agent platform.

   – Enter the network name of the S@N.IT! Server platform in the **S@N.IT! Server** field if the platform is not dedicated to be a S@N.IT! Server.

   – Enter the **Domain Name**, if any, where the Service Account user will be created:

     . if the platform does not participate in a domain, leave this field blank,

     . if the platform participates in a Windows 200x domain enter the pre–Windows 2000 domain name. Generally this name is the prefix of the global Windows domain name: first name before the first period (.) in upper case.

     For example if the Windows 200x domain is `mysan.myinc.com`, the pre–Windows 2000 domain name is: `MYSAN`.

To check this name go to the domain controller, run Start/Programs/Administrative Tools/Active Directory Domains and Trust, ask for the properties of the domain in which your platform participates and get the pre–Windows 2000 domain name.

Depending on the administration policies and if you own the privileges, even if the platform participates in a domain, you can leave the **Domain Name** field blank. Then the Service Account user will be created locally on your platform.

– Enter the Service Account user name. This user name will be used to start the S@N.IT Scheduler service on the platform.

– Enter the Password of the Service Account user. The default password is **S@N.it!**.

The screens below show the parameters that must be specified:

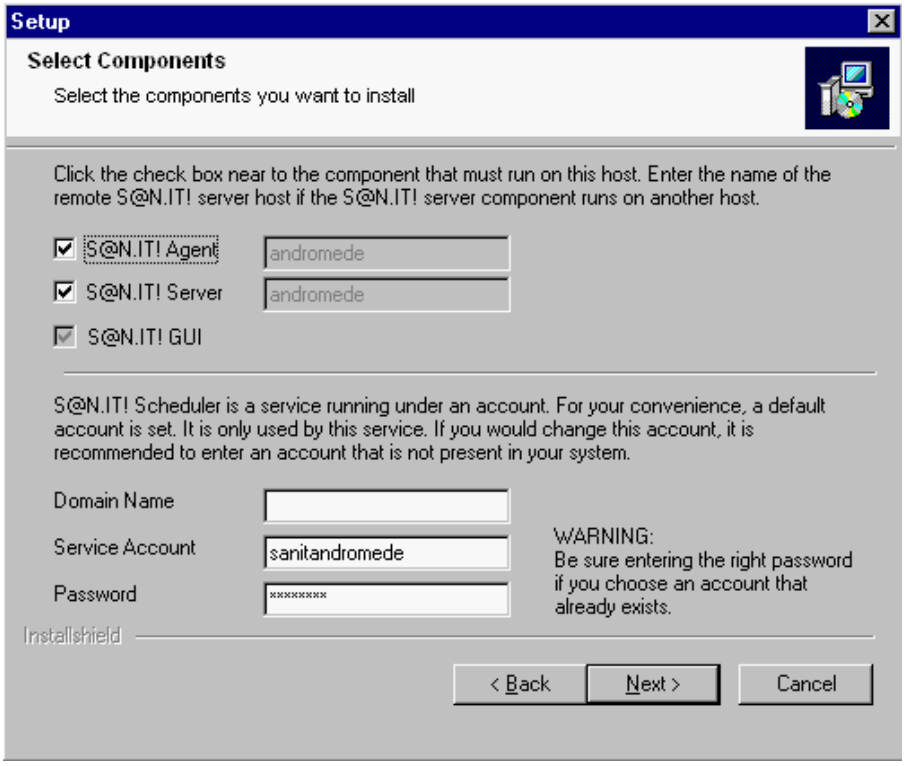

Figure 33. Windows Installation: select components

Click on “Next” button.

5. Reboot the Windows host.

The default installation directory is:

```
<%SystemDrive%:\Program Files\Bull\S@N.IT\>
```

What happens during installation:

- S@N.IT! configuration files are updated. This configuration can be updated by launching the S@N.IT! GUI on the local machine (see Chapter 5. *Configuration and Launching*).
- A Windows user is created (to run the S@N.IT! service) according to the input name.
- The **S@N.ITScheduler** Windows service is created and started under the previously created user environment.
- A shortcut to S@N.IT! GUI is added to the Start menu.
- The LUN Access Control is activated on the current S@N.IT! Agent platforms and:
  – all LUNs of supported disk subsystem models are made inaccessible. LUNs have to be explicitly mapped through the S@N.IT! GUI,

– all LUNs of non supported but identified disk subsystem models are made accessible; S@N.IT! automatically maps these LUNs,

– all LUNs of non–disk subsystems devices are made accessible; S@N.IT! automatically maps these LUNs.

Software De-installation

To de-install S@N.IT! Limited Edition on Windows hosts, perform these steps:

1. Log in the S@N.IT! platform using the same system administrator account than the one used for installation.
2. If the S@N.IT! is upgraded, go to step 3.

   If the S@N.IT! is definitively removed, perform these actions:

   – un–map all LUNs now,

   – de–activate LUN Access Control.
3. Remove the S@N.IT! Base program using the standard Windows de–installation method:

   Start>Settings>Control Panel

   Add/Remove Programs
4. Log off from the system **administrator** account.

What happens during de–installation:

- The LUN Access Control is not modified on the current S@N.IT! Agent platform and the mapping of LUNs stays as is.
- The S@N.IT! scheduler Windows service is stopped and removed and the Windows user (for this service) is removed.
- The shortcut is removed from the Start menu.
- All files that have been installed are removed (but not the files dynamically created by S@N.IT!).
- The installation directory is not removed.

Windows Installation of S@N.IT! Data Center

Delivery

The Windows *S@N.IT! Data Center* software is delivered within the *S@N.IT! Data Center* CD-ROM as a single self–installable binary (**SanITSetupFull.exe**). This software must be installed on the S@N.IT! Server only.

Prerequisites

The Windows *S@N.IT! Limited Edition* software must be installed before *S@N.IT! Data Center* software.

Software Installation

1. Log in the Windows host using an **administrator** account.
2. Insert the *S@N.IT! Data Center* CD-ROM.
3. Start “Windows Explorer” and select the **Windows32** directory from the CD-ROM drive.
4. Double click on the **SanITSetupFull.exe** file and follow the instructions provided.
5. The following screen is opened. Fill in the IP addresses.

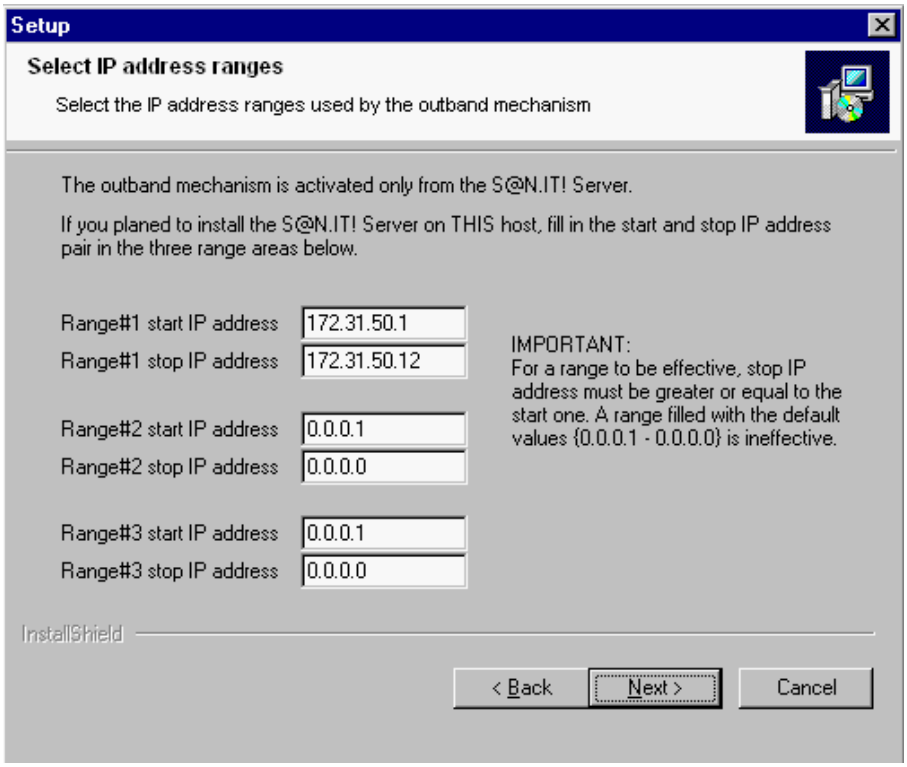

Figure 34. Windows Installation: select IP addresses

The **S@N.ITScheduler** Windows service is automatically stopped and restarted with the current configuration. This configuration can then be updated (if required) by launching the S@N.IT! GUI on the local Windows server (see Chapter 5. *Configuration and Launching*).

Software De-installation

The S@N.IT! Data Center software can be de–installed by removing the **S@N.IT! Full** windows program.

1. Log in the S@N.IT! platform using the same system **administrator** account as the one used for installation.
2. Remove the **S@N.IT! Full** program using the standard Windows un–installation method:

   Start>Settings>Control Panel
   Add/Remove Programs
3. Log off from the system **administrator** account.

This operation does not de–install the S@N.IT! Limited Edition software, but it stops the **S@N.ITScheduler** Windows service which must then be restarted.

Configuration

Selecting the Java version

S@N.IT! software requires Java run time 1.4.2 or higher. If this none of these is the newest Java version for a host, this can be changed using the **JavaPath** configuration parameter to the name of the directory that contains the right java version.

Example:

```
JavaPath=C:\Program Files\JavaSoft\JRE\1.4.2_05\bin
```

Windows User provided commands

The commands provided by the S@N.IT! administrator can be:

- Executable files(binaries)

- Windows scripts (**wscript** – recommended on Windows 200x, or **.bat** files)

The S@N.IT! services launch these commands as daemons so they cannot be interactive.

The command path must be provided during the configuration of the S@N.IT! software according to the following rules:

- "\" characters must be replaced by "/"
- Paths containing space characters must be surrounded by double quotes (").
- Wscript Windows scripts must be preceded by the path of the **wscript** command itself.

Windows wscript examples:

```
UserNotificationCommand=C:/WINNT/system32/wscript C:/mypath/myScript.js

UserNotificationCommand=C:/WINNT/system32/wscript "C:/Program
Files/myScript.js"
```

Windows batch file examples:

```
UserNotificationCommand=C:/mypath/myCommands.bat

UserNotificationCommand="C:/my other path/myCommands.bat"
```

Windows executable examples:

```
UserNotificationCommand=C:/mypath/myprogram.exe

UserNotificationCommand="C:/my other path/myprogram.exe"
```

Error logging on Windows S@N.IT! Server

If the **EnableSystemErrorLog** configuration parameter of the Windows S@N.IT! Server is set to true, a message will be recorded in the Event Viewer Application log file, each time a component is detected as faulty.

LUN Access Control

Important:

- Use exclusively S@N.IT! to perform LUN Access Control. Do not use Emulex (elxcfg and ntutilnt) and QLogic (SANsurfer FC HBA Manager or scli) configuration tools.
- In the QLogic environment, both LUN Access Control and LUN mapping operations apply only to devices that are connected to the SAN and working properly.

Mechanism

- The ACTIVABLE state is not supported. After initial installation of the S@N.IT! software, the host must be rebooted for the LUN Access Control state to become ACTIVE. Then the Windows host sees only the LUNs that are explicitly mapped and the LUNs of not supported storage devices.
- If an INCONSISTENT state is detected during the S@N.IT! installation process, the system tries to set the consistent ACTIVE state and the installation terminates successfully.
- If an INCONSISTENT state is detected when S@N.IT! runs, the INCONSISTENT state is set for the related Host (and made visible through S@N.IT! GUI). In this case perform an "Activate LUN Access Control" operation to recover the consistent ACTIVE state.

  Important: In the QLogic environment, all LUNs are unmapped during the "Deactivate LUN Access Control" operation.
- The LUN mapping/unmapping operations must be performed while the LUN Access Control state is ACTIVE.

Mapping LUNs

After a LUN mapping operation:

- In a QLogic driver environment, reboot the Host.

- In an Emulex driver environment, you have nothing to do. After a while the newly mapped LUNs become automatically known from the operating system.

Un-mapping LUN

Before a LUN un-mapping operation:

- On Windows 2000 and Windows 2003 hosts, disable the related disk devices.

After the un-mapping operation:

- In a QLogic driver environment, reboot the Host.
- In an Emulex driver environment, you have nothing to do. After a while the newly un–mapped LUNs become automatically unknown from the operating system.

Miscellaneous

S@N.IT! commands path

command	directory
sanit	<%installation directory%>\bin
san_info, san_snap, san_activate, san_deactivate, san_map_lun,san_unmap_lun	<%installation directory%>\bin

Management of SAN configuration changes

- When a new SAN component (storage device, switch) is connected to, or disconnected from the SAN, no operation is required on the Windows hosts.
- When a new tape library or a non supported storage subsystem is connected to the SAN, the LUNs for this new SAN component must be explicitely mapped, as described below.

  On each Windows S@N.IT! Agent, perform the following operations:

  1. The LUN Access Control must be active.
  2. Open a Start>Programs>Command Prompt window.
  3. Enter the **\bin** directory in the S@N.IT! installation directory (default: **%SystemDrive%:\Program Files\Bull\S@N.IT**).
  4. Type the following command:

     ```
     ConfigAllUDLUN -C
     ```

  5. Depending on the result of the command, the operation is different:

     . When no un–mapped LUNs of non–supported SAN component is found, the following message is displayed:

     ```
     >No unknown device LUN to map
     ```

     . When un–mapped LUNs of non–supported SAN component are found, a message like the following one is displayed (example for an Overland LXB library):

     ```
     >Following unknown device LUN are mapped through 'Emulex LP-8000
     Adapter 0'>Wwn(100000E00200180E):
     LUN 0^QUANTUM^^DLT7000|
     LUN 1^QUANTUM^^DLT7000|
     LUN 2^OVERLAND^^LXB
     IMPORTANT:
         REBOOT the host andromede NOW to complete this command.
     ```

  6. Exit the Command Prompt window to complete the procedure
  7. If LUNs have been mapped:

     . In a QLogic driver environment, reboot the Host.
     . In an Emulex driver environment, you have nothing to do. After a while the newly mapped LUNs become automatically known from the operating system.

Linux Platforms

S@N.IT! features

S@N.IT! LUN access control mechanism is not supported on Linux S@N.IT! Agents. It remains in INACTIVE state.

Linux Installation of S@N.IT! Limited Edition

Installation on Intel 32–bit Platforms

Delivery

The Linux *S@N.IT! Limited Edition* software is delivered in the *S@N.IT! Limited Edition* CD-ROM as two RPM packages for Intel 32–bit:

- **san_it.devices–<*v.r.m–ff*>.Linux.i386.rpm** contains the supported SAN devices
- **san_it.all–<*v.r.m–ff*>.Linux.i386.rpm** contains binaries, commands and configuration files.

Where ***v.r.m*** is the version, and ***ff*** is the release.

Prerequisites

Hardware requirements:

– Intel platforms with 32–bit architecture and a TCP/IP connection

– Emulex fibre channel adapters or Qlogic QL23xx Fibre Channel adapters (for S@N.IT! Agents).

Software requirements:

– RedHat AS 2.1 or AS 3.0 or SuSe SLES 8.0

– Emulex lpfc RPM package (available in the *Fibre Channel non-AIX drivers* Bull CD-ROM, or from Emulex Web site) or QLogic HBA API (available in the *S@N.IT! Limited Edition* CD–ROM).

– Java run time 1.4.2 (delivered in the *S@N.IT! Limited Edition* CD–ROM).

Note: refer to SRB for up to date pre–requisites information.

Prerequisites Software Installation

Emulex lpfc RPM

Refer to Emulex documentation for the installation and configuration of the **lpfcdd** driver.

Once the driver and its HBA API are installed, check the HBA API configuration file (/**etc/hba.conf**). It should only contain one non–commented line with the following information:

```
com.emulex.emulexapilibrary /usr/lib/libemulexhbaapi.so
```

Qlogic HBA API

1. Insert the *S@N.IT! Limited Edition* CD–ROM in the CD–ROM drive.
2. Copy the contents of the /Linux32/qlogic directory (libinstall, libremove, qlapi–v2.01beta6–rel.tgz) to a directory (say `/tmp/qlinstall`) on the Linux host.
3. Go to this directory:

   ```
   cd /tmp/qlinstall
   ```

4. Launch the libinstall command:

   ```
   ./libinstall
   ```

Once the driver and its HBA API are installed, check the HBA API configuration file (**/etc/hba.conf**). It should only contain one non–commented line with the following information:

```
qla2x00          /usr/lib/libqlsdm.so
```

Java run time

To install the Java Run Time on Linux hosts, perform these steps:

1. Login as **root**.
2. Insert the *S@N.IT! Limited Edition* CD-ROM in the CD-ROM drive.

   The CD-ROM filesystem is automatically mounted to one of the following directories:

   /mnt/cdrom (RedHat and Advanced Server distributions)

   /media/cdrom (SuSe distribution).
3. Run the following commands:
   a. `cd /home/users`
   b. For RedHat and Advanced Server distributions, run:

      ```
      sh /mnt/cdrom/Linux32/java/j2re-1_4_2_06-linux-i586-rpm.bin
      ```

      For SuSe distribution, run:

      ```
      sh /media/cdrom/Linux32/java/j2re-1_4_2_06-linux-i586-rpm.bin
      ```

   c. `rpm -Uvh j2re-1_4_2_06-fcs-linux-i586.rpm`
4. After the installation of the S@N.IT! software, you must configure the **JavaPath** parameter to the correct value (see *Selecting the Java version*, on page 12-16).
5. Take out the CD-ROM using the `eject` command.

Software Installation

To install S@N.IT! Limited Edition on Linux 32–bit hosts, perform these steps:

1. Login as **root**
2. Insert the *S@N.IT! Limited Edition* CD-ROM into the CD-ROM drive.

   The CDROM filesystem is automatically mounted to one of the following directories:

   /mnt/cdrom (RedHat and Advanced Server distributions)

   /media/cdrom (SuSe distribution)
3. Run the following commands:

   For RedHat Advanced Server distributions:

```
rpm -Uvh /mnt/cdrom/Linux32/san_it.devices-6.0.1-00.Linux.i386.rpm
rpm -Uvh /mnt/cdrom/Linux32/san_it.all-6.0.1-00.Linux.i386.rpm
```

   For SuSe Distribution:

```
rpm -Uvh /media/cdrom/Linux32/san_it.devices-6.0.1-00.Linux.i386.rpm
rpm -Uvh /media/cdrom/Linux32/san_it.all-6.0.1-00.Linux.i386.rpm
```

4. Take out the CD-ROM using the `eject` command.
5. Verify that the installation is successful with the following command:

```
rpm -qa | grep san_it
```

What happens during installation:

- Configuration files are delivered in the **/etc/sanit** directory. This directory will also contain the data of the S@N.IT! Server.
- Executable files (commands and jar) are delivered in the **/usr/sanit** directory.

- The default trace directory **(/var/tmp/sanit**) is created.
- During a first installation the **sanit.cfg** file is created with **ServerRole** and **AgentRole** set to true and **Server** parameters set to 127.0.0.1 (IP loopback address); this means that the Linux host is configured as both a S@N.IT! Agent and a S@N.IT! Server. If this is not the desired behaviour, launch the S@N.IT! GUI on the local host and update the local configuration (see Chapter 5. Configuration and Launching).

Installation on 64–bit Platforms

Delivery

The Linux *S@N.IT! Limited Edition* software is delivered in the *S@N.IT! Limited Edition* CD-ROM as two RPM packages for Intel 64–bit (directory: /**Linux64**):

- **san_it.devices–<*v.r.m–ff*>.Linux.ia64.rpm** contains the supported SAN devices
- **san_it.all–<*v.r.m–ff*>.Linux.ia64.rpm** contains binaries, commands and configuration files.

Where ***v.r.m*** is the version, and ***ff*** is the release.

Prerequisites

Hardware requirements:

– Bull NovaScale platforms

– Emulex or Qlogic fibre channel adapters.

Software requirements:

– RedHat AS 2.1 or 3.0, or SuSe SLES 8.0

– For SAN agents (server connected to the SAN): Emulex lpfc RPM package (including support for SNIA's HBA API), or Qlogic HBA API (available in the *S@N.IT! Limited Edition* CD–ROM).

– Java run time 1.4.2 (delivered in the *S@N.IT! Limited Edition* CD–ROM).

Note: refer to SRB for up to date pre–requisites information.

Prerequisites Software Installation

Emulex lpfc RPM:

Refer to Emulex documentation for the installation and configuration of the **lpfcdd** driver. Check the /**etc**/**hba.conf** file. It should contain one non–commented line with the following information:

```
com.emulex.emulexapilibrary /usr/lib/libemulexhbaapi.so
```

Qlogic HBA API

1. Insert the *S@N.IT! Limited Edition* CD–ROM in the CD–ROM drive.
2. Copy the contents of the /Linux64/qlogic directory (libinstall, libremove, qlapi–v2.01beta6–rel.tgz) to a directory (say `/tmp/qlinstall`) on the Linux host.
3. Go to this directory:

   ```
   cd /tmp/qlinstall
   ```

4. Launch the libinstall command:

   ```
   ./libinstall
   ```

Once the driver and its HBA API are installed, check the HBA API configuration file (/**etc**/**hba.conf**). It should only contain one non–commented line with the following information:

```
qla2x00          /usr/lib/libqlsdm.so
```

Java run time

Java run time 1.4.2 (delivered in *S@N.IT! Limited Edition* CD–ROM).

Software Installation

To install S@N.IT! Limited Edition on Linux 64–bit hosts, perform these steps:

1. Login as **root**
2. Insert the *S@N.IT! Limited Edition* CD-ROM into the CD-ROM drive.

   The CDROM filesystem is automatically mounted to one of the following directories:

   /**mnt**/**cdrom** (RedHat and Advanced Server distributions)

   /**media**/**cdrom** (SuSe distribution)
3. Run the following commands:

   For RedHat and Advanced Server distributions:

```
rpm –Uvh /mnt/cdrom/Linux64/san_it.devices–6.0.1–00.Linux.i386.rpm
rpm –Uvh /mnt/cdrom/Linux64/san_it.all–6.0.1–00.Linux.i386.rpm
```

   For SuSe Distribution:

```
rpm –Uvh /media/cdrom/Linux64/san_it.devices–6.0.1–00.Linux.i386.rpm
rpm –Uvh /media/cdrom/Linux64/san_it.all–6.0.1–00.Linux.i386.rpm
```

4. Take out the CD-ROM using the `eject` command.
5. Verify that the installation is successful with the following command:

```
rpm –qa | grep san_it
```

What happens during installation:

- Configuration files are delivered in the /**etc**/**sanit** directory. This directory will also contain the data of the S@N.IT! Server.
- Executable files (commands and jar) are delivered in the /**usr**/**sanit** directory.
- The default trace directory **(**/**var**/**tmp**/**sanit**) is created.
- During a first installation the **sanit.cfg** file is created with **ServerRole** and **AgentRole** set to true and **Server** parameters set to 127.0.0.1 (IP loopback address); this means that the Linux host is configured as both a S@N.IT! Agent and a S@N.IT! Server. If this is not the desired behaviour, launch the S@N.IT! GUI on the local host and update the local configuration (see Chapter 5. *Configuration and Launching*).

Software De–installation on Linux 32–bit and 64–bit Platforms

To de–install S@N.IT! Limited Edition on Linux hosts, perform these steps:

1. Login as **root**
2. Launch the following commands:

```
rpm –e san_it.all
rpm –e san_it.devices
```

Linux Installation of S@N.IT! Data Center

Delivery

The Linux *S@N.IT! Data Center* software needs to be installed only on S@N.IT! Server. It is delivered in the *S@N.IT! Data Center* CD–ROM as one RPM package:

- **san_it.full–<*v.r.m–ff*>.Linux.i386.rpm** for Linux 32–bit platforms
- **san_it.full–<*v.r.m–ff*>.Linux.ia64.rpm** for Linux 64–bit platforms

Where ***v.r.m*** is the version, and ***ff*** is the release.

Prerequisites

The Linux *S@N.IT! Limited Edition* software must be installed before the *S@N.IT! Data Center* software.

Software Installation

To install S@N.IT! Data Center on Linux hosts, perform these steps:

1. Login as **root**
2. Insert the *S@N.IT! Data Center* CD-ROM into the CD-ROM drive.

   The CD-ROM filesystem is automatically mounted to one of the following directories:

   /**mnt**/**cdrom** (RedHat and Advanced Server distributions)

   /**media**/**cdrom** (SuSe distribution)
3. Run one of the following commands:
   - For RedHat and Advanced Server distributions on Linux 32–bit platfroms:

```
rpm –Uvh /mnt/cdrom/Linux32/san_it.full-6.0.1-00.Linux.i386.rpm
```

   - For SuSe Distribution on Linux 32–bit platfroms:

```
rpm –Uvh /media/cdrom/Linux32/san_it.full-6.0.1-00.Linux.i386.rpm
```

   - For RedHat and Advanced Server distributions on Linux 64–bit platfroms:

```
rpm –Uvh /mnt/cdrom/Linux64/san_it.full-6.0.1-00.Linux.i386.rpm
```

   - For SuSe Distribution on Linux 64–bit platfroms:

```
rpm –Uvh /media/cdrom/Linux64/san_it.full-6.0.1-00.Linux.i386.rpm
```
4. Take out the CD-ROM using the `eject` command.
5. Verify that the installation is successful with the following command:

```
rpm –qa | grep san_it.full
```

What happens during installation:

- The S@N.IT! services are automatically stopped and restarted with the current configuration. This configuration can then be updated if required, by launching the S@N.IT! GUI on the local Server (see Chapter 5. *Configuration and Launching*).

Software De–installation

To de–install S@N.IT! Data Center on Linux hosts, perform these steps:

1. Login as **root**
2. Launch the following commands:

```
rpm –e san_it.full
```

Configuration

Selecting the Java version

S@N.IT! software requires Java run time 1.4.2. If this is not the default Java version for a host, change it using the **JavaPath** configuration parameter.

Example:

```
JavaPath=/usr/java/j2re1.4.2_06/bin/java
```

Linux User provided commands

The commands provided by the S@N.IT! administrator can be:

- Executable files (binaries)
- Shell scripts.

The S@N.IT! services launch these commands as daemons so they cannot be interactive.

Examples:

```
UserNotificationCommand=/etc/sanit/NotifyCommand.template

UserNotificationCommand=/mypath/myprogram
```

Error logging on Linux S@N.IT! Server

If the *EnableSystemErrorLog* configuration parameter of the Linux S@N.IT! Server is set to true, a message will be recorded in ***/var/log/messages*** each time a component is detected as faulty.

Miscellaneous

S@N.IT! Agent and S@N.IT! Server daemons

S@N.IT! Agent and S@N.IT! Server daemons can be stopped and restarted using the command:

```
service sanitd [start|stop|restart]
```

S@N.IT! commands path

command	directory
sanit	/usr/bin
rc.sanit	/etc
san_info, san_snap	/usr/sanit/bin

Management of SAN configuration changes

When a new SAN component (storage device,switch) is connected to the SAN, or when a SAN component is removed (disconnected) from the SAN, it is necessary to restart the fibre channel driver.

If the fibre channel driver has been installed as a module, run the following commands:

- For Emulex adapters:

```
rmmod lpfcdd
insmod lpfcdd
```

- For Qlogic adapters:

```
rmmod ql2300
insmod ql2300
```

If the fibre channel driver has been linked to the Linux kernel, reboot the Linux host.

Solaris Platforms

S@N.IT! Features

Solaris platforms cannot be configured as S@N.IT! Server.

S@N.IT! LUN access control mechanism is not supported on Solaris S@N.IT! Agents. It remains in INACTIVE state.

Solaris Installation of S@N.IT! Limited Edition

Delivery

The Solaris *S@N.IT! Limited Edition* software is delivered in the *S@N.IT! Limited Edition* CD–ROM as two packages:

- **SANITdev.sparc_solaris58_32** contains the supported SAN devices
- **SANITall.sparc_solaris58_32** contains binaries, commands and configuration files.

Prerequisites:

Hardware requirements:

– Sparc platforms with 32–bit architecture and a TCP/IP connection.

– Emulex fibre channel adapters (for S@N.IT! Agents).

Software requirements:

– Solaris 8 or 9.

– Emulex package version **5.02c** available in the *Fibre Channel non-AIX drivers* Bull CD-ROM, or from Emulex Web site.

– Java run time 1.4.2 (delivered within *S@N.IT! Limited Edition* CD–ROM).

Note: refer to SRB for up to date pre–requisites information.

Prerequisites Software Installation

Emulex driver package

Refer to Emulex documentation for the installation and configuration of the **lpfc** package.

Java run time

To install the Java Run Time on Solaris hosts, perform these steps:

1. Login as **root**
2. Insert the *S@N.IT! LUN Access Control* CD-ROM into the CD-ROM drive.

   The CD-ROM filesystem is automatically mounted to the /**cdrom**/***<CDROM Part Number>*** directory (for example `/cdrom/76741629_001_2`)
3. Run the following commands:

```
cd /users
sh /cdrom/<CDROM Part Number>/Solaris/java/j2RE-1_4_2_06-solaris-sparc.sh
```

   This installs the Java Run Time into the /**users**/**j2re1.4.2_06** directory. After the installation of the S@N.IT! software, you must configure the **JavaPath** parameter to the correct value (see *Selecting the Java version*, on page 12-19).
4. Take out the CD-ROM using the `eject` command.

Software Installation

To install S@N.IT! Limited Edition on Solaris hosts, perform these steps:

1. Login as **root**
2. Insert the *S@N.IT! LUN Access Control* CD-ROM into the CD-ROM drive.

   The CD-ROM filesystem is automatically mounted to the /**cdrom**/***<CDROM Part Number>*** directory (for example `/cdrom/76741629_001_2`)
3. Run the following commands:

```
pkgadd -d /cdrom/<CDROM Part Number>/Solaris/SANITdev.sparc_solaris58_32
pkgadd -d /cdrom/<CDROM Part Number>/Solaris/SANITall.sparc_solaris58_32
```

4. Take out the CD-ROM using the `eject` command.
5. Verify that the installation is successful with the following command:

```
pkginfo | grep SANIT
```

What happens during installation:

- Configuration files are delivered in the /**etc**/**sanit** directory. This directory will also contain the data of the S@N.IT! Server.
- Executable files (commands and jar) are delivered in the /**usr**/**sanit** directory.
- The default trace directory **(**/**var**/**tmp**/**sanit**) is created.
- During a first installation the **sanit.cfg** file is created with **ServerRole** and **AgentRole** set to true and **Server** parameters set to 127.0.0.1 (IP loopback address); this means that the Linux host is configured as both a S@N.IT! Agent and a S@N.IT! Server. Launch the S@N.IT! GUI on the local host and update the local configuration (see Chapter 5. *Configuration and Launching*).

Software De–installation

To de–install S@N.IT! Limited Edition on Solaris hosts, perform these steps:

1. Login as **root**
2. Launch the following commands:

```
pkgrm SANITall
pkgrm SANITdev
```

Configuration

Selecting the Java version

S@N.IT! software requires Java run time 1.4.2. If this is not the default Java version for a host, change it using the **JavaPath** configuration parameter.

Example:

```
JavaPath=/users/j2re1.4.2_06/bin/java
```

Miscellaneous

S@N.IT! commands path

command	directory
sanit	/usr/bin
rc.sanit	/etc
san_info, san_snap	/usr/sanit/bin

Management of SAN configuration changes

When a new SAN component (storage device, switch) is connected to the SAN, or when a SAN component is removed (disconnected) from the SAN, it is necessary to restart the fibre channel driver configuration. This requires to run the following command:

```
touch /reconfigure
```

Then, reboot the host.

GCOS7 Platforms (Diane)

S@N.IT! Features

All S@N.IT! features are supported on a GCOS7 platform.The S@N.IT! GUI displays the GCOS7 volumes in the File System tab.

GCOS7 Installation

Prerequisites

Hardware requirements

- DPS 7000/XTA server
- Emulex fibre channel adapters

Software requirements

- V7000 version V2.13 or later.

Installation

The GCOS7 platform has a Windows–like interface on which S@N.IT! will be installed. Please refer to Windows Platforms, on page 12-5 for installation and post–installation operations.

GCOS8 Platforms

S@N.IT! Features

The S@N.IT! agent features are supported on:

- The AIX based IOSP and DBSP server processors on GCOS 8 DPS9000/TA and TA2 platforms.
- The Linux based FAME server and DBSP server processor on GCOS 8 Novascale 9000 platforms.

The server processors of a specific GCOS 8 host system are shown as a group within the S@N.IT! GUI.

GCOS8 Installation

Prerequisites

See the specific sections on AIX Platforms, on page 12-1 and Linux Platforms, on page 12-12 for detailed prerequisites for these operating systems. Below are the requirements from a GCOS 8 perspective.

Hardware requirements

- For Helios:
  - Novascale 9000 Linux server
  - V9K1.0.12 software or later
- For Olympus:
  - IOSP (I/O Server Processor) of a DPS9000/TA or DPS9000/TA2 platform
  - IOSP software version D6 or later is required
- For both Helios and Olympus: DBSP (Database Server Processor)
- Emulex fibre channel adapters

Software requirements

- For Olympus:
  - GCOS 8 Software Release 5.1 or later
  - G8CM version 5.6 or later

Installation

The GCOS 8 DPS9000/TA and TA2 platforms have an AIX–like interface on which S@N.IT! will be installed. Please refer to AIX Platforms, on page 12-1 for installation and post–installation operations.

The DBSP could be either AIX or Linux. Please refer to the specific section for installation and post–installation operations.

Glossary

This glossary contains abbreviations, key–words and phrases that can be found in the S@N.IT! documentation.

AIX
Advanced Interactive eXecutive. IBM UNIX™ operating system derived from AT&T UNIX™ System V.

ASN.1 (Abstract Syntax Notation One)
A notation that enables both complicated types to be defined and values of these types to be specified.

BDC
Backup Domain Controller.

Bridge
A device that provides an FC (Fibre Channel) interface for SCSI devices.

CD
Channel Director

CLI (Command Line Interface)
A command that provides a user interface to perform actions on SAN. Useful to build scripts.

DAE (disk array enclosure)
A storage device that includes an enclosure, up to 10 or 30 disk modules (depending on model), one or two Fibre Channel LCCs, and one or two power supplies.

DAS / NDAS (Disk Array Subsystem)
A family of storage subsystems. (See also **DAE** and **DPE**).

DBSP
Data Base Server Processor (for GCOS8)

DPE (disk array processor enclosure)
A storage device that includes an enclosure, up to 10 disk modules, one or two SPs, one or two Fibre Channel LCCs, and one or two power supplies.

Fabric
The term fabric is used to refer to a set of interconnected switches, even if the set is limited to a single switch.

FC–AL (Fibre Channel Arbitrated Loop)
An arrangement of Fibre Channel stations such that messages pass from one to the next in a ring.

FCP
Fibre Channel Protocol for SCSI.

GUI
Graphical User Interface.

HA-CMP (High Availability Cluster Multi–Processing)
It is the core software for AIX clustering. It provides mechanisms that recognizes changes within a cluster and coordinates the use of AIX features to create a highly available environment for critical data and applications.

HBA (Host Bus Adapter)
SCSI or Fibre Channel adapter.

Hub
A device to which several others are attached, providing a common point of connection to all other devices in a network.

ID
The unique address of a SCSI device. 8–bit SCSI can have up to eight IDs; 16–bit up to sixteen IDs; 32–bit up to 32 IDs.

IOSP
Imput/Output Server Processor (for GCOS8)

JBOD (just a bunch of disks)
Another name for DAE (Disk Array Enclosure).

LCC
Link Control Card.

LPP
Licensed Program Product.

LUN (logical unit)
One or more disk modules (each having a head assembly and spindle) bound into a group — usually a RAID group. The operating system sees the LUN, which includes one or more disk modules, as one contiguous span of disk space.

LUN mapping
process that allows to control access to LUN from the host connected to the SAN.

LUN masking
process that allows to control access to LUN from the storage subsystem itself.

MIB
Management Information Base

MP
Multi-processor.

MSCS
Microsoft Cluster Server. High Availability in windows NT: distributed architecture, DB server and CI must be on a separate node.

NetLS
Network License System.

NLS
National Language Support.

OPP
Optional Program Product.

PCI
Peripheral Component Interconnect (Bus).

PDC
Primary Domain Controller.

RAID (Redundant Array of Independant Disks)
A RAID provides convenient, low-cost and highly reliable storage by saving data on more than one disk simultaneously.

RSF
Remote Service Facilities.

SAN (Storage Area Network)
A high speed network that establishes a direct connection between storage elements and servers or clients.

SCSI (Small Computer Systems Interface)
An intelligent peripheral I/O interface with a standard, device independent protocol that allows many different peripheral devices to be attached to the host's SCSI port.

Server
(or Host). A computer that runs an operating system.

SNMP (Simple Network Management Protocol)
A means of communication between network elements allowing management of gateways, routers and hosts.

SP (Storage Processor)
A printed-circuit board with memory modules and control logic that manages the storage-system I/O between the server FC adapter and the disk modules. The SP in a DPE storage system sends the multiplexed fibre channel loop traffic through a link control card (LCC) to the disk units. For higher availability and greater flexibility, a DPE can use a second SP.

Switch
A network device that switches incoming protocol data units to outgoing network interfaces at very fast rates, and very low latency, using nonblocking, internal switching technology.

TCP/IP (Transmission Control Protocol/Internet Protocol)
An industry–standard, nonproprietary communication protocol suite that allows connectivity between equipment from different manufacturers.

WWN (World Wide Name)
A Name Identifier which is worldwide unique, and represented by a 64-bit unsigned binary value.

Zoning
Zoning is a feature that enables the creation of several logical areas on a unique physical network. Devices can communicate only with other devices in the same zone.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X

Vos remarques sur ce document / Technical publication remark form

Titre / Title : Bull S@N.IT! User's Guide

Nº Reférence / Reference Nº : 86 A2 59EF 06

Daté / Dated : December 2004

ERREURS DETECTEES / ERRORS IN PUBLICATION

AMELIORATIONS SUGGEREES / SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Vos remarques et suggestions seront examinées attentivement.
Si vous désirez une réponse écrite, veuillez indiquer ci-après votre adresse postale complète.

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.
If you require a written reply, please furnish your complete mailing address below.

NOM / NAME : _____ Date : _____

SOCIETE / COMPANY : _____

ADRESSE / ADDRESS : _____

_____

Remettez cet imprimé à un responsable BULL ou envoyez-le directement à :

Please give this technical publication remark form to your BULL representative or mail to:

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

Technical Publications Ordering Form

Bon de Commande de Documents Techniques

To order additional publications, please fill up a copy of this form and send it via mail to:
Pour commander des documents techniques, remplissez une copie de ce formulaire et envoyez-la à :

BULL CEDOC
ATTN / Mr. L. CHERUBIN
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

Phone / Téléphone : +33 (0) 2 41 73 63 96
FAX / Télécopie +33 (0) 2 41 73 60 19
E–Mail / Courrier Electronique : srv.Cedoc@franp.bull.fr

Or visit our web sites at: / Ou visitez nos sites web à:
http://www.logistics.bull.net/cedoc
http://www–frec.bull.com **http://www.bull.com**

CEDOC Reference # Nº Référence CEDOC	**Qty** Qté	**CEDOC Reference #** Nº Référence CEDOC	**Qty** Qté	**CEDOC Reference #** Nº Référence CEDOC	**Qty** Qté
_ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ [_ _]	
_ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ [_ _]	
_ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ [_ _]	
_ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ [_ _]	
_ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ [_ _]	
_ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ [_ _]	
_ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ [_ _]		_ _ _ _ _ _ _ _ _ [_ _]	
[_ _] : **no revision number means latest revision** / pas de numéro de révision signifie révision la plus récente					

NOM / NAME : ____________________ Date : __________

SOCIETE / COMPANY : ____________________

ADRESSE / ADDRESS : ____________________

____

PHONE / TELEPHONE : ____________________ FAX : ____________________

E–MAIL : ____________________

For Bull Subsidiaries / Pour les Filiales Bull :
Identification: ____________________

For Bull Affiliated Customers / Pour les Clients Affiliés Bull :
Customer Code / Code Client : ____________________

For Bull Internal Customers / Pour les Clients Internes Bull :
Budgetary Section / Section Budgétaire : ____________________

For Others / Pour les Autres :
Please ask your Bull representative. / Merci de demander à votre contact Bull.

PLACE BAR CODE IN LOWER LEFT CORNER

BULL CEDOC
357 AVENUE PATTON
B.P.20845
49008 ANGERS CEDEX 01
FRANCE

ORDER REFERENCE
86 A2 59EF 06

Bull

Utiliser les marques de découpe pour obtenir les étiquettes.
Use the cut marks to get the labels.

AIX, Linux, Solaris,
Windows, GCOS7,
GCOS8

S@N.IT!
User's Guide

86 A2 59EF 06

AIX, Linux, Solaris,
Windows, GCOS7,
GCOS8

S@N.IT!
User's Guide

86 A2 59EF 06

AIX, Linux, Solaris,
Windows, GCOS7,
GCOS8

S@N.IT!
User's Guide

86 A2 59EF 06