Nimsoft® Monitor™

Getting Started Guide

version 6.00

Document Revision History

Document Version	Date	Changes
1.0	6/30/2010	Initial version *Nimsoft Server Getting Started Guide*
2.0	10/24/2011	Simplified and revised.
3.0	6/29/2012	Revisions for NMS v6.00

Contact Nimsoft

For your convenience, Nimsoft provides a single site where you can access information about Nimsoft products.

At http://support.nimsoft.com/, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- Nimsoft Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about Nimsoft product documentation, you can send a message to support@nimsoft.com.

Legal Notices

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Nimsoft LLC disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Nimsoft LLC shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Nimsoft LLC and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Nimsoft LLC as governed by United States and international copyright laws.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Nimsoft LLC's standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Trademarks

Nimsoft is a trademark of CA.

Adobe®, Acrobat®, Acrobat Reader®, and Acrobat Exchange® are registered trademarks of Adobe Systems Incorporated.

Intel® and Pentium® are U.S. registered trademarks of Intel Corporation.

Java(TM) is a U.S. trademark of Sun Microsystems, Inc.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Netscape(TM) is a U.S. trademark of Netscape Communications Corporation.

Oracle® is a U.S. registered trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of the Open Group.

ITIL® is a Registered Trade Mark of the Office of Government Commerce in the United Kingdom and other countries.

All other trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

Contents

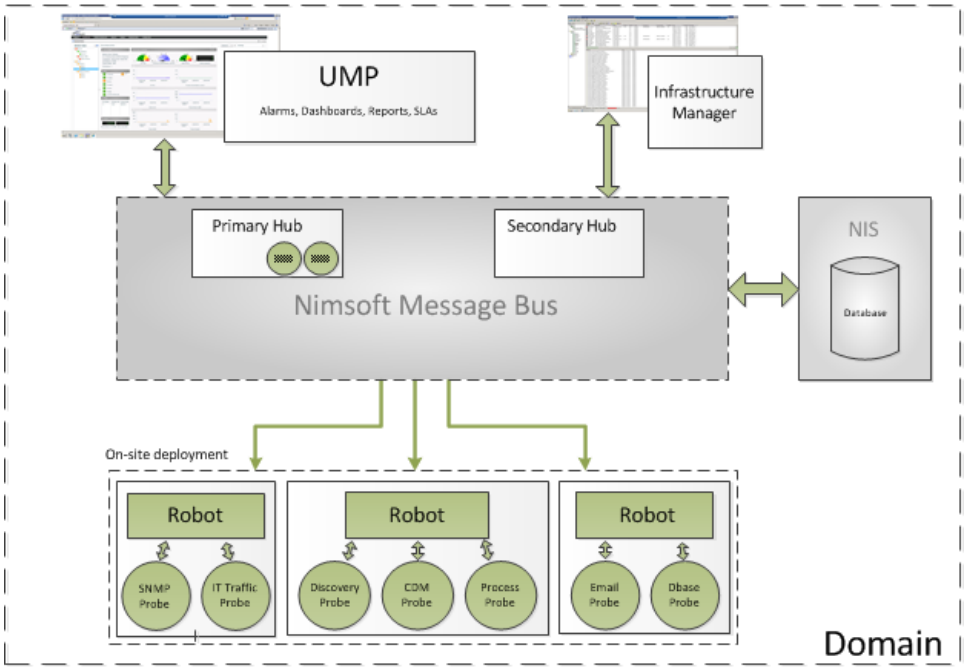
Chapter 1: Nimsoft Monitor

Nimsoft Monitor, one component of the Unified Management solution, provides management for systems, applications and networks. Nimsoft Monitor is made up of these components:

- Nimsoft Server (which includes the Nimsoft Message Bus, primary Hub, and Nimsoft Information Store or NIS)
- Unified Management Portal (called UMP, provides flexible and extensible dashboard views into the managed environment, as well as certain administrative capabilities)
- Infrastructure Manager (provides administrative and management control of the Nimsoft Infrastructure, as well as an alarm window to view alarms and messages)
- The Nimsoft Infrastructure (includes Hubs, Robots, and Probes, which are distributed throughout the managed IT environment)
- The Nimsoft Alarm and Service Level Management (SLM) solutions

The primary source of information on product functionality and configuration is the detailed on-line help and documentation distributed with each of these products.

The following diagram illustrates how these components are interrelated.



Nimsoft Infrastructure

- Domain - All of the Nimsoft Infrastructure components are grouped into a logical set called a Domain.
- Primary Hub - a message concentrator and re-distributor. Its main responsibility is to respond quickly to incoming messages, dispatch these messages to connected subscribers and/or queue, and to maintain system information, such as name-tables, Robot configuration files, etc. It is designed to handle a large number of messages posted/published by its Robots.

  More than one Hub can be installed, in which case secondary Hubs may be used to group robots that perform similar operations, or have the same geographic location, or possess the same departmental code, and so on. Installing at least two Hubs is recommended, as the configuration information will be replicated on the Hubs and have a better chance of surviving a power or system outage.

  Other components connect to the Hub to receive dedicated messages and perform specific activities. For example, the Alarm Server (nas) service collects through a dedicated queue to the Hub and catalogs raw alarms from the probes. It then performs message suppression, filtering, and uploads alarm information to the NIS, from where it can be accessed and displayed.
- Robot - the first line of management for the Probes. The Robot controller starts and stops the Probes at the required times, and the Robot spooler collects, queues, and forwards messages from the Probes to the specified Hub. The Robot also provides a simple database service for its Probes, so that collected data survives power outages. A computer that is monitored by a Probe needs a Robot installed on it, as the Nimsoft installation method uses the Robot to install Probes on the machine.
- Probes - installed on computers within the managed environment, Probes access and monitor specific local resources or events, then send alarms/messages to the controlling Robot. Certain probes, when installed on the Primary Hub, operate as small software programs and provide a number of back-end services that support the overall operation of Nimsoft Monitor.

Extending a Domain Across Firewalls

Most companies today have one or more firewalls in their network, both internally between different networks and externally against a DMZ or Internet.

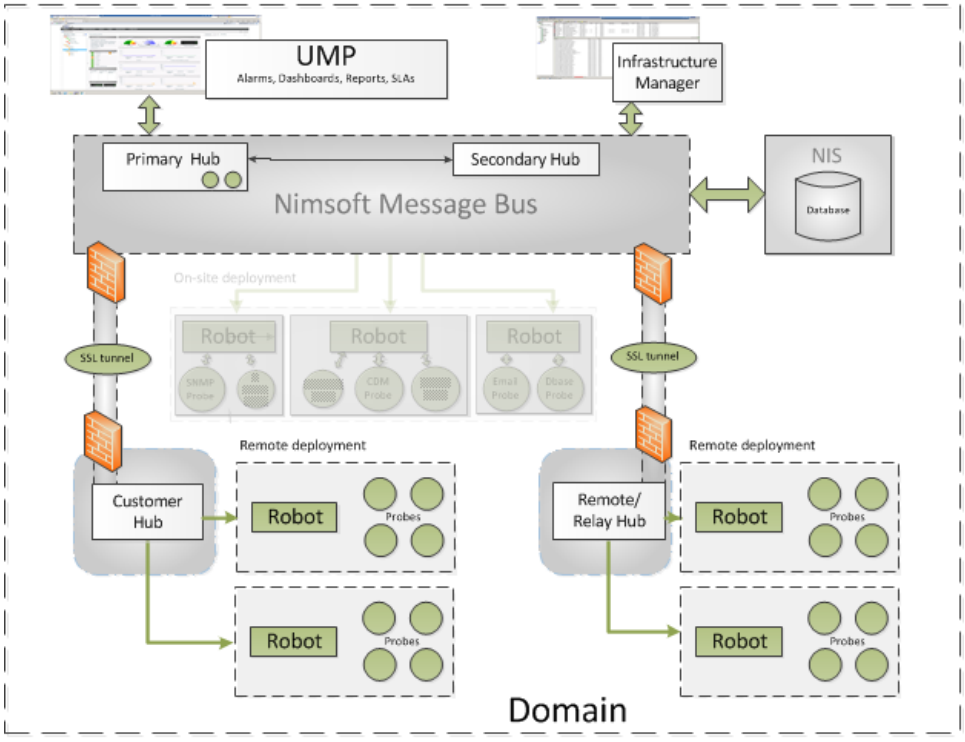
Network administrators are often reluctant to open a firewall to a number of IP addresses or ports in order to make it possible for IT management applications to work. This makes it difficult to administrate and monitor the whole network from a central location.

The solution is to set up a Tunnel between two Hubs that is separated by a Firewall. The Tunnel sets up a VPN-like (Virtual Private Network) connection between the two Hubs and enables all Nimsoft requests and messages to be routed over the Tunnel and dispatched on the other side. This routing will be transparent to all the users within Nimsoft. The only requirement for setting up a Tunnel is that one of the Firewalls opens for connection to the target Hub on one port.

Security is the main issue when opening a Firewall for external connections. The Tunnel is implemented using the SSL (Secure Socket Layer) protocol, which is currently the most widely deployed security protocol today (it is the protocol behind Secure HTTP (HTTPS)). The security is handled in two ways; certificates to authenticate the Client and encryption to secure the network traffic over the network:

Authorization and Authentication: The Tunnel provides authorization and authentication by using certificates. Both the Client and the Server need valid certificates issued by the same CA (Certificate Authority) in order to set up a connection. In the case of setting up a Tunnel, the machine receiving the connection (the Server) is its own CA and will only accept certificates issued by itself.

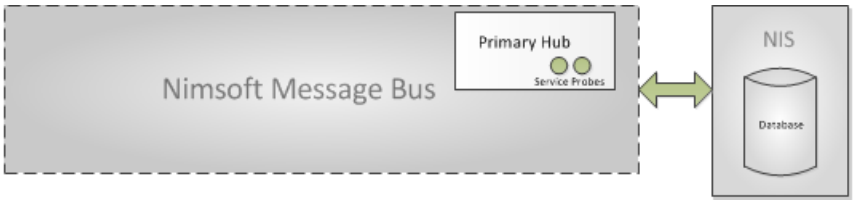
Encryption: The encryption settings span from None to High. No encryption means that the traffic is still authenticated and is therefore recommended for Tunnels within LANs and WANs. Selecting higher encryption levels are more resource intensive for the machines at both ends of the tunnel.



Installation of the Nimsoft infrastructure includes provision of the DMZ wizard, which aids in setting up tunnels between Nimsoft Hubs. See the *Nimsoft Monitor Installation Guide*, or the online help available by double-clicking the Hub probe within Infrastructure Manager.

Chapter 2: Nimsoft Server

Nimsoft Server comprises the Nimsoft message bus, the primary Hub, and a set of service probes that manage the flow of alarms and QoS data messages from producers to consumers, including the Nimsoft Information Store (NIS), which resides on a MSSQL, MySQL, or Oracle database.

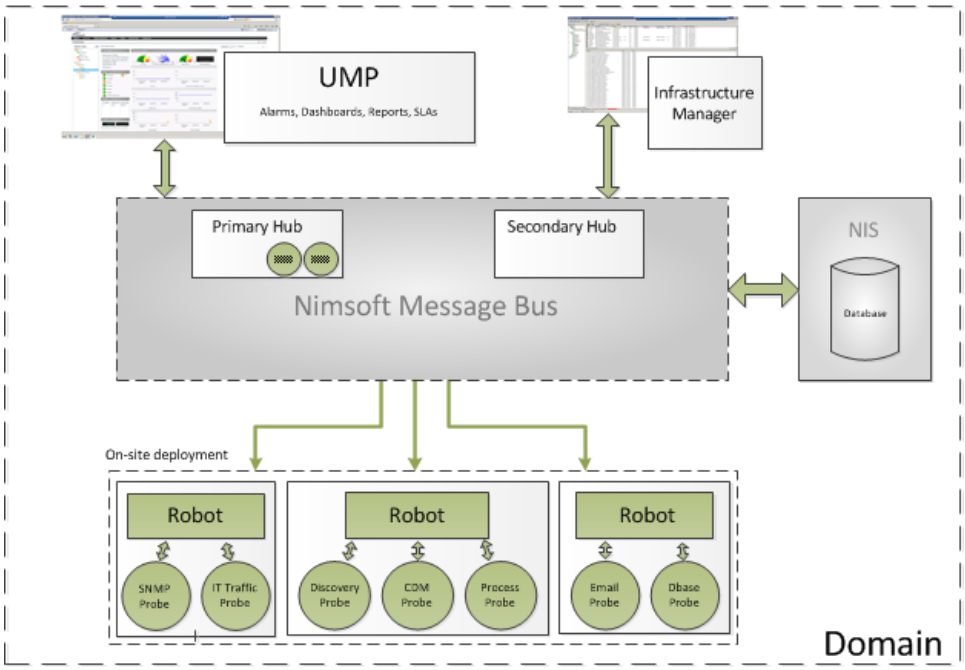


Domain

The Domain is the top-level node in the hierarchy, and a site is normally set up with one Domain. It is used to group one or more Hubs in a logical set such as a company or enterprise.

Various security aspects, such as user profiles, user permissions and access rights are distributed within the Domain.

The illustration below shows a Nimsoft Domain--encompassing Server, database, applications, and distributed infrastructure (robots and probes):



Bus

Nimsoft message handling

The Nimsoft bus should be viewed as a message bus that provides a set of services to client and server processes connected to it. The two major components of the Nimsoft bus are the Robot and Hub processes. They provide the client/server applications with an entry-point (Robot) and an exit-point (Hub) to and from the bus. The message flow on the bus is managed using routing and naming schemes.

Nimsoft Message Model

The Nimsoft message model is based on the request/response and the publish/subscribe models. Request/Response is the standard way of communicating over the network. A client issues a request to a server and the server responds to the request. The publish/subscribe model is useful when a client wishes to send of some kind of data without a designated receiver. This could be messages containing performance-data, an alert, and data to be inserted into some database, or messages targeted for gateway servers. The server merely listens on one or more specific subjects (registered by the Hub), and is notified by events whenever data is available.

Hub

The Hub is a message concentrator and re-distributor. It is the collection point for all messages coming from the various installed Robots. Many other components can connect to the Hub to receive dedicated messages and perform other specific activities. One such component is the Nimsoft Alarm Server (nas).

The Hub responds quickly to incoming messages, dispatches these messages to connected subscribers and/or queues and maintains a variety of system information, such as name-tables, etc. The Hub may be used to 'group' a set of Robots that perform the same operations, have the same geographical locations, have the same departmental code, etc. It is designed to handle a large amount of messages that are published by its Robots, and dispatches the messages to its clients. Messages published under a subject without subscribers or queues will automatically be discarded.

Robot-Hub message flow

All postings performed by the various Probes will be handled by the spooler within the Robot. The spooler will attempt to pass the message over to its current Hub, and will de-spool the messages when the Hub responds positively to the request. When the Hub receives the message it takes a look at the message subject field, and dispatches the message to the appropriate recipients of this posting. The recipients may be other Probes functioning as gateway processes (converting "alarm" postings, for example, to SNMP traps) or dedicated servers listening on subjects specifically designed for this.

The Subscribe mechanism

The Subscribe mechanism will enable clients of the Nimsoft message bus to select messages based on their subjects rather than on the sender address. Whenever a client decides to receive message postings from the Nimsoft bus, it sends a subscribe request to the Hub. The client will receive messages matching the subscribed subjects from the Hub. A client may use the following methods when subscribing:

- Subscribe - connects to the Hub, and gets messages as long as the client is alive and running
- Attach - the Hub configures a Message queue to hold the messages if the client is passive (not running). So when the client comes back up, all messages will be passed on (even the ones that were received when the client was inactive).

The message queues

The Hub is designed to handle a large amount of incoming message and request traffic. To maintain a high message throughput, message queues are used. A message queue is a FIFO (first in first out) queue used to temporarily hold the messages until a client is ready to read the data. As mentioned earlier, messages without subscribers will be discarded. Therefore it is vital to use message queues to ensure reliability and to be able to guarantee the message delivery. The Hub messages queues are found in the /Nimsoft/hub/q directory.

The Name Service

All Probes controlled by a Robot will be registered in a probe list managed by the Robot controller. In addition to the probe list, a list of all active Probes (probes that listen to a bound port, and respond to a command set), is distributed to the Hub for rapid delivery upon request (from, for example, the Infrastructure Manager). The names found in these tables are the basis for the name-to-IP port resolution, and constitute what we define as a Nimsoft Address. A client may query the controller for a name/IP resolution in a similar manner as queried from DNS or WINS, based on the service name (for example, nas).

Queues, subscribers and message delivery

The transfer of messages to and from a Hub is done through queues. There are different ways of configuring queues, allowing a wide range of communication setups. In most everyday situations, the queue is a transparent part of the infrastructure. All you as an administrator normally will need to do is to define communication links between components (Robots, Probes and Hubs), and the necessary queues are automatically set up for you. The queues fall into two main categories:

- A permanent queue is stored in the local Hub database, and survives a restart of the Hub. This type of queue is set up where it is important to ensure that the message is delivered even if the receiver is down when the message is sent. Typically, this is the case when sending messages to nas (Nimsoft Alarm Server) to be stored in a database. The controller at the receiving Robot attaches to the queue at startup and fetches all the messages queued up while it was down, if any. After that, new messages are fetched as soon as they are received by the Hub and put into the queue.
- A temporary queue is a more ad-hoc type of mechanisms used for less critical communication paths. For example, an alarm viewer portlet can open a subscription to alarm messages from the Hub. Then, a temporary queue is created on the Hub, and messages are forwarded to this queue as long as the alarm viewer is active. When you close the viewer, the queue is removed.

Robot

The Robot is the first line of management for the Probes. The Robot starts and stops the Probes at the required times, collects, queues and forwards messages from the Probes onto the specified Hub. Each computer that is being monitored by a Probe will need a Robot installed on it.

The Robot has three dedicated tasks:

- Control the Probes attached to the Robot
- Ensure that messages published by the probes reach the Hub
- Provide a simple database service for its Probes

The Robot, on a 24x7 basis, ensures that the attached probes run without human intervention. It relieves the system administrator of repetitive, time consuming tasks such as performing system surveillance, application monitoring, network monitoring, etc.

A Robot consists of two major elements, the controller and the spooler, as well as a small database:

- The controller manages the probes with respect to startup time and running modes. A probe may be configured to run, for example, every night between 22:00 and 23:00 (known as a timed probe), or it may be configured to run continuously (as a daemon probe).
- The spooler will, in most cases, forward the messages directly to the current Hub. It attempts to 'bulk' the messages whenever possible, to decrease network load. You may configure the spooler to store the messages locally (to a disk file), and send them when given criteria (severity, time and amount) are met.
- The third part of the Robot is called the Hub Database server (hdb). Many tasks in a Nimsoft solution involve the storage of data for threshold monitoring and data trending. The hdb is a database service used by the the controller and spooler (and optionally other probes with storage needs) as a lightweight database service.

Probes

Probes are small, dedicated pieces of software that monitor specific resources or events. Each Probe can be easily configured for specific monitoring requirements.

As seen from Nimsoft Server, the Robot controls a set of Probes and provides the necessary communication ports for all of them. Seen from the connected system, the probe is an internal process capable of:

- Accessing internal system resources from probe-specific program code
- Sending messages on the Nimsoft bus via the messaging features in the Nimsoft API
- Receiving commands from the Nimsoft Hub and maintaining a configuration file via the command features of the Nimsoft API

Usually, the Probe has its current configuration stored in a local configuration file. This file is maintained from the Infrastructure Manager console with Probe Oriented Configuration (POC) or from UMP with Service Oriented Configuration (SOC).

Security Model

The main security issues are to determine whether to restrict users and/or Probes based on predefined permissions within the Domain. These include:

- Access: who has permission to do what
- Authentication: is the client who he/she/it claims to be?
- Encryption: make it impossible for others to read the data

Access Control Lists (ACLs)

Access Control Lists consists of a set of access properties and permissions. When the administrator creates new users, or modifies the properties for an existing user, he attaches the user to an ACL. Users attached to an ACL will have the properties and permissions defined for that ACL.

The administrator can also create new ACLs or modify the properties for the existing ones. ACLs can be administered from Infrastructure Manager.

Chapter 3: The Unified Management Portal

Note: Detailed user documentation for UMP is available from its online help.

The Nimsoft Unified Management Portal (UMP) provides these capabilities:

- Viewing and managing alarms
- Accessing your Web reports
- Viewing and managing alarms
- Viewing discovery status
- Launching the dashboard designer
- Viewing custom dashboards
- Managing users
- Managing the computer systems
- Graphing QoS data
- Viewing summaries of data in table format
- Viewing network topology and Root Cause Analysis diagrams

The following applications, or portlets, are available within UMP:

- **Account Admin** - Allows you to create, modify, or delete users. You can also set passwords for users.
- **Alarm Console** - The main window for viewing and managing alarms. The window displays information about alarms in a table format, and you can use toolbar icons and menu options to view information and take action on alarms. The Alarm Console is integrated with Infrastructure Manager, and it can also be opened in the Service Level Manager.
- **Custom Dashboards** - Allows you to perform the following actions:
  - Access your custom dashboards, which display QoS data and alarms from monitored systems on your network.
  - View and manage your alarms.
- **Dashboard Designer** - Allows you to design custom dashboards to display QoS data and alarms using meters, alarm objects, gauges, charts, tables, panels, and so on.

- **Discovery Status** - Consists of a pie chart showing the discovery status of systems on your network. Immediately after installation of the Nimsoft Server software, the Nimsoft Discovery component starts searching your network for computer systems, provided that the Discovery option was selected during the installation of the Nimsoft Server. The process runs continuously and updates the diagram to show the current status, ensuring that computer systems that are removed or newly installed are reflected in the pie chart. Clicking on the pie chart displays a table with information about the systems in the pie chart.
- **List Designer** - Allows you to design lists to be displayed in the List Viewer application.
- **List Viewer** - Displays data in a table format. The data can be in the form of text, numbers, gauges, alarms, or graphs.
- **Maintenance Mode** - Allows you to set the computer systems on your network to Maintenance Mode so they are temporarily unmonitored. The monitoring parameters for the system are retained, and when maintenance mode ends they are monitored again as before.
- **Nimsoft Remote Admin** - Provides a management console for discovery (DS) and configuration (ACE) data in NIS. It shows the state of all Configuration Items in the database. Using the NIS Manager, you can set the state and specify the monitoring properties for the various computer systems discovered on the network.
- **QoS Chart** - Allows you to see a visual representation of QoS data. You select the host, QoS measurement, target, and time range, and the data is displayed as a graph. You can display multiple measurements on a single graph, and can view multiple graphs at a time. You can choose the graph format (line, area, or column graph) and can maximize the graph to view it at a larger size. You can save a set of graphs as a report to view later.
- **Relationship Viewer** - Displays the relationships among devices on your network in intuitive, visual diagrams. Also performs root cause analysis (RCA) to determine the device causing an outage and suppresses alarms from dependent nodes.
- **Reports** - Contains these types of web reports:
  - Quality of Service (QoS) reports, which must be manually created using the report_engine probe GUI. This GUI is launched by double-clicking the report_engine probe in Infrastructure Manager. See the Nimsoft online probe documentation for details on the report_engine.
  - Service Level Agreement (SLA) reports, which are automatically created for Service Level Agreements built in the Service Level Manager.
- **Unified Reports** - A comprehensive set of Business Intelligence (BI) tools, providing robust static as well as interactive reporting and data analysis capabilities. The Unified Reports support drag-and-drop dashboarding, built-in charting, web reporting, and report scheduling, distribution and historical versioning.
- **Web Content** - Allows you to link to a web page.

Dashboard Designer

The Dashboard Designer application is your arena for designing and accessing your Dashboards. You may design Dashboards to monitor computer systems on your network for QoS data and alarms using various template widgets, such as alarm objects, meter objects, charts, and tables:

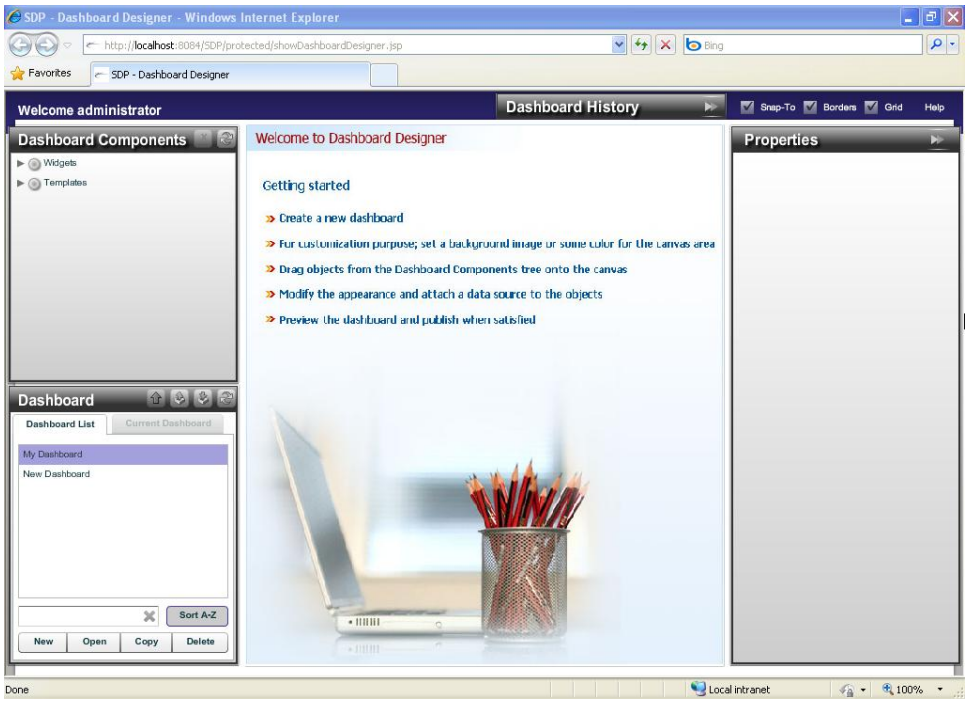
- Alarm objects can be filtered to reflect the state of the computers you want.
- Meter objects can be connected to different data sources (QoS, probes, variables etc).
- Panels can be used to build Dashboards with several levels in a tree structure.
- Table objects can be used if you want to present the output from a query to the NIS as a table in a dashboard.

The layout of the dashboard components and the background canvas can be configured with a wide range of colors, fonts sounds and data sources.

You may also import and use on of the Dashboard templates available. There are four Dashboard Templates available; two for network devices and two for server systems.

You will also find several general objects, like text objects, images etc. The Dashboard Designer also contains a Preview tool, letting you see the appearance and layout of the Dashboard before publishing it.

The Dashboards will, when saved and published, be available in the Unified Monitoring Portal. There you can see the state and QoS values of the monitored systems and also manage the alarms.



The Dashboard Designer window contains the following main sections:

Dashboards

This section, located in the lower left corner, lets you open, edit or copy your existing Dashboards, or you can create a new Dashboard. You may even download one or more of the Dashboard templates included, modify them and use them as your own Dashboards. The section also contains functionality for publishing the Dashboards, making them available in the Dashboards list.

Dashboard Components

This section contains the building blocks, or widgets, (objects) you can use when designing your Dashboards. You simply drag and drop objects or templates on the Dashboard you are editing or creating. Objects can be alarm components, meters, panels etc.

In addition to the widgets, there is a node called Templates. You may save an object as a template. If you have configured an object and want to save and use it in the future when designing other Dashboards, right-click the object and select ‘Save as Template’

Dashboard History

This section contains functionality for logging changes made to the current dashboard on the canvas. This supports an undo/redo mechanism for a number of operations; typically for adding, deleting, resizing and moving objects.

Properties

This section contains the properties available for the selected object. When designing a Dashboard, you drag objects from the Dashboard Components section and drop them on the canvas. Selecting an object on the canvas, the properties available for the selected object will appear in the Properties section. Configure the objects by assigning the properties you want.

Custom Dashboards

You can create custom dashboards using the Dashboard Designer. Which dashboards you see depends on the permissions set in the ACL for your user account.

Dashboard Pane

The color of the icons in the tree structure represents the highest alarm severity for alarm objects on the dashboards. Double-click an icon and the corresponding dashboard is launched in the dashboard pane.

The dashboards can contain meters, alarm objects, gauges, charts, tables, panels, and so on.

Alarm and panel objects reflect the severity level of the alarm with the highest severity. Double-clicking an alarm object brings up the alarm list, enabling you to manage the alarms.

Mini Map tool

The Mini Map tool zooms in on an area of a dashboard. A minimized version of the dashboard is shown in the Mini Map window. A slider lets you zoom in or out of the dashboard on the canvas.

Managing Alarms in Dashboards

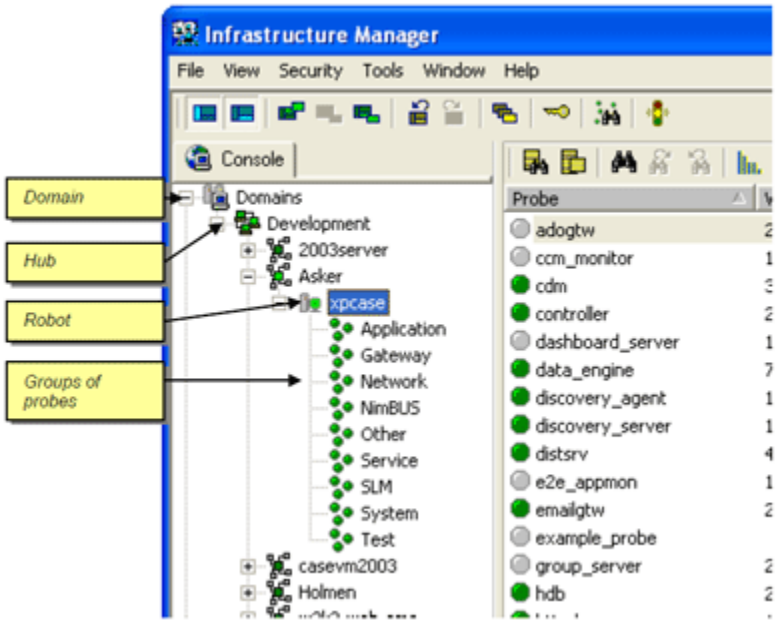
If the dashboard contains alarm objects, you can display the Related Alarms list by double-clicking an alarm icon that is not green (there are no alarms associated with green objects).

Chapter 4: Infrastructure Manager

The Infrastructure Manager is an important interface for configuration and management of the system. It provides:

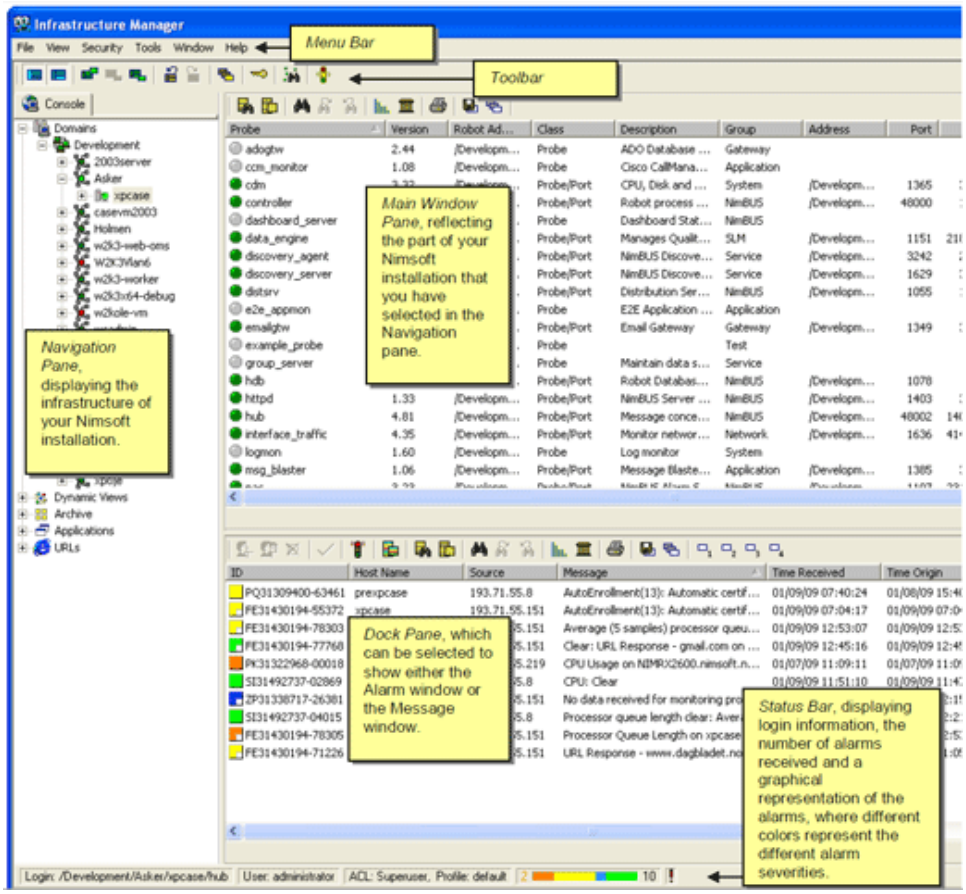
- An explorer-style overview of systems being monitored
- Management and administration tools
- Drag and drop Probe deployment and installation
- An alarm window to view all alarms and messages

The Infrastructure Manager connects to an active Hub and allows you to control, configure and manage all Robots and Probes connected to that Hub. You can navigate through the Nimsoft infrastructure by expanding elements in the tree structure, configure the various components by double-clicking the components in the right-hand list, and deploy robots and probes using drag-and-drop.



The Infrastructure Manager window

The main components of the Infrastructure Manager window are shown below:



For more information, including user guidance and reference, see the *Infrastructure Manager* guide, available from the Downloads tab on the Nimsoft support website (Downloads > Technical Documentation > Infrastructure Manager).

Chapter 5: Alarm Console: Overview

The Alarm Console allows the user to view and operate on the alarms that he or she is allowed to access. The console is fully event driven and updates automatically. The user can define complex filters to quickly get to specific subsets of alarms. Based upon the privileges set up for the user, the user can perform a set of management operations as well, like creating and attaching notes, managing actions and filters, or setting alarms to visible/invisible states. For regular users, operations such as accepting and acknowledging alarms are available. The Alarm Console also offers transaction history and query functionality against historical data.

NOTE: This documentation describes functions and options that are available to a user with maximum privileges. Some menu options and buttons in your application may be unavailable (grayed out) depending on your privileges. The alarms you can see and the actions you can perform are defined in an ACL.

Alarm Overview

The Alarm solution is an application based on the Nimsoft Infrastructure. This application consists of three main parts in addition to the Nimsoft infrastructure:

- The Alarm Monitoring probes check the host computer for symptoms of error situations. This may be checking free disk space, log-file contents, performance problems or stopped system processes. When a problem is found, a Nimsoft message describing the problem is sent via the Robot to the Nimsoft Hub.
- Nimsoft delivers several standard probes for monitoring a wide range of operating systems and applications for error situations which are known to be likely to occur in that particular type of system. Nimsoft works closely with the vendors of such systems to provide focused monitoring for the everyday problems affecting their users and support staff.
- The infrastructure (basically the hubs) route the message to a Alarm server (nas). The nas stores the message in a small, local database.

Alarms

The Alarm product utilizes the infrastructure to create system and network monitoring solutions. Nimsoft delivers out-of-the-box solutions for a wide range of operating systems and applications. These solutions provide a quick start and will cover about 80% of the needs for server and workstation monitoring in most organizations. However: experience has shown that covering the remaining 20% is very costly. These needs vary from site to site, and a product covering them all will be extremely difficult both to develop and to learn. Also, the load on the monitored system will be higher than we believe is necessary.

We believe that most customers are best served by providing an infrastructure where they can develop their own solutions, targeted directly at the problems causing them the most trouble. This approach requires a development platform which:

- Does not force people to waste time on learning a proprietary programming language or scripting language (or, for that matter, a standard language they do not already know)
- Does not force people to waste time on inventing network access and security solutions to all the different operating systems used
- Does not force people to waste time on distributing and installing their solutions on all the monitored machines.

Alarm Server (nas)

The main purpose of the nas is to receive and manage incoming alarm messages. The nas:

- Supports message suppression
- Provides clients with services such as:
  - Update events.
  - Repository services (get, list, close etc.)
- Supports message filtering
- Supports automatic actions (auto-operator)
- Mirroring capabilities
- Alarm message handling

The Alarm Server (nas) is a service probe that attaches itself to a Hub queue (nas), and receives alarm messages distributed by the Hub. An alarm message is generated by a probe somewhere on the Nimsoft bus. This is a "broadcast-type" message without any particular receiver (a posting) that may be picked up by any processes subscribing to the "alarm" subject. The nas subscribes to the "alarm" subject and acts upon the incoming message by storing information about the alarm into a database in the nas subdirectory.

The Alarm Server responds to a command-set reachable by anyone with the correct access, as well as notifying through the use of message postings whenever state changes occur. Anyone may listen to the postings (notifications) made by the nas.

Message suppression

Many error situations in the monitored system can result in a huge number of alarms. For example, this can be the case if the logmon probe monitors a logfile for an application which enters an infinite loop, logging errors within the loop. This may result in a huge number of identical alarms, creating unnecessary load on the system, network and the Nimsoft Server. This can be avoided by using the message suppression mechanism.

The message suppression models supported by the nas, are:

- Standard suppression is a simple model that suppresses messages with an exact match on message subsystem id, severity level and message text
- Key suppression is a model based on a suppression key that follows a message. When the key suppression is enabled messages with matching suppression key will be suppressed.

Example

With Key suppression on, these two messages from the same probe are equal:

Filesystem '/usr' is filled 95% (suppkey: FsProbe-/usr)

Filesystem '/usr' is filled 55% (suppkey: FsProbe-/usr)

The result of this would be one message in the alarm server database, but it would have recorded both of them as valid transactions (and therefore logged them in the transaction log). So if the sequence was as displayed (95% first, then 55% as the last status) then the administrator would experience the state as a filesystem with 55% filling grade (which is the correct way to see things).

An administrator may choose to ignore the suppression mechanism based on key, due to the fact that he/she wants to view the messages as they are reported by the probes.

Automated acknowledge

The suppression key described above can also be used to automatically clean up in the alarm list when the probe detects that the critical situation is resolved. This is done by enabling automatic acknowledge based on key. This means that alarms with severity level "clear" automatically acknowledges any previous alarms with the same suppression key. In the example above, a reasonable configuration of the disk-monitoring probe would be to send the first alarm (95% full) with severity level "serious", while the last one (55% full) could have severity level "clear". When the last alarm arrives, everything is back to normal, and the administrator does not have to respond to the first alarm after all. Then the alarm is automatically acknowledged by the nas, leaving the administrator with a "to-do list" with as little "noise" as possible.

Alarm transactions

It is quite useful to follow the complete message life from the initial message and to when the message is closed (acknowledged) and via multiple suppressions. The Alarm Server is capable of logging all transactions to a specific transaction logfile. This is accomplished through a filtering mechanism tunable by the administrator. To keep the transaction logfile as manageable as possible, it is automatically administered (copied to a 'save' location) at configured intervals. The saved transaction logs are renamed to the following format: trans_timestamp.log where timestamp is the time (in seconds since epoch) it was managed.

Subsystems

In the Alarm window, the different alerts are classified by their subsystem ID, identifying which part of the system the alert relates to. This is a hierarchical list of codes, allowing you to group alarms in as wide or narrow groups as you want. This list is stored in the nas, and if you develop your own probes or customize the standard ones, you may also define your own list of subsystems. This list also maps the subsystem code into a text string for improved readability.

Notification messages

The Alarm Server notifies the world about changes to its alarm database by issuing postings on the Nimsoft bus. When an alarm message is received and its footprint is not previously recorded, an alarm_new message is generated. However, if the footprint already exists, an alarm_update message is generated. Whenever a client closes (acknowledges) an alarm it will be removed from the currently active alarms, and an alarm_close message will be generated. All transactions such as new, suppress and close are logged to the transaction log, and may be viewed through the nas configuration tool.

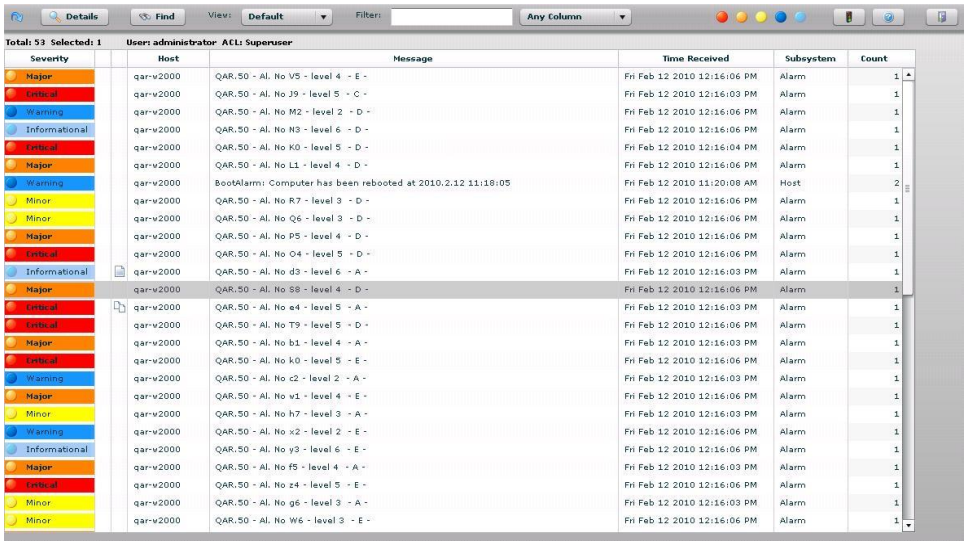
Handling Alarms

Alarms will appear in the Alarm window. This Windows application displays the alarm messages in the nas database for the operator. Alternatively, you can install a gateway for forwarding the alarms to other messaging infrastructures such as E-mail, GSM/SMS, pager or SNMP messages, or integrate Alarm more tightly to a systems management framework using one of the available framework integration kits. Either way, the operator is automatically informed about the problem a few seconds or minutes after the symptom appears, rather than having to check manually.

Alarms automatically handled by the auto operator Alarms can also be automatically handled by setting up profiles in the Auto operator in the nas probe.

Alarm Console window

The Alarm Console is the main window for viewing and managing alarms. The window displays information about alarms in a table format, and you can use toolbar icons and menu options to view information and take action on alarms.



For more information, including user guidance and reference, see the Alarm Console User Guide, available from the Downloads tab on the Nimsoft support website (Downloads > Technical Documentation > Alarm Console User Guide).

Glossary

Nimsoft Infrastructure

Hub

The Hub is a service in the Nimsoft infrastructure that performs the following actions:

- Manages a group of Robots
- Collects and redistributes messages published by the Probes
- Maintains several central services including store-and-forward, failover, security and naming services
- Manages message subscribers such as gateway services, applications and servers.

Infrastructure Manager

The Infrastructure Manager handles management operations within the Nimsoft environment. You can navigate through the Nimsoft infrastructure by expanding elements in the tree structure, configure the various components by double-clicking the components in the right-hand list, and deploy Robots and Probes using drag-and-drop.

Nimsoft Address

A Nimsoft address consists of four basic elements, the *Domain, Hub, Robot,* and *Probe*, each separated by a forward slash (**/**). For example, take the following address:

`/Nimsoft/oslo/wscase/nas`

That Nimsoft address is interpreted as follows:

- **Nimsoft** is the Nimsoft Domain name
- **oslo** is the Nimsoft Hub name
- **wscase** is the Nimsoft Robot name
- **nas** is the Probe name. In this example, it is the Nimsoft Alarm Server, so we know this address refers to the alarm server on the primary Hub in a domain.

The Nimsoft API has functions that resolve a Nimsoft address to an IP-address and a port.

Probe

A Probe is a program (usually quite small) developed using the Nimsoft API in any of the supported languages such as C/C++, VB, Perl, Java and COM. It connects to the Nimsoft bus infrastructure to send and receive messages, and is administered from Infrastructure Manager with POC (Probe Oriented Configuration) or from UMP if it supports SOC (Service Oriented Configuration).

Probes serve primarily two purposes within the Nimsoft Infrastructure:

- Gathering QoS data from network devices, databases, servers, and applications and placing it on the Nimsoft bus (monitoring probes). Commonly deployed monitoring Probes include the following:
  - cdm (monitors host cpu utilization, disk and memory)
  - interface_traffic (records network interface data on MIB-II compliant devices)
  - url_response (measures response time from the probe to the a web page)
  - vm_monitor (gathers QoS data for instances of virtual machines running on a host machine).
- Providing product utility functions to the Nimsoft infrastructure (utiity Probes), such as the following:
  - web portal services (httpd)
  - dashboarding
  - grouping (group_server)
  - data queuing (data_engine)
  - email services (emailgtw)
  - data archiving (distsrv).

Publishing messages

A message is *published* on the Nimsoft bus when it is simply sent to the nearest Hub without being destined for any particular receiver. The Nimsoft infrastructure automatically ensures that the message is delivered to all Probes subscribing (see page 35) to the SubjectID (see page 35) found in the message.

Quality of Service (QoS)

Many Probes are capable of sending trending data periodically. These messages are formatted and known as Quality of Service messages. They normally contain data such as response times and availability used for Service Level Monitoring and reporting. The data is captured by the Nimsoft Data Engine and is recorded into the database.

The message layout is as follows:

Field	Description
QoS	Name of the Quality of Service data
hostname	Specifies the originator of the data
source	Specifies the source of the data, e.g. ping
sampletime	When the sample was recorded (epoch value)
samplevalue	The actual value
samplerate	The rate of the sample interval
samplemax	The maximum value (if any), e.g. disk size.

Robot

The Robot is the first line of management for the Probes. The Robot starts and stops the Probes at the required times, collects, queues and forwards messages from the Probes onto the Nimsoft bus, where it is delivered to the specified Hub.

Subject

All Nimsoft messages must contain a Subject ID. This is a text string classifying the message for all components on the Nimsoft, allowing them to subscribe to some messages and ignore others. All messages with the same subject should also have identical data structure.

Example: The alarms handled in the Nimsoft Alarm solution are simply Nimsoft messages with **Subject="alarm"**. All components of the Nimsoft Alarm solution work on messages with this subject and ignore subjects used by other solutions.

Subscribe

When a client (e.g. a Probe or gateway) *subscribes* to Nimsoft Messages, it informs the Hub that it wants a copy of all messages it detects on the Nimsoft bus with a certain (set of) Subject ID (see page 35) (s). You may have several probes subscribing to the same subject. They will all automatically receive all messages with that subject ID.

Timed probes vs. daemon probes

A *timed* Probe runs once and then terminates, awaiting the next point in time when it is configured to start. A *daemon* Probe is always active if activated by the operator. If it stops, it is immediately restarted by the Robot.

Domain

The Domain is the top-level node in the hierarchy. It is used to group multiple Hubs into a logical set such as "company" or "enterprise." Various security aspects such as user profiles, user permissions and access rights are distributed within the Domain.

SLM Terms and Definitions

Calculation profile

Calculation profiles are user-created profiles that can be used when defining the calculation properties for Service Level Objects and Quality of Service Constraints. These profiles are based on built-in plug-ins distributed with the Nimsoft Service Level Manager application.

When defining calculation profiles, the profiles will be grouped either as SLO calculations as QoS calculations, depending on if the selected plug-in supports single-data or multi-data series.

Calculation method

A calculation method is the set of rules and conditions determining the way the SLA compliance is calculated.

Compliance period

The compliance period defines the period of time that the SLA should meet the requirements stated by the compliance percentage, typically a day, a week or a month.

The illustration shows the relationship between the *operating period* and the *compliance period*.

The gray area illustrates the compliance period, and the blue area illustrates an operating period with 5 time-specifications.

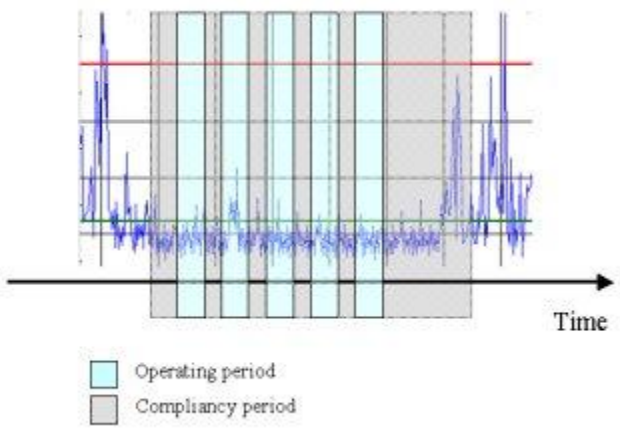
Example:

Let's say that the compliance period (gray area) is defined to be one full week (Monday to Monday). Our requirements for the SLA is defined to be *between 08:00 and 17:00 every weekday* (operating period). Notice that the data (blue plot) within the operating periods (blue area) are included in the SLA computations.

Compliance percentage

The compliance percentage is defined to be the percentage of time that the QoS, constrained by e.g. operating period and thresholds, should be considered compliant within the compliance period..

Each sample is checked within the compliance period and summarized as **failed** or **successful**. The result is compared against the expected compliance percentage (defined by the user).



Consider the data represented by the illustration. The **red** line represents the threshold value, the **green** line represents the average value and the **blue** line represents the actual sample values.

How many samples within the operating period are above the threshold settings?

Zero samples breach the threshold line within the operating periods, thus fulfilling 100% of our compliance requirements. The 5 samples that breach the threshold are outside the compliance period which was e.g. Monday to Monday, with operating periods every weekday from 08:00 to 17:00.

Let's assume that the total number of samples within the operating period is 129, with 9 samples breaching the threshold. This implies that 6.98% of the samples are accounted for as **out of compliance** (9 * 100/129).

If our Service Level Agreement requires a compliance of 98.50% (or better) and the only data defined in this SLA is the above data, then our requirements to the SLA is **breached** due to a current compliance percentage of 100% - 6.98% = 93.02%.

Data Types

We use the following different data types when calculating the compliance:

Automatic (Interval)

QoS Data is recorded at intervals, individually specified in the probe configuration for each of the probes.

Asynchronous

QoS Data is recorded only each time the measured value changes.

Domain

The Domain is the top-level node in the Nimsoft hierarchy, and a site is normally set up with one Domain. It is used to group one or more Hubs in a logical set such as a company or enterprise.

Hub

The Hub is a message concentrator and re-distributor. It is the collection point for all messages coming from the various installed Robots. Many other Nimsoft components can connect to the Hub to receive dedicated messages and perform other specific activities. One such component is the Nimsoft Alarm server.

Operating period

The operating period constrains the QoS samples to one or more time-specifications within the compliance period. This means that samples falling outside these time specifications will not influence the SLO/SLA compliance requirements. Each operating period is defined as a union of one or more time-specifications.

Probes

Probes are small dedicated pieces of software that monitor specific resources or events. Each probe can be easily configured for your own specific monitoring requirements.

Robot

The Robot is the first line of management for the Probes. The Robot starts and stops the Probes at the required times, collects, queues and forwards messages from the Probes onto the Nimsoft bus, where it is delivered to the specified Hub.

Service Level Agreement (SLA)

A Service Level Agreement (SLA) is an agreement to deliver a service within a specified/fixed time-period to an extent where both parties agree on a measurable service levels. The parties may be an IT department delivering services to another department within the company, or by a company and an external service provider.

The services included in the SLA may be a collection of monitored objectives we call Service Level Objectives (SLO). These objectives (or group of objectives) are monitored by dedicated programs (often standard probes) that monitor e.g. network connectivity, application (Oracle, Exchange, e-mail) availability and service (DNS, DHCP) availability.

The SLA is an agreement to deliver a service within a specified/fixed time-period to an extent where both parties agree on a measurable service levels. The parties may be an IT department delivering services to another department within the company, or by a company and an external service provider.

The services included in the SLA may be a collection of monitored objectives we call Service Level Objectives (SLO). These objectives (or group of objectives) are monitored by dedicated programs (often standard probes) that monitor e.g. network connectivity, application (Oracle, Exchange, e-mail) availability and service (DNS, DHCP) availability.

Service Level Objective (SLO)

A Service Level Objective (SLO) is a combination of one or more *component measurements* (Quality of Service) to which constraints are applied. A SLO is said to be in compliance if the underlying measurement values are within the specified constraints. SLO's may have operating periods during which the SLO has to be compliant.

Service Level Manager (SLM)

The Service Level Manager (SLM) is the application where service level configuration and monitoring is performed. The application needs a valid license to operate.

Quality of Service (QoS)

The Quality of Service (QoS) is the atom of the Service Level Management. The QoS is the actual value (sample) collected and used centrally to determine the state of the service level objective. This value is normally collected by a probe like *cdm*, *net_connect*, *url_response* etc. The value is first used for alarm purposes, but if the probe is configured to deliver Quality of Service, then a QoS message is issued.

A QoS constrained by threshold, source, target and operating period settings is used as the building blocks for SLO's.

QoS object

The QoS object is defined by its **QoS name**, **source** and **target**. All sample-data with this unique combination form data-series that may be used as part of any Service Level Objective.

QoS constraint

A constrained QoS object is defined by its **QoS name**, **source**, **target**, **threshold** and **operating period**. This constrained object is the building block of the Service Level Objective.

Alarms

Acknowledge

All new alarm messages received by a Nimsoft Alarm Server (nas) are initially marked un-acknowledged and presented to an operator. When the operator has verified whether there was a problem and possibly fixed it, the operator can acknowledge the message, indicating that the problem is under control. The message is then deleted from the NAS database, but a copy is kept in the history database.

Alarm message

An Alarm is a general message with its subject set to **alarm**. The message is normally generated by a probe responding to a threshold breach, and published onto the Nimsoft bus as a "raw" alarm message. The raw alarm's header contains these main parameters:

- Message text
- Message severity level
- Message subsystem
- Message originator/source
- Timestamp

Alarm Levels

The supported alarm levels are as follows:

Constant	Value
NIML_CLEAR	0
NIML_INFORMATION	1
NIML_WARNING	2
NIML_MINOR	3
NIML_MAJOR	4
NIML_CRITICAL	5

History

When an Alarm message is acknowledged, it is deleted from the NAS database, but it is still kept in a history database. The contents of this database can be viewed from from the Alarm Console.

Subsystem

The subsystem ID is a field in all Alarm messages containing one or more numbers separated by periods, for example **2.31.4**. The subsystem ID corresponds to modules within the monitored system such as security or disk systems.The Alarm Console groups the incoming alarms according to subsystem, allowing you to quickly view alarms related to, using the same examples, security issues or a disk system.

Suppression

In response to IT and network service availability, Nimsoft Alarm probes sometimes generate a number of identical alarms. If suppression is enabled, such messages are treated as one alarm to reduce unnecessary messages being presented to the operator.

Pattern matching

Pattern matching is a simple way of scanning a string for any given pattern. You can match characters or strings of characters using these special characters: asterisk (*****) , question-mark (**?**) and square-brackets **[ ]** .

The target string is scanned for the characters *****, **?**, and **[**. If one of these characters appears, the word is regarded as a pattern, and the filter-field is marked as **True** (a match).

*****	Matches any string, including the null string.
?	Matches any single character
[...]	Matches any one of the enclosed characters. A pair of characters separated by a hyphen (**-**) matches any character lexically between the pair, inclusive. If the first character following the opening **[** is a **!**, any character *not* enclosed is matched.

Some examples:

subset	Match a string that contains **subset** somewhere within it (beginning to end).
^at start*	Match a string that contains **at start** beginning at the first character position.
(123 |[Ee]xit)	Match a string that contains **123** or **Exit** or **exit** somewhere within it (beginning to end).
[Ee][Rr][Rr][Oo][Rr]	Match a string that contains **ERROR** in various capitalization permutations somewhere within it (beginning to end).

Regular expressions

Regular Expressions (RegEx) provides a powerful syntax to find pattern matches within a set of characters or strings. Consult the documentation maintained by Perl.org for more in-depth understanding of how regular expressions are formed and used.