

Nimsoft® Monitor™

nas Guide v4.20 series



Legal Notices

Copyright © 2013, CA. All rights reserved.

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.

For information on licensed and public domain software, see the *Nimsoft Monitor Third-Party Licenses and Terms of Use* document at: http://docs.nimsoft.com/prodhelp/en_US/Library/index.htm?toc.htm?1981724.html.

Contact Nimsoft

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

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

Providing Feedback About Product Documentation

Send comments or questions about CA Technologies Nimsoft product documentation to nimsoft.techpubs@ca.com.

To provide feedback about general CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Documentation Changes

This table describes the version history for this document.

Version	Date	What's New?
4.2	March 2013	Updated for current NMS version.
4.1	November 2012	Fixed localization format to display properly in UMP.
4.0	June 2012	Added alarm enrichment functionality.
3.7	December 2011	Fixed email sending wrong auto operator execute timestamp value in email send; added Apply button after activating/deactivating AO schedules.
3.6	July 2011	Fixed database upgrade issues when NIS bridge is enabled.

Related Documentation

Documentation for other versions of the nas probe ([../nas.html](http://docs.nimsoft.com/prodhelp/en_US/Probes/nas.html))

The [Release Notes](#) for the nas probe

Getting Started with CA Nimsoft Probes

(http://docs.nimsoft.com/prodhelp/en_US/Probes/GettingStarted/index.htm)

Monitor Metrics Reference Information for CA Nimsoft Probes

(http://docs.nimsoft.com/prodhelp/en_US/Probes/ProbeReference/index.htm)

Contents

Chapter 1: nas 4.20	9
Overview	9
Prerequisites	10
alarm_enrichment probe	11
nas probe	12
 Chapter 2: alarm_enrichment and nas Probe Deployment	 15
 Chapter 3: alarm_enrichment Configuration	 17
Setting up cmdbs Environment	18
Setting up Enrichment Rules	19
Setting up Routing Rules	21
Flood protection	22
 Chapter 4: nas Configuration	 23
The Setup Tab.....	23
General.....	24
Transaction Log	26
Message Suppression.....	32
Subsystems.....	34
Forwarding & Replication.....	36
NIS Bridge	40
The Status Tab	43
The Auto-Operator Tab	45
Properties.....	45
Profiles	46
Triggers.....	60
Scripts.....	66
Pre-processing Rules	70
Scheduler	74
Computer State Monitor.....	80
Pattern Matching in Auto-Operator.....	83
Setting an Operating Period	84
The Name Services Tab	86
Name Services Properties	87

Name Services Address Table	88
The Notes Tab	89

Chapter 5: The Script Editor **93**

Keyboard shortcuts	100
--------------------------	-----

Chapter 6: The Alarm List **101**

Appendix A: The NAS Extentions to Lua **109**

Alarm	110
Database	112
Action	113
Nimsoft.....	114
Note.....	115
Trigger	115
File	116
Timestamp.....	117
PDS	119
Language Extension.....	121

Appendix B: The NAS Command Interface **127**

assign_alarms.....	127
close_alarms.....	128
date_forecast	128
db_query	128
get_alarms.....	129
get_ao_status.....	129
get_info	130
get_sid	130
host_summary	130
nameservice_create	130
nameservice_delete	131
nameservice_list	131
nameservice_lookup	131
nameservice_setlock.....	131
nameservice_update.....	132
note_attach	132
note_create	132
note_delete	133
note_detach	133

note_list	133
Reorganize.....	133
repl_queue_post.....	133
repl_queue_info.....	134
script_delete	134
script_rename	134
script_list.....	134
script_run	134
script_validate.....	135
set_loglevel	135
set_visible.....	135
transaction_list.....	135
trigger_list	136

Chapter 1: nas 4.20

This section describes NAS version 4.1.

This section contains the following topics:

[Overview](#) (see page 9)

Overview

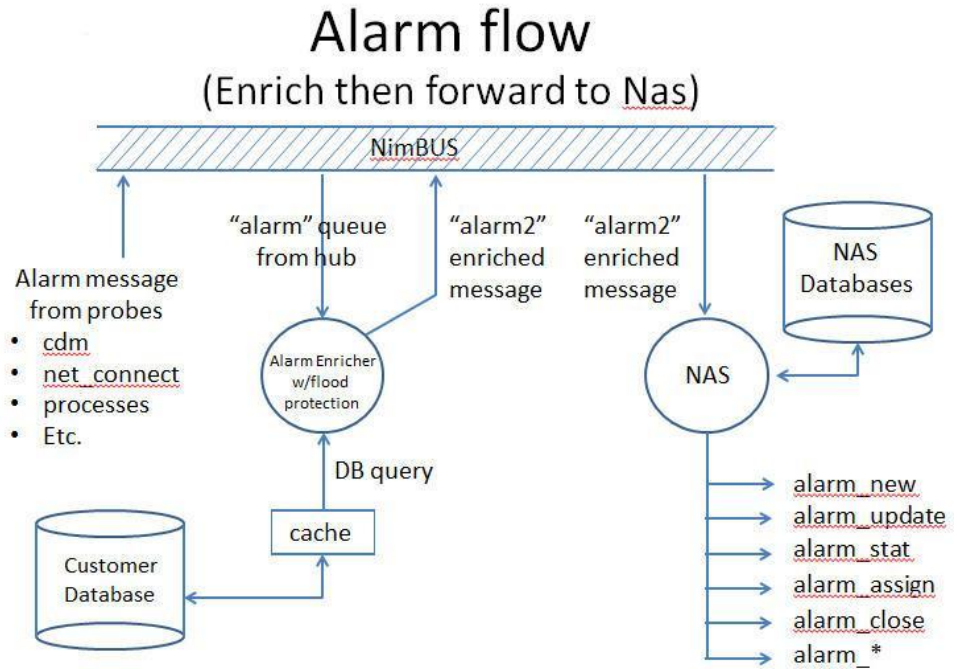
The Nimsoft Alarm Server (nas) stores and administers alarm messages for the Nimsoft Alarm product. The nas package contains two probes: alarm_enrichment and nas.

The alarm_enrichment probe is a pre-processor probe for the nas probe. Alarm_enrichment attaches itself to a permanent queue and receives alarm messages distributed by the Hub. The messages flow into the alarm_enrichment probe, where alarm storms are detected, and messages may be enriched with additional information read from external data sources, using a Configuration Management Database (cmdb). The alarms are renamed to alarm2 and are then sent to the nas probe for further processing.

The nas probe is a service probe that attaches itself to a permanent queue and receives alarm2 messages distributed by the Hub. The nas probe acts upon the incoming alarm message, received from the alarm_enrichment probe, by storing information about the message into a database.

The NAS 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. Any application may subscribe to the events generated by the nas. The Enterprise Console and event-viewer both subscribe to these events.

Note: The NAS will not accept and manage alarms without message text, which means the *Message text* field cannot be left empty.



The alarm_enrichment and nas probes are packaged, installed and configured as a single unit. They will display in your system archive as separate probes, however they are configured using the nas probe configuration and saved in the nas.cfg file.

Prerequisites

Important! The nas package requires hub version 5.81 or higher. New distsrv probes will abort rather than update the hub.

alarm_enrichment probe

The alarm_enrichment probe can be configured to read data from various datasources. Each datasource is referred to as a CMDB. Only JDBC-compliant SQL-database sources are supported currently.

Each datasource is defined by its JDBC connect string, user login, password and a query to extract the data. Every datasource provides a user-defined name to be referenced in the enrichment rules. Each enrichment_rule can reference one datasource. A datasource can be used by many enrichment rules.

Once you have defined the CMDBs = datasources, you have to define at least one enrichment rule.

Each enrichment rule defines a matching condition to match on alarms which should be forwarded to this enrichment rule. The enrichment rule defines what alarm enrichment should be performed and from what data source additional information for this alarm should be read.

When an alarm comes in it will be copied to a new event where:

- the message identifier NimId is modified to ensure it is still unique
- the fields qsize, md5sum and subject are removed from the incoming alarm
- all fields starting with "hop" are copied by prepending it with "original_" so that the field "hop0" becomes "original_hop0" in the outgoing alarm.

The alarm is matched against the configured alarm enrichment rules. An overwrite rule defines an alarm attribute and a value to which the alarm attribute should be set. Once an alarm has been processed against the alarm enrichment rules it is passed on to the nas probe for further processing.

At a minimum you will need one routing rule to forward your alarms to your Nimsoft Alarm Server (nas). There might be a situation where you would want to create more than one routing rule.

Items to consider:

- Ensure the data sources you are using are ready for the amount of requests the alarm_enrichment probe might be making to get alarm information.
- Keep an eye on latency to make sure your data source can return results quickly.
- When accessing large and busy databases consider running a shadow database for read-only query purposes.

nas probe

The nas probe has the following features:

- Auto-Operator, aids the administrator in managing the alarm database.
 - Close/acknowledge certain alarms based on matching rules.
 - Automatically assign an alarm to a person / group.
 - Automatically send a GSM/SMS message whenever a criteria is met.
 - Send e-mail message whenever a criteria is met.
 - Execute a command for fixing the problem.
 - Use scripts when processing alarm messages matching the criteria defined for the Auto Operator profile. Scripts can also be run by the Scheduler and by the pre-processing rules filters. You may create/edit these scripts yourself, using the LUA programming language.
- Notes

Possible to create text notes to be attached to Alarms.
- Transaction logging

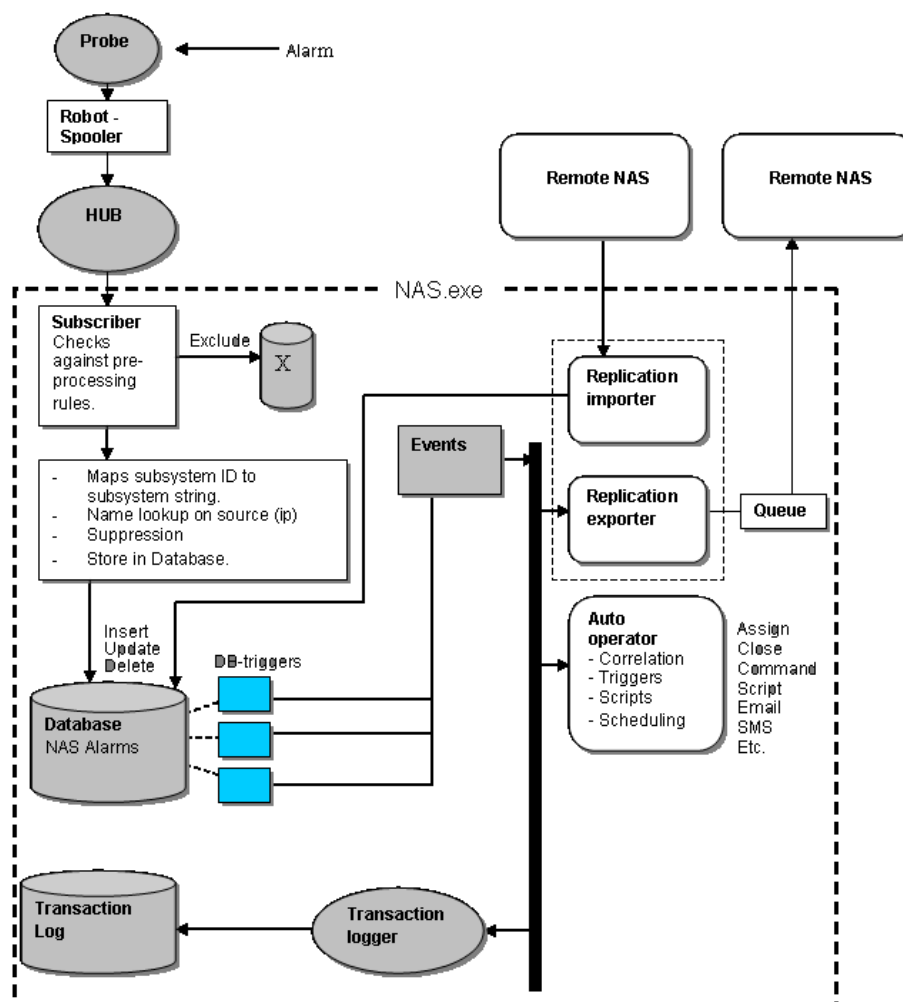
The Alarm Server is capable of logging all steps in the life of an alarm (the alarm transaction) from when the alarm is generated until it is acknowledged.
- Message suppression

Message suppression is a feature used to avoid storing multiple alarm messages caused by the same problem. When receiving a large number of identical alarm messages, you may by setting the *Alarm Suppression divisor* to e.g. 100, store only the first alarm message received, delete the next 99 identical alarm messages, then store number 101, delete the next 99 identical alarm messages etc. Default divisor is 100.
- Assignment roster

A list of operators or other assignment targets, such as a helpdesk, making it easier to assign alarms from within the Auto-Operator profiles or the various alarm consoles.
- Scheduling

Scheduling making it easy to administer alarm filters and auto operator profiles (activating or deactivating) and run scripts.
- Replication

Forward alarm messages to another NAS. This is useful for getting alarms from lower level nas probes (behind a firewall) to an upper level nas probe that can be monitored by UMP.



Chapter 2: alarm_enrichment and nas Probe Deployment

The alarm_enrichment and nas probes rely on the Java package installed with the Nimsoft Management Server.

There are two ways to distribute the probe archive packages. You can distribute the package within Infrastructure Manager or use the standalone Nimsoft Distribution application.

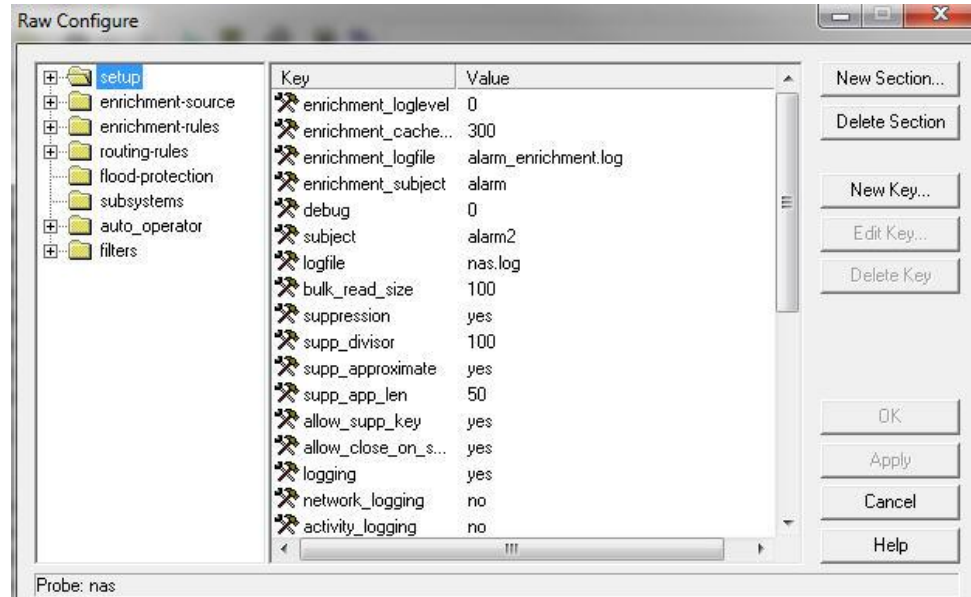
The archive will contain the nas package. This package will deploy and start both the alarm_enrichment and the nas probes. You cannot use nimldr to distribute the nas package.

When this package is deployed:

- Both probes will be automatically started.
- The storm protection functionality will be disabled until the user defines the settings in the nas configuration GUI. The alarm_enrichment probe will perform storm protection.
- The enrichment functionality will be turned off until the user defines the cmdb database location.
- Two queues are created by default:
 - alarm enrichment "alarm" queue
 - nas "alarm2" queue
- The nas probe will perform storage, replication, etc.

Chapter 3: alarm_enrichment Configuration

The alarm_enrichment probe is configured using the Raw Configure option in the nas probe. The configuration settings for this probe is stored in the nas configuration file.



The alarm_enrichment configuration settings are contained in the enrichment-source, enrichment-rules, and routing-rules sections of the raw configuration for the nas probe.

The alarm_enrichment probe subscribes to "alarm" messages, modifies the alarm and submits a new message to the NAS with a modified subject of "alarm2." The nas probe subscribes to the "alarm2" messages.

Users are allowed to change the subject (queue) names. By default, alarm_enrichment probe uses the "alarm" subject and forwards messages to the "alarm2" subject for the nas probe. If the subject name is changed the content in the queues will be lost.

This section contains the following topics:

[Setting up cmdbs Environment](#) (see page 18)

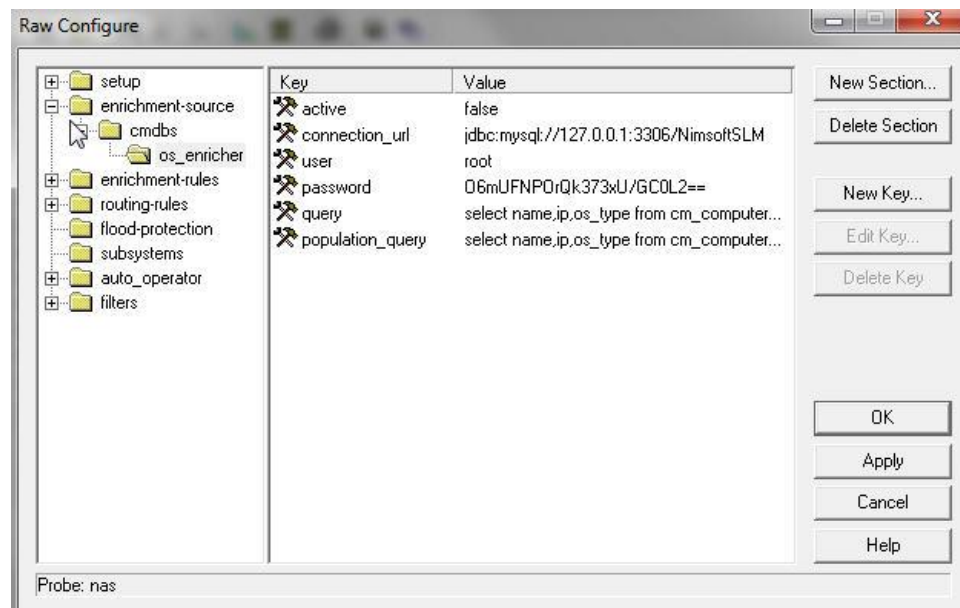
[Setting up Enrichment Rules](#) (see page 19)

[Setting up Routing Rules](#) (see page 21)

[Flood protection](#) (see page 22)

Setting up cmdbs Environment

In the CMDBs section you define one to many data sources to read data from.



Each CMDDB requires a name/tag to uniquely identify the enrichment-source for subsequent enrichment-rules sections. In the example provided we have used `os_enricher` as the CMDDB name. The CMDDB allows you to set the following parameters:

active

Allows you to activate or deactivate the CMDDB

connection_url

The url to the database

Examples:

`connection_url = jdbc:oracle:thin:@//172.17.4.12:1521/ORCL`

`connection_url = jdbc:sqlserver://172.17.8.12:1433;DatabaseName=NimsoftSLM`

`connection_url = jdbc:mysql://172.17.0.12:3306/choslm`

user

The user login name for the database

password

The password associated with the database user. The password is entered in plain text, however it is encrypted and stored in its encrypted form in the configuration file.

query

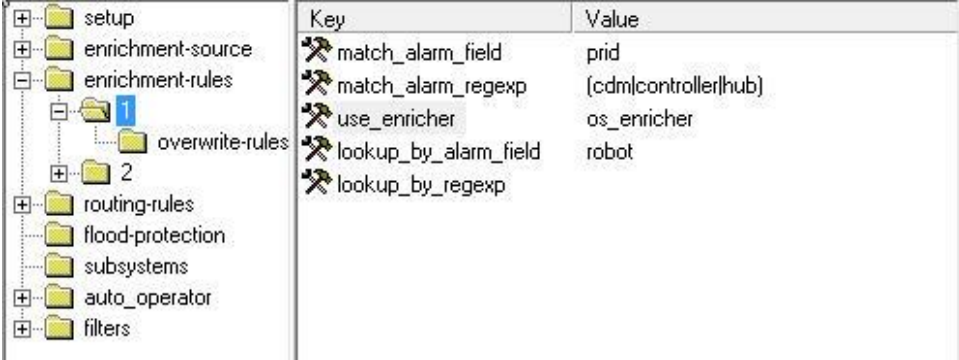
The query to be executed when retrieving a cmdb item from the data source. Specify one "?" where the ID of the item can be filled in your query.

population_query

The pre-population query that should be executed on startup of the probe and at regular intervals. There should not be a "?" in this query as no ID substitution will occur.

Setting up Enrichment Rules

The enrichment rules define the actual alarm enrichment process. First you must define what alarms to match and then you must define the values to add/overwrite in the alarm.



Key	Value
match_alarm_field	prid
match_alarm_regexp	(cdm controller hub)
use_enricher	os_enricher
lookup_by_alarm_field	robot
lookup_by_regexp	

The alarm matching parameters are:

use_enricher

Enter the name of the cmdb enrichment source

match_alarm_field**match_alarm_regexp**

This pair of parameters are used to identify messages to be enriched.

The first identifies the PDS field to match upon and the second identifies legal values. In our example, we will use the value contained in the "prid" field. And, we will enrich the message only apply enrichment when "prid" field (in the incoming message) matches one of the regular expressions "cdm" or "controller" or "hub."

lookup_by_alarm_field**lookup_by_regexp**

Identify the alarm field you want to match in the enrichment-source.

Consider your enrichment-source specifies the following query:

```
select name,ip,os_type from cm_computer_system where name=?
```

In this example, the 'name' column (in our enrichment-source DB) happens to represent the computer name. Notice that the configuration suggests that the value for the "robot" field in the incoming message will be substituted (in prepareStatement) for the "?".

If your robot name was 'robot123', then your query becomes:

```
select name,ip,os_type from cm_computer_system where name='robot123';
```

Enter the alarm field you want to match in the data source. You can enter more than one field name in this parameter. You must enter a separator between the field names.

Overwrite Rules

	Key	Value
+	udata.source	[cmdb.description]
+	udata.user_tag_1	[cmdb.customer]
-	udata.user_tag_2	AUGMENTED
+	udata.custom_1	[cmdb.sla_level]
+	udata.custom_2	[cmdb.priority]
+	udata.custom_3	-1
+	udata.custom_4	[cmdb.location]
+	udata.custom_5	[cmdb.cmdb_key]

Many fields in a message can be enriched at the same time. The query may have many fields/columns returned. Each overwrite rule specifies one field in each alarm to be modified.

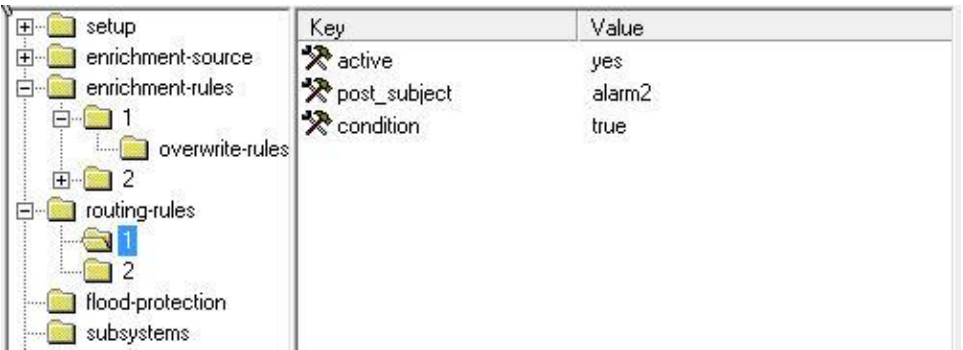
The key contained in the alarm will be replaced by the value returned from the query. Content in square brackets "[]" will be replaced based upon your query results. Contents without the square brackets will be static content.

Nimsoft alarms can be multi-hierarchical and the parameter you want to modify might not be in the main structure of the incoming alarm, but in a substructure such as udata. You must use the full address (hierarchical path) to the parameter.

Setting up Routing Rules

The routing rules allow you to send alarms under different subjects or even to different targets depending on the field values.

Each routing rule needs a unique name. The routing rules are executed in alphabetical order.



The routing rule parameters are:

active

Indicates whether the rule is active or not.

post_subject

The new subject the alarm will be sent with to the nas.

Note: The alarm_enrichment probe contains two default routing rules, one for alarms sent to the nas, the other for storm protection.

condition

The condition under which this rule will be executed.

Flood protection

Flood (also called "storm") protection allows you to automatically enable/disable rules based on the amount of alarms being processed per minute. This functionality is configured in the nas GUI. You can specify one of four states for this protection: disabled, suppression id, probe or robot.

When flood/storm protection is enabled, the alarm messages considered part of a storm can be sent out using an alternate subject, such as "NAS_QUARANTINE". This will only work if another probe (or listener) subscribes to the subject "NAS_QUARANTINE".

Default flood protection requires the following parameters:

routing_rules_during_flood

Determines the routing rule to use during a message storm. Default subject is NAS_QUARANTINE and the messages are sent to the subscriber for this subject.

routing_rules_no_flood

Determines the routing rule to use for alarms not in a storm condition. Messages are sent to the nas probe.

Chapter 4: nas Configuration

The NAS is configured by double-clicking the NAS probe in the Infrastructure Manager application. This brings up the configuration tool for the probe.

This section contains the following topics:

[The Setup Tab](#) (see page 23)

[The Status Tab](#) (see page 43)

[The Auto-Operator Tab](#) (see page 45)

[The Name Services Tab](#) (see page 86)

[The Notes Tab](#) (see page 89)

The Setup Tab

The **Setup** tab allows you to configure various elements of the Alarm Server, such as what suppression methods should be used, what to add to the transaction log, forwarding and replication and so on. This section will discuss these elements in detail. This tab contains the following subsections:

- General
- Transaction Log
- Message Suppression
- Subsystems
- Forwarding & Replication
- NIS Bridge

General

This tab allows you to set various general parameters such as how detailed the NAS should log progress in its log-file.

The screenshot shows the 'Setup' window for the Nimsoft Alarm Server. The 'General' tab is selected, showing various configuration options. The 'Log-file' is set to 'nas.log'. The 'Log-level' is set to 'Less detail'. The 'Publish alarm updates every' is set to '100 duplicate messages'. The 'Activate support for internationalization' checkbox is checked. The 'Storm Protection' is set to 'Disabled', 'Storm Threshold' is '3000', 'Storm Subject' is 'NAS_QUARANTINE', 'Storm Period' is '5min', and 'Storm Capacity' is '1000'. A note at the bottom explains the purpose of the property-sheet.

Setup | Status | Auto-Operator | Name Services | Notes

General | Transaction Log | Message Suppression | Subsystems | Forwarding & Replication | NIS Bridge

Log-file: nas.log

Log-level: Less detail | More detail

Publish alarm updates every 100 duplicate messages.

☒ Activate support for internationalization

Storm Protection: Disabled | Storm Threshold: 3000

Storm Subject: NAS_QUARANTINE | Storm Period: 5min | Storm Capacity: 1000

This property-sheet configures the Nimsoft Alarm Server. You may set up filters that exclude incoming alarm messages, add Auto-Operator methods that manages both incoming and stored alarms. Most action menus are reached by right-clicking the list-control in question. More information is obtained through the on-line documentation.

The fields are:

Log-file

The name of the nas logfile can be changed to the name specified. Before this new logfile name will change the probe must be restarted manually.

Log-level

Sets the level of details written to the log-file. Log as little as possible during normal operation to minimize disk consumption, and increase the amount of detail when debugging.

Publish alarm updates every n duplicate messages

This option specifies how often duplicate NAS events (as subscribed by consoles/gateways) are published.

A message is considered duplicate when message text, subsystem id and severity are equal to the previous message with the same suppression key. This will reduce the events received by the consoles.

Example: If you set this parameter to 10, the message count for suppressed messages will be updated after 10 occurrences of the same message.

Activate support internationalization

Supports the internationalized alarms published by probes.

Storm Protection

You can enable storm protection on the NAS and also determine the key signature elements, such as suppression-id, probe, or robot.

Storm Subject

The subject of the storm alarm message. The default is NAS_QUARANTINE.

Storm Threshold

The number of alarms allowed before the alarms become a message storm.

Storm Period

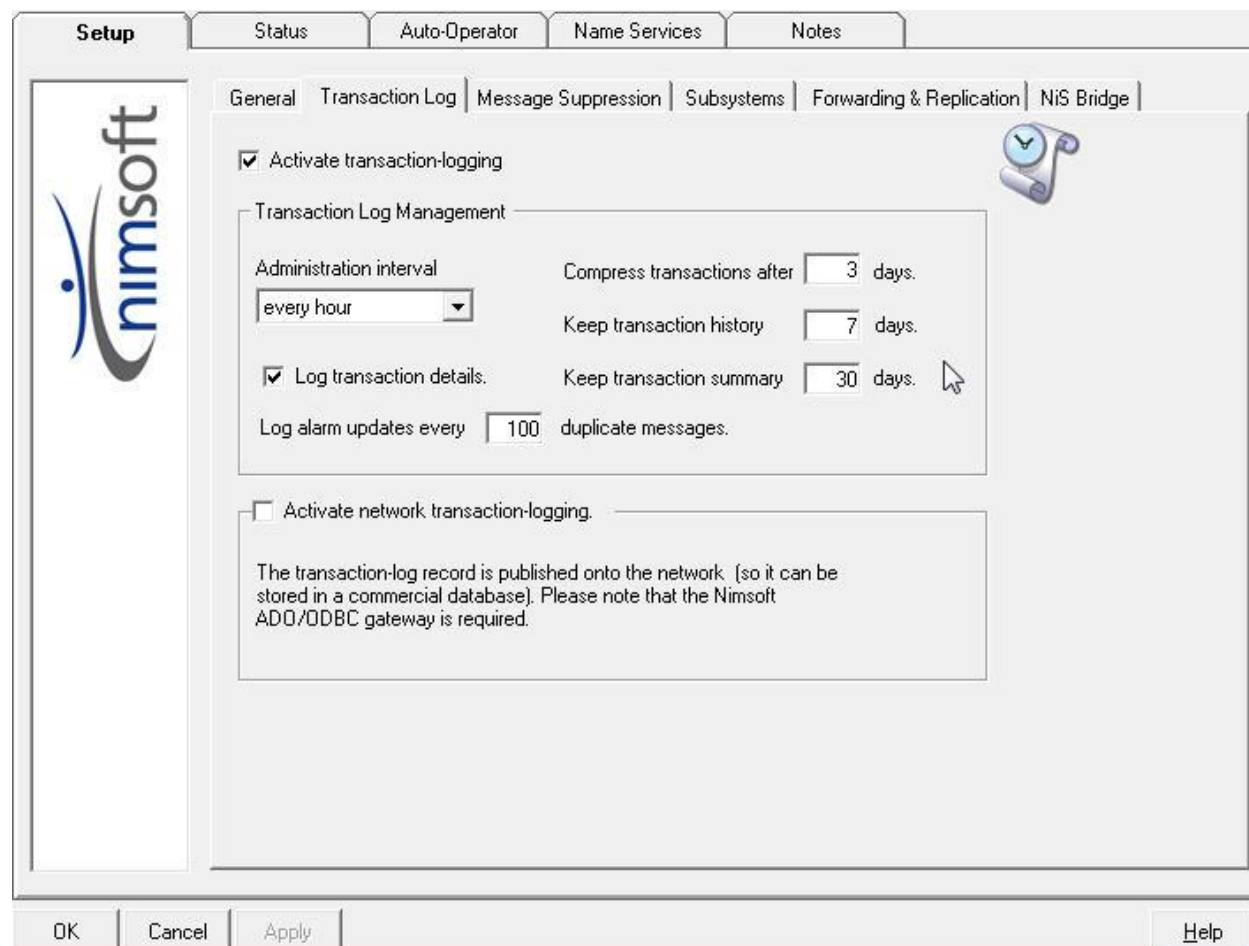
The length of time monitored for number of alarms for storm threshold. Example: 1000 alarms in 5 minutes.

Storm Capacity

The size of the "quarantine list" of alarms.

Transaction Log

The NAS is capable of logging all transactions to a specific transaction log-file. This is accomplished through a filtering mechanism that can be configured by the administrator. It is quite useful to follow the complete message life cycle from the initial message to when the message is closed (acknowledged). To keep the transaction log-file as manageable as possible, it is automatically compressed at the configured administration interval.



The screenshot shows the 'Setup' dialog box with the 'Transaction Log' tab selected. The 'nimsoft' logo is on the left. The 'Transaction Log' tab is active, showing options for transaction logging. A 'Transaction Log Management' box contains settings for administration interval, compression, and history. Below this, there are checkboxes for 'Log transaction details' and 'Log alarm updates every'. At the bottom, there is a checkbox for 'Activate network transaction-logging' and a note about publishing records to the network.

Setup | Status | Auto-Operator | Name Services | Notes

General | **Transaction Log** | Message Suppression | Subsystems | Forwarding & Replication | NIS Bridge

☒ Activate transaction-logging

Transaction Log Management

Administration interval: every hour (dropdown) | Compress transactions after: 3 days (input) | Keep transaction history: 7 days (input) | Keep transaction summary: 30 days (input)

☒ Log transaction details. | Log alarm updates every: 100 duplicate messages.

☐ Activate network transaction-logging.

The transaction-log record is published onto the network (so it can be stored in a commercial database). Please note that the Nimsoft ADO/ODBC gateway is required.

OK | Cancel | Apply | Help

The fields are:

Activate transaction-logging

If this checkmark is set, the NAS logs all steps in the life of an alarm (the alarm transaction) from the alarm is generated until it is acknowledged.

The data is stored in the NAS database *transactionlog.db*, located in *Program Files/Nimsoft/Probes/service/nas*

Transaction Log Management

Administration interval

The interval at which the NAS monitors the size of the transaction log files and truncates them.

Valid options are:

- Every hour
- Every 2 hours
- Every 6 hours
- Every 12 hours
- Daily

Compress transactions after

The events (of type *suppression*) for alarms stored in the transaction log will be deleted after the number of days specified. Default is 7 days.

Keep transaction history

For how long (in days) the transaction history is stored. The transaction history stores all events for each of the alarms handled by the NAS in the database.

Keep transaction summary

For how long (in days) the transaction summary is stored. The default value is 30 days.

The transaction summary for each alarm is stored as one row in the database.

Log transaction details

This options specifies how often duplicate NAS events are stored in the event transaction-log.

A message is considered duplicate when message text, subsystem id and severity are equal to the previous message with the same suppression key. This will reduce the size of the transaction-log and speed up transaction-log queries.

If the *Log transaction details* option is not checked, this log will be empty.

Log alarm updates every n duplicate messages

Enter the number of duplicate messages required before updating the log alarm.

Activate network transaction-logging

Instructs the NAS to publish its transaction-log record onto the Nimsoft, so it can be picked up by the Nimsoft ADO/ODBC gateway for central storing (see the description *Network Transaction logging* below the table).

This option instructs the NAS to publish its transaction-log records onto the network so it can be picked up by the Nimsoft ADO/ODBC gateway for central storing. This option requires the adogtw probe to be installed and configured.

This is useful if you want to use the data recorded in the transaction log-file (containing all alarms) to make reports, statistics etc.

Setting Up Network Transaction Logging

To set up network transaction logging you must select the Activate Network transaction logging option on the Setup > Transaction Log page in the nas probe configuration. The adogtw probe must also be deployed and configured.

Define tables within the database as shown below:

- For a **SQL Server** database:

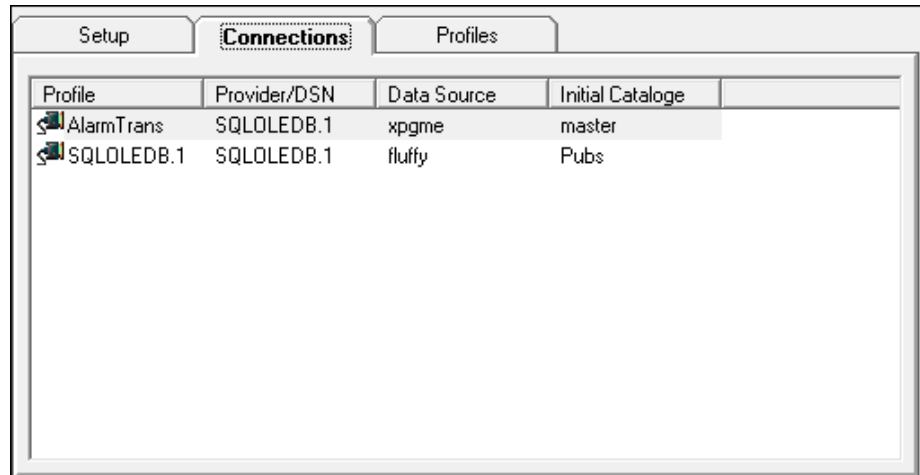
```
CREATE TABLE [dbo].[AlarmTransactionLog] (
  [TypeId] [int] NULL ,
  [TypeDesc] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
  [Created] [datetime] NULL ,
  [Processed] [datetime] NULL ,
  [Hostname] [char] (64) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
  [Source] [char] (64) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
  [AlarmId] [char] (24) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
  [AlarmSeverity] [int] NULL ,
  [AlarmSeverityDesc] [char] (16) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
  [AlarmSid] [char] (48) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
  [AlarmSubsystem] [char] (64) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
  [AlarmMessage] [varchar] (512) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
  [TypeData1] [char] (64) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
  [TypeData2] [char] (64) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
  [TypeData3] [char] (64) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
```

- For an **Oracle** database:

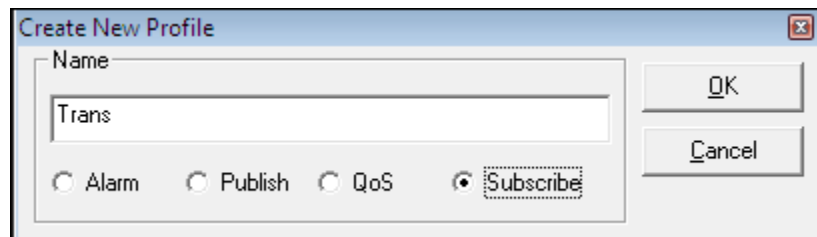
```
CREATE TABLE ALARMTRANSACTIONLOG
(
  TYPEID          NUMBER(10),
  TYPEDESC        VARCHAR2(10),
  CREATED         VARCHAR2(50),
  PROCESSED       VARCHAR2(50),
  HOSTNAME        VARCHAR2(64),
  SOURCE          VARCHAR2(64),
  ALARMID         VARCHAR2(24),
  ALARMSEVERITY   NUMBER(10),
  ALARMSEVERITYDESC VARCHAR2(16),
  ALARMSID        VARCHAR2(48),
  ALARMSUBSYSTEM  VARCHAR2(64),
  ALARMMESSAGE    VARCHAR2(512),
  TYPEDATA1       VARCHAR2(64),
  TYPEDATA2       VARCHAR2(64),
  TYPEDATA3       VARCHAR2(64)
)
```

This example describes how to set up the nas log network transactions to a table in a database. In this example, we assume that the table *AlarmTransactionLog* is created in the database.

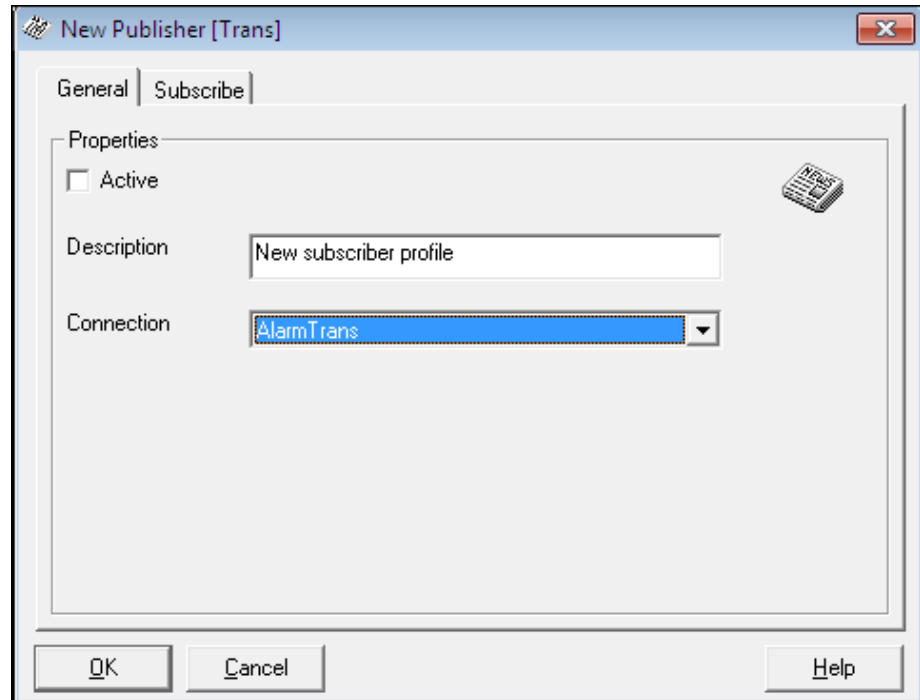
1. Add a connection to the database.



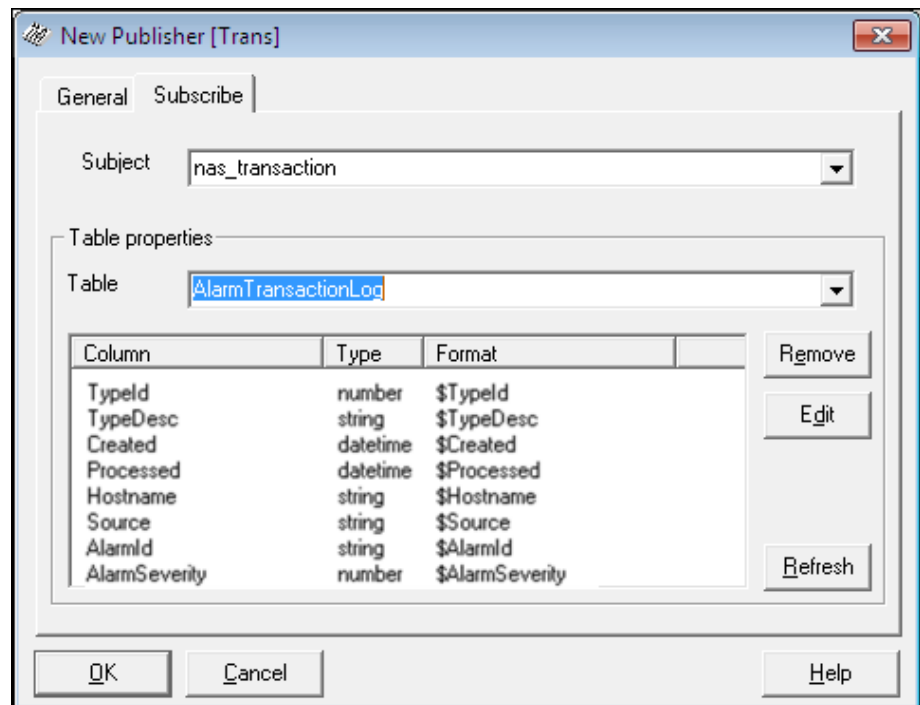
2. Add a new profile and select **Subscribe**.



- Click the **General** tab and select the connection that you created.



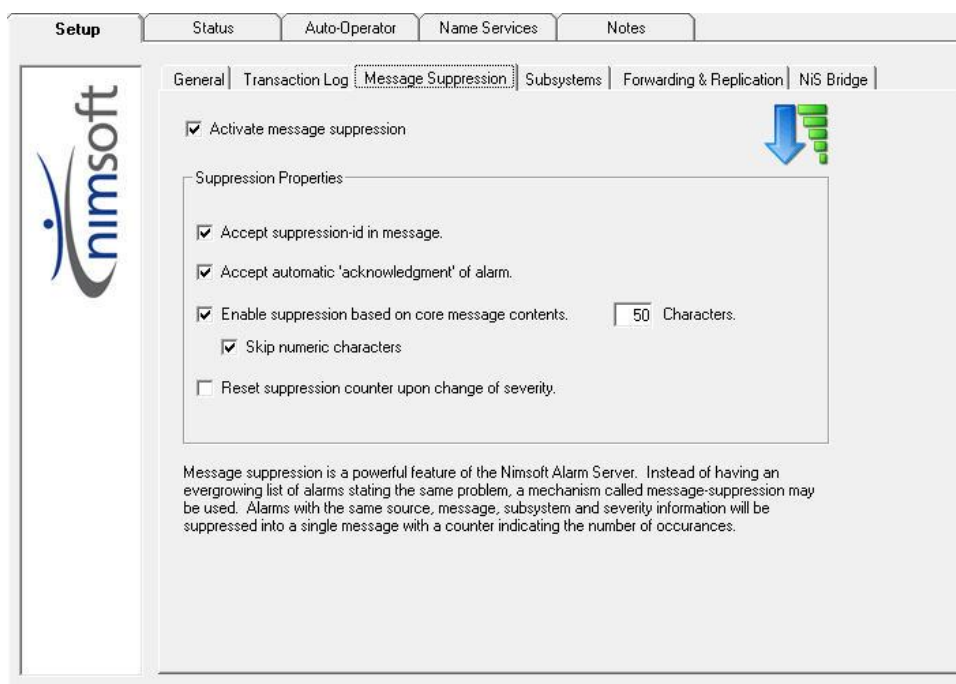
- Click the **Subscribe** tab and select **Subject** as **nas_transaction** and **Table** as **AlarmTransactionLog**.



- Activate the profile, save it, and watch the table gets filled.

Message Suppression

Message suppression is a feature used to avoid storing multiple alarms caused by the same problem. Alarms with the same *source*, *message*, *subsystem* and *severity* information will be suppressed into a single message with only a counter indicating the number of occurrences.



The NAS supports two different message suppression models:

- A model suppressing messages with an exact match on message subsystem id, severity level and message text (standard suppression).
- A model based on a suppression key following the message (note that the following terms may be used, all meaning the same: suppression key, suppression ID and checkpoint ID).

Sometimes an administrator may choose to ignore the suppression mechanism based on suppression key if they want to view the messages as the probes report them. When the key suppression is enabled, messages with matching *suppression key* will be suppressed. This means that the following two messages from the same probe are equal:

Filesystem '/usr' is filled 95% (*supkey*: FsProbe-/usr)
 Filesystem '/usr' is filled 55% (*supkey*: 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 were as displayed (95% first, then 55% as the last status) then the administrator would experience the state as a file-system with 55% filling grade (which is the correct way to see things).

The fields are:

Activate message suppression

When this option is selected, the messages suppression features are activated in order to avoid multiple instances of the same alarm-event.

Accept suppression-id in message

If this checkmark is set, the NAS decides whether a message has occurred before from an internal ID (the suppression key).

If not, the entire alarm text must be absolutely identical for the messages to be considered identical by the suppression mechanism.

Accept automatic 'acknowledgement' of alarm

If this checkmark is set, an alarm with level "clear" will acknowledge and delete all alarms based on the same suppression key.

Enable suppression based on core message content

When this option is selected, alarms where the specified number of characters (e.g. 50) in the message is identical with a message that has occurred before will be suppressed. Note that numbers in the alarm messages does not count, only alpha characters are compared.

This feature is for probes that do not send a suppression-id in the alarm messages, for example *ntevl*.

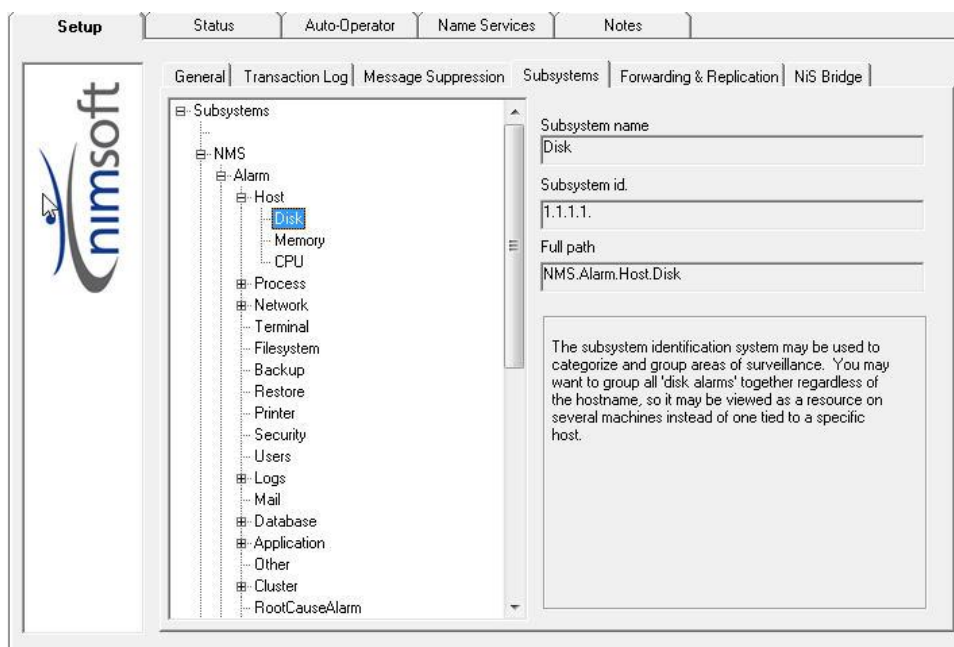
Reset suppression counter upon change of severity

Resets the suppression counter when the severity of an alarm has changed.

Subsystems

A sid (*subsystem identification number*) used to categorize alarms sent along with the alarm message from the probes.

The NAS maps this number to a *string* when the alarm message is received.



The fields are:

<Subsystems list>

A list of all subsystems defined to the nas. By right-clicking here, you may edit the list (add and delete).

Subsystem name

A descriptive text name for the subsystem.

Subsystem id

The number sequence, separated by dots, identifying the subsystem.

Full path

The text sequence.

The figure shows that the NAS maps the subsystem identification number **1.1.1.1** to **Nimsoft.Alarm.Host.Disk**.

This allows for grouping information. The subsystem tab simplifies the management of the subsystem IDs, known to the *nas*. You may add and delete nodes to existing branches, or create new ones. The configuration tool tries to be smart in determining the actual value of the leaf-node, by incrementing the rightmost element. In addition to a context menu on the tree control, you may use the INS/DEL keys on the keyboard. The **sid** list is loaded during startup and restart.

Forwarding & Replication

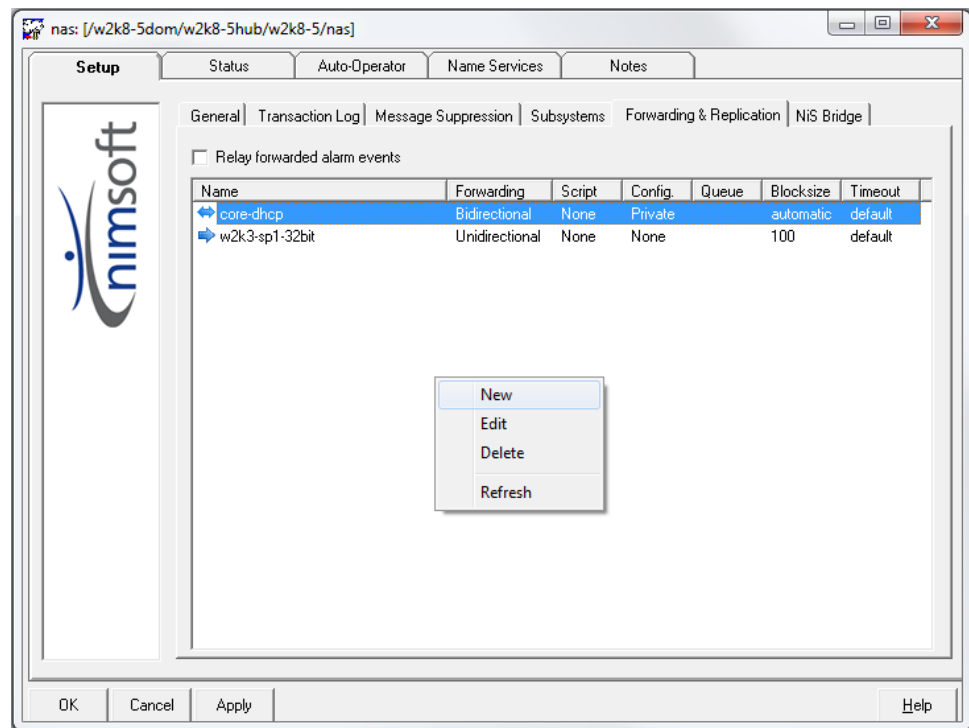
You can define other alarm servers with which you want to exchange alarms and/or scripts. Right-clicking in the list lets you add, edit or delete such connections.

Checking the "Relay forwarded alarm events" option, alarms received from a remote NAS will be forwarded.

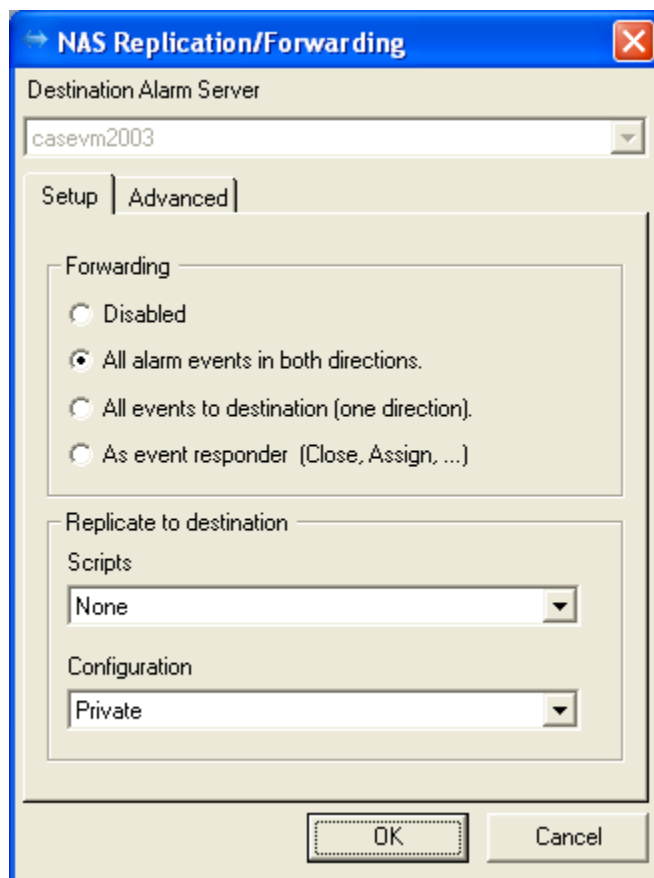
Note 1: When setting up forwarding and replication and making configuration changes on more than one nas, you should first open and edit the GUI for one nas, apply the changes and then exit the GUI. Then you should open and edit the GUI for the next nas, etc. Otherwise the settings may not be saved correctly.

Example: NAS B receives alarms from NAS A, and NAS B forwards alarms to NAS C:

The alarms NAS B receives from NAS A will be forwarded to NAS C only if the "Relay forwarded alarm events" option is set on NAS B.



Note 2: The *Queue* column in the window shows the current number of items (files, messages etc.) in the replication queue, waiting to be processed. The other columns are explained in the table below.



The **Setup** tab fields are:

Destination Alarm Server

Select the destination alarm server from this list.

This is the alarm server with which you want to exchange alarms and/or scripts.

Forwarding

Select the forwarding properties for the selected nas:

Disabled

Disables the selected NAS replication/forwarding profile.

Note: If you remove the forwarding configuration from the destination server and the sending nas is down, the sending nas will retain its forwarding configuration even after it is reactivated. It will continue to send import files, unnecessarily consuming resources on both alarm servers. You must manually remove the configuration from the sending nas.

All alarm events in both directions

All alarm events will be sent to and received from the NAS selected as the destination alarm server.

All events to destination

All alarm events will be sent to the NAS selected as destination alarm server.

As event responder

Allows the NAS selected as destination alarm server to act as an event responder, close and assign alarm messages from the NAS forwarding the alarm messages.

If setting up a queue as "All events to one direction" on NAS A, the queue will appear as "As event responder" on NAS B.

Replicate to destination

Scripts

Select if you want the scripts available on the NAS also be available for the destination NAS defined.

None means not available for the destination alarm server defined.

Private means that scripts will be available on the destination NAS defined, but it can not be modified there (no write access).

Shared means that scripts will be available on the destination NAS defined, in the same script structure as the source NAS, and it is possible to modify the script. Changes will be mirrored between the two NAS's.

NOTE if sharing scripts with a destination NAS:

If modifying a shared script on the destination NAS, you should create a folder where you save the modified script. Otherwise it will be overwritten if the script is modified on the origin NAS (the NAS where the script was created).

Configuration

Select if you want the configuration settings (profiles) available on the NAS also be available for the destination alarm server defined.

None means not available for the destination alarm server defined.

Private means that the NAS configuration file will be available on the destination NAS defined.

The file will be located under the directory

`..\Nimsoft\probes\service\nas\replication\config\<name of the replicated nas server>\nas.cfg`

If you want to use this configuration file on the destination server, you must paste it manually to `..\Nimsoft\probes\service\nas\nas.cfg`.

The **Advanced** tab fields are:

Max. Transfer Blocksize (messages)

This parameter sets the maximum number of messages transferred at each interval. You may select one of the values available, or preferably select automatic (default).

The NAS will then attempt to use a blocksize of 10000 messages. If the NAS fails to send so many messages (after 10 attempts), the blocksize will automatically be divided by 10, and the NAS attempts to transfer 1000 messages. If still problems, the blocksize will again be divided by 10 (to 100). This continues until the NAS succeeds to send the current blocksize.

Then the NAS uses this blocksize for 10 intervals, and then increments the blocksize with 100. If this works OK, the blocksize will again be incremented by 100 for the next 10 intervals. This continues until the highest possible blocksize is reached.

Timeout (seconds)

The sending NAS transfers messages to receiving HUB(s) at regular intervals. This timeout defines the maximum number of seconds the sending NAS attempts to transfer messages to a receiving NAS before starting a new interval.

If using Max. Transfer Blocksize = *Automatic* (see above), the blocksize will be reduced after 10 unsuccessful attempts.

Note 3: Sync issue when disabling and then enabling replication

Example:

- Set up unidirectional replication between two NAS's.
- Send 2 alarms with the same suppression key from the 'sending' NAS.
- Disable the replication.
- Send 3 more alarms with the same suppression key from the 'sending' NAS.

- Enable the replication again.
- Again send 3 alarms with the same suppression key from the 'sending' NAS.

All alarms sent after the replication was disabled (in this example 6 alarms) will be ignored by the 'receiving' NAS.

Solution:

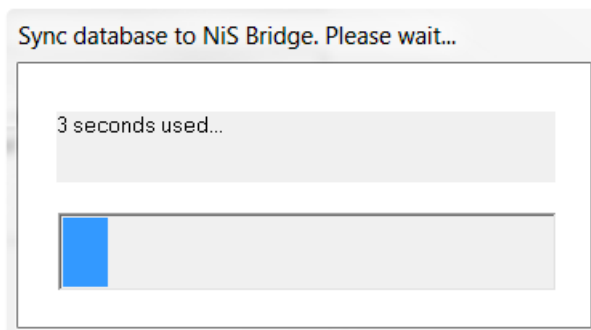
When re-activating replication between two NAS's, you should manually delete all alarms on the 'receiving' NAS that are received from the 'sending' NAS.

NiS Bridge

The **NiS Bridge** tab is visible when the data_engine probe is deployed with nas on the same hub, and communication between the two components is not blocked.

The screenshot shows the 'NiS Bridge' tab selected in the top navigation bar. Below the tabs, there is a checkbox labeled 'Activate NiS bridge' which is checked. Underneath this is a section titled 'Transaction Log Management' enclosed in a box. Inside this box, there are four settings: 'Administration interval' set to 'every hour' via a dropdown menu, 'Compress transactions after' set to '7 days', 'Log transaction details' checked, and 'Keep transaction history' set to '30 days'. To the right of the 'Log transaction details' checkbox, there is a setting 'Keep transaction summary' set to '90 days'. Below the configuration box, there is a descriptive text: 'The NiS bridge duplicates alarm data into the Nimsoft Information Store database. Alarms, notes and transaction data are stored for use by reporting tools.'

Note that when the nas GUI opens, a pop-up window displays for a few seconds to let you know that the NiS Bridge is synchronizing with the database:



By default the **Activate NiS bridge** box is unchecked and NiS Bridge is inactive. Check the box to activate this feature.

Important: The NiS Bridge is designed to have one, and only one, nas populating its database tables (NAS_*). Enabling the NiS Bridge on multiple nas engines is not supported.

The following tables are created in the NiS when **Activate NiS bridge** is checked:

- NAS_ALARMS – the current open-alarm table.
- NAS_TRANSACTION_SUMMARY – the transaction summary table (one row per alarm)

When **Log transaction details** is also checked, these additional tables are created and populated:

- NAS_NOTES – the notes in the system.
- NAS_ALARM_NOTE – the mapping between the note and the alarm.
- NAS_TRANSACTION_LOG – the event transaction table (new,suppressed,close,assign,..).

The NAS logs all transactions to NAS_TRANSACTION_LOG, allowing you to follow the complete message life cycle from the initial message until the message is closed (acknowledged).

These tables are maintained by the NAS using the NiS bridge configuration data. To keep the size of these tables in the NiS database as manageable as possible, they are automatically compressed at the configured administration interval.

The GUI fields are:

Activate NiS Bridge

If this checkmark is set, the NAS logs all steps in the life of an alarm (the alarm transaction) from the time the alarm is generated until it is acknowledged. This data is stored in the NiS database.

Transaction Log Management

Administration interval

The interval at which the NAS monitors the size of the transaction log tables in the NiS database and truncates them.

Valid options are:

- Every hour
- Every 2 hours
- Every 6 hours
- Every 12 hours

- Daily

Compress transactions after

The events (of type *suppression*) for alarms stored in the NiS database will be deleted after the number of days specified. Default is 7 days.

Keep transaction history

How long (in days) the transaction history is stored. The transaction history stores all events for each of the alarms handled by the NAS in the NiS database.

Keep transaction summary

How long (in days) the transaction summary is stored. The default value is 30 days.

The transaction summary for each alarm is stored as one row in the NiS database.

Log transaction details

This option specifies how often duplicate NAS events are stored in the NiS database.

A message is considered duplicate when message text, subsystem id and severity are equal to the previous message with the same suppression key. This will reduce the size of the transaction tables in the NiS database and speed up transaction queries.

If the *Log transaction details* option is not checked, the transaction details will not be stored and the tables in the database will be empty.

Note: You must use SQL queries to get the transaction alarms from the NiS database.

The Status Tab

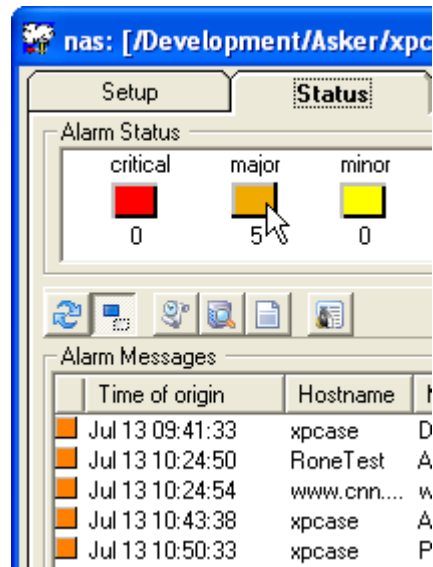
The **Status** tab displays the operator status information from the selected nas. The Alarm Server is queried for:

- nas software version information.
- Alarm Status (summary information showing the number of alarms with the different severity levels and the total number of alarms received).

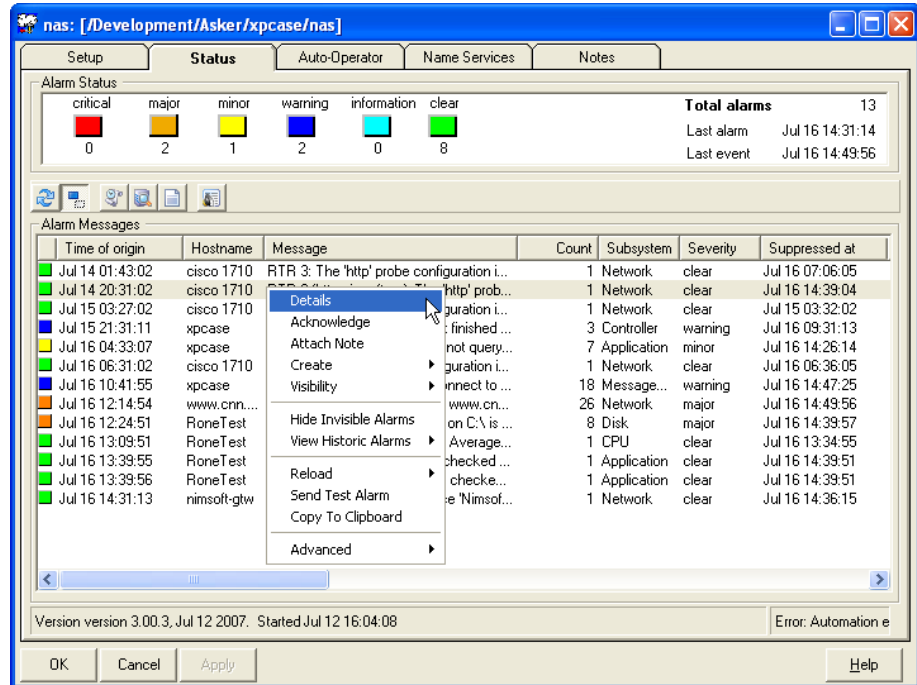
Double-clicking one of the icons in the Alarm status window, e.g. **major**, will filter the alarms, and only alarms with severity level major will be listed in the Alarm messages list.



Clicking the Refresh button updates the alarm messages list.



- The current alarm list.



A command menu is available when you right-click in the alarm-messages list. You can perform various administrative tasks from this menu, such as sending test-alarms, viewing alarm details, acknowledging alarms, viewing historic events etc. The *count* column in the alarm-messages list indicates that the alarm has been received *n* times.

For details on how to manage the alarms, see the section The Alarm List.

Notifications (events)

The Alarm Server notifies the world about changes to its alarm database by issuing event-messages to the Infrastructure Manager Alarm window. 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. The Alarm Server will generate a statistical event message, **alarm_stats**, containing the summary information (on severity level) for all open alarms.

The Auto-Operator Tab

The Auto-Operator (AO) feature is meant to aid the administrator in managing the alarm database. You can define various profiles, based on matching rules (such as severity level, alarm message text, subsystem ID) to:

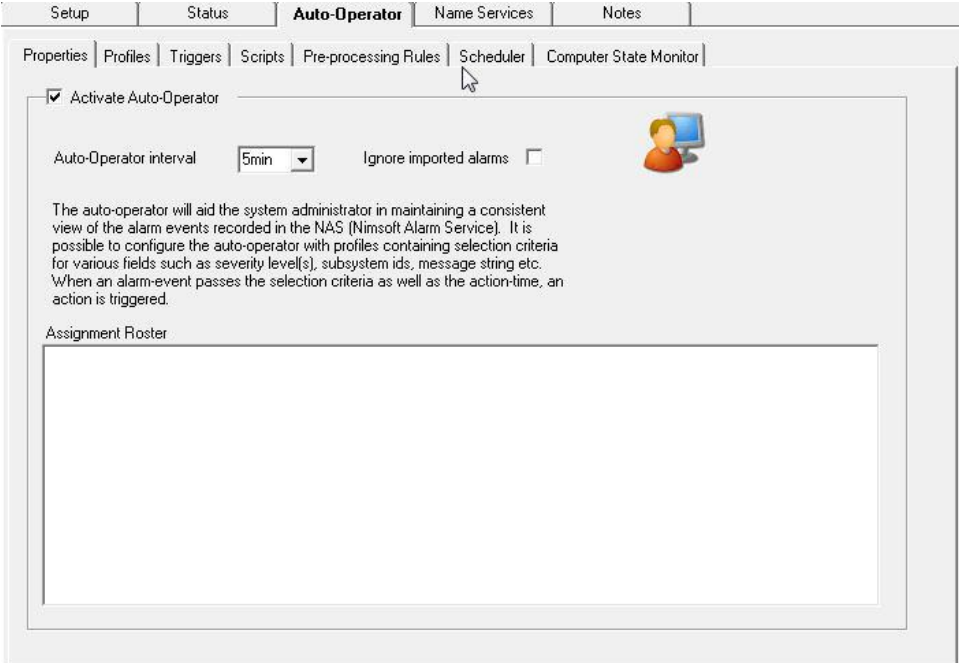
- Close/acknowledge certain alarms.
- Automatically assign an alarm to a person/group.
- Send an e-mail or a GSM/SMS message whenever a specific rule is met.

Properties

The auto-operator will aid the system administrator in maintaining a consistent view of the alarm events recorded in the NAS (Nimsoft Alarm Service).

On the *Auto-Operator* tab (see the section The Setup Tab), it is possible to configure the auto-operator with profiles containing selection criteria for various fields, such as severity level(s), subsystem ID, message string etc. When an alarm event passes the selection criteria as well as the action time, an action is triggered.

The properties on this *Setup > Auto Operator* tab let you activate the Auto-Operator. When not checked, the Auto-Operator tab will be disabled (greyed out).



The screenshot shows the 'Auto-Operator' tab in the 'Setup' section of the Nimsoft Alarm Service interface. The 'Activate Auto-Operator' checkbox is checked. The 'Auto-Operator interval' is set to '5min'. The 'Ignore imported alarms' checkbox is unchecked. A small icon of a person at a computer is visible. Below these settings, there is a text box containing the following text: 'The auto-operator will aid the system administrator in maintaining a consistent view of the alarm events recorded in the NAS (Nimsoft Alarm Service). It is possible to configure the auto-operator with profiles containing selection criteria for various fields such as severity level(s), subsystem ids, message string etc. When an alarm-event passes the selection criteria as well as the action-time, an action is triggered.' At the bottom, there is a section labeled 'Assignment Roster' with an empty table below it.

The fields are:

Activate Auto-Operator

Enables/disables the auto-operator feature.

When this option is selected, the Auto-Operator sections are enabled.

Auto-Operator interval

This is a global interval that can be used by AO profiles and AO scheduler.

Ignore imported alarms

The Auto-operator ignores alarms imported from other nas probes. See Forwarding & Replication for more information.

Assignment roster

The *Assignment roster* list allows you to specify assignment targets, such as a helpdesk. By adding a target to this list, you will be able to assign alarm(s) to the target from within the Auto-Operator.




You add new assignment targets by right-clicking in the window and selecting **New**. A new target will by default be named *New operator*. Select the new target, right-click and select **Rename** and type a name of your own choice.

The right-click menu also allows you to delete assignment targets.


Profiles

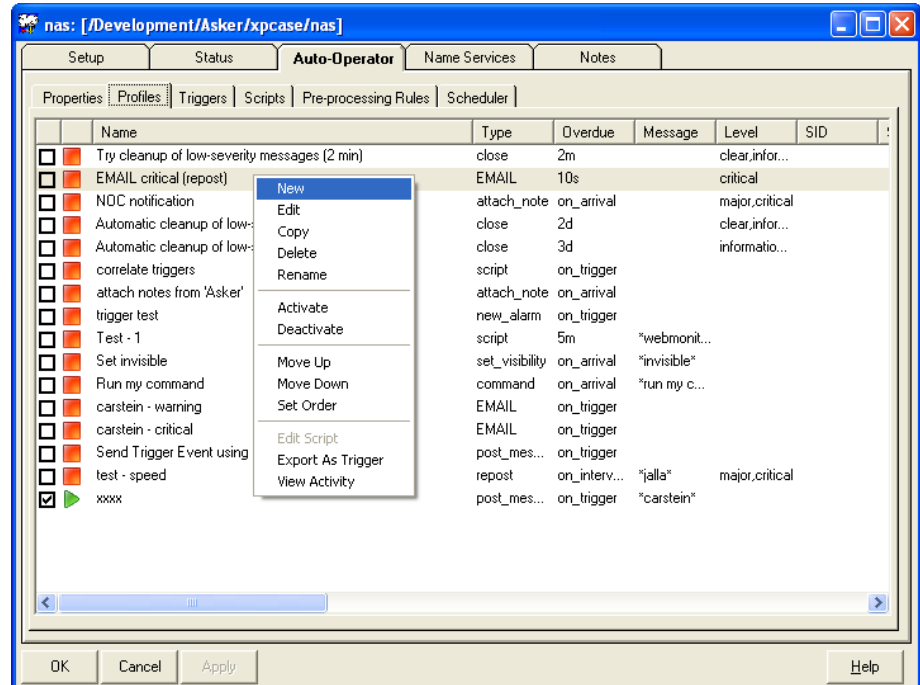
This tab lists all Auto-operator defined profiles.

Note the different icons for the defined profiles in the list:

- Not activated
 EMAIL critical (repost)
- Activated
 testing
- Not manually activated, but activated by a calendar profile defined on the Calendar tab
 assign to cseeberg

- Manually activated, but deactivated by a calendar profile defined on the Calendar tab

 Attach Millennium



You can perform the following actions from the right-click menu:

New

Allows you to create a new profile.

Edit

Allows you to edit the selected profile properties.

Copy

Copies the selected profile and allows you to rename the new profile.

Delete

Deletes the selected profile. A confirmation screen appears.

Rename

Allows you to rename the selected profile.

Activate

Activates the selected profile.

Deactivate

Deactivates the selected profile.

Move Up

Moves the selected profile one place up in the list.

Move Down

Moves the selected profile one place down in the list.

Set Order

The profiles will be executed in the order they were created (see the order column in the list). If modifying the list, using the move up and move down options mentioned above, you may select this to be the new executing order by selecting the Set Order option.

Edit Script

This option is activated only if a script is selected to be run on match in the profile, otherwise the option is not enabled.

The script defined in the profile will be opened in the script editor (see the section The Script Editor).

Export as Trigger

The properties defined in the Matching Criteria section of the profile will be exported as a Trigger.

The Matching Criteria section of the profile properties dialog opens. Edit the properties if needed and click **OK**.

Enter the new trigger name, then click **OK**.

The trigger will appear in the list under the Triggers tab.

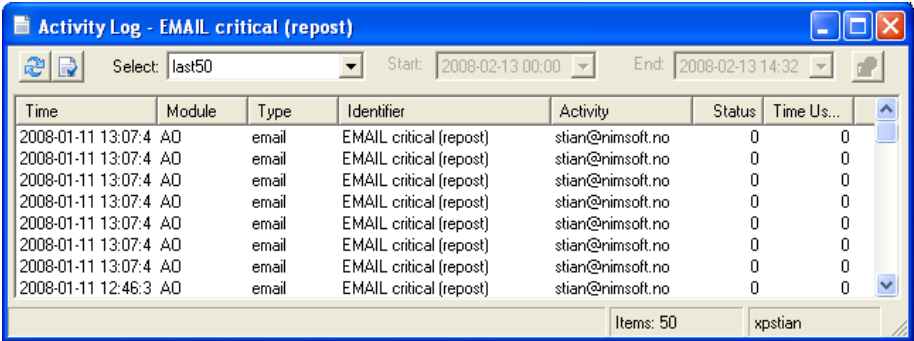
View Activity

Opens the activity log, where you can view the activity for the profile for a specific time frame.

Valid options are:

- Today
- Last hour
- Last week
- Last month
- Last 24 hours
- Last 7 days
- Last 3 days
- Last 50 days
- Select a date

This option allows you to specify a start day and an end day for the period.

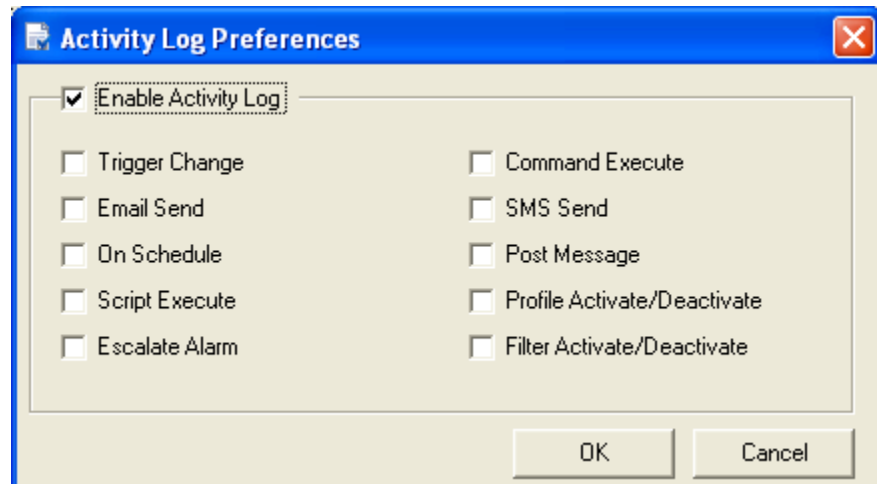


The screenshot shows a window titled "Activity Log - EMAIL critical (repost)". It has a toolbar with a refresh icon and a "Select:" dropdown menu set to "last50". To the right of the dropdown are "Start:" and "End:" date pickers. The "Start:" date is "2008-02-13 00:00" and the "End:" date is "2008-02-13 14:32". Below the toolbar is a table with the following columns: Time, Module, Type, Identifier, Activity, Status, and Time Us... (likely Time Used). The table contains 8 rows of data, all showing "EMAIL critical (repost)" activity from "stian@nimsoft.no" with a status of "0" and "0" time used. The last row has a time of "2008-01-11 12:46:3". At the bottom right of the window, it says "Items: 50" and "xpstian".

Time	Module	Type	Identifier	Activity	Status	Time Us...
2008-01-11 13:07:4	AO	email	EMAIL critical (repost)	stian@nimsoft.no	0	0
2008-01-11 13:07:4	AO	email	EMAIL critical (repost)	stian@nimsoft.no	0	0
2008-01-11 13:07:4	AO	email	EMAIL critical (repost)	stian@nimsoft.no	0	0
2008-01-11 13:07:4	AO	email	EMAIL critical (repost)	stian@nimsoft.no	0	0
2008-01-11 13:07:4	AO	email	EMAIL critical (repost)	stian@nimsoft.no	0	0
2008-01-11 13:07:4	AO	email	EMAIL critical (repost)	stian@nimsoft.no	0	0
2008-01-11 13:07:4	AO	email	EMAIL critical (repost)	stian@nimsoft.no	0	0
2008-01-11 12:46:3	AO	email	EMAIL critical (repost)	stian@nimsoft.no	0	0

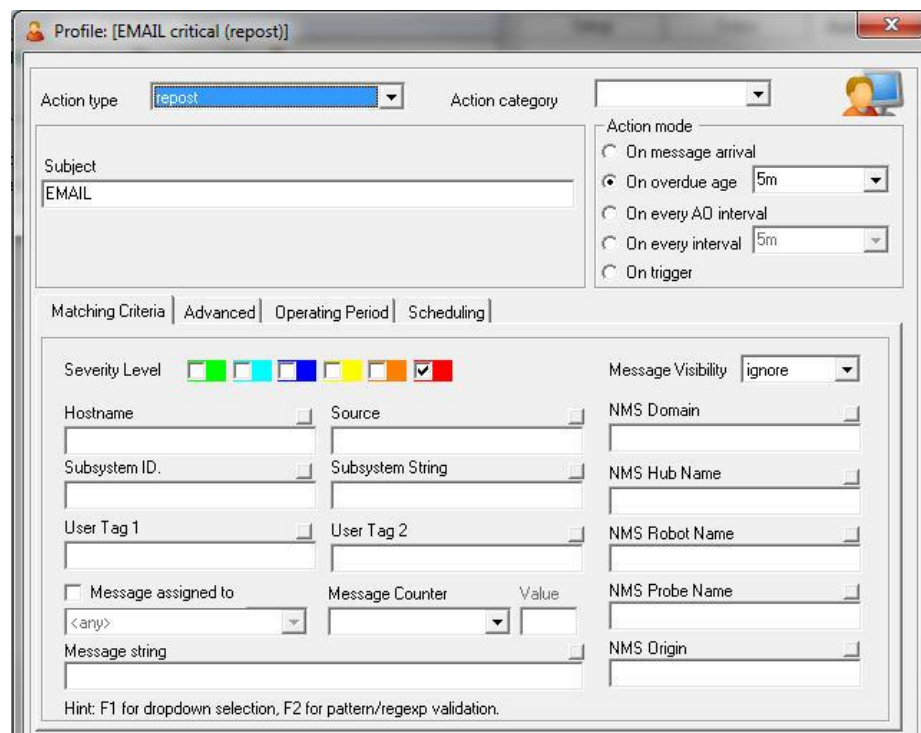


Clicking the *Preferences* button opens the *Activity Log Preferences* dialog, enabling you to select additional items to be logged in the Activity log. Click **Enable Activity Log** option to enable the options on the dialog.



Creating or Editing a Profile

The Profile dialog allows you to create or edit the profile properties.



The fields are:

Action type

Specify what the auto-operator does when receiving an alarm meeting the matching criteria specified for this profile.

Important! The options in the section below the action type field change depending on the action type selected from the drop down menu.

The options are:

assign

Assigns alarms matching the filters below to an operator selected from the pull-down menu.

Select the username you want to assign the alarm to in the section below the action type field.

These alarms will also be sent to recipients defined in profiles in the emailgw probe.

attach_note

Attaches a note to the alarm. Select the note in the section below the action type field. Available notes are defined under the Notes tab (see The Notes Tab for more information).

close

Closes / acknowledges alarms matching the filter(s) defined in the matching criteria section of the screen. This is useful for removing old alarms etc.

command

Executes the specified command locally. This may be a command beeping on a pager, or something that actually tries to fix the problem the alarm reports about.

Enter the command in the section below the action type field.

EMAIL

Sends the alarm as an e-mail destined for the emailgw when the defined alarms matching criteria are met.

Note that you are allowed to use a comma-separated list of e-mail addresses.

- *If an e-mail address is specified in the recipient field:*

The alarm is sent as an e-mail to the specified e-mail address.

- *If a profile (defined in the emailgw) is specified in the recipient field:*

The NAS checks the profile in the emailgw to find the e-mail address defined for the profile, and the alarm is sent as an e-mail to that address.

- *If the recipient field is empty:*

The alarms matching the criteria will be collected in a report.

This report will be sent as an e-mail to profiles defined as *Report recipients* in the emailgtw at regular intervals (approximately 5 minutes).

Enter the recipient, subject and message body in the section below the action type field.

escalate_level

Escalates the severity level of the alarm(s) matching the filter criteria selected below. The severity level is incremented to the next level.

Note that the options on the *Advanced tab* are grayed out and can not be selected if this action type is selected.

new_alarm

Composes and sends a new alarm message onto the Nimsoft. The message field accepts expansion.

The alarm may contain the following optional elements in addition to a message body and severity level:

- Subsystem ID.
- Source.
- Suppression id.

post_message

Alarms will be posted as a message with the specified subject and message text.

Enter the subject and message text in the section below the action type field.

If you enter the subject as EMAIL, the alarms with the message body specified will also be sent to recipients defined in the profiles in the *emailgtw* probe.

repost

Retransmits the alarm message under another subject ID. This may be useful if you want to send special alarm messages to your 'own' correlation engine.

Select the subject ID in the section below the action type field.

If you enter the subject EMAIL, these alarms will also be sent to recipients defined in the profiles in the *emailgtw* probe.

script

Executes the script specified.

Select the script to execute and any parameters in the section under the action type field.

These scripts are defined in the Scripts section. Use the Script editor to create and edit these scripts (see the section The Script Editor).

set_visibility

Select the visibility for the alarm message.

This mode is a filter type for incoming alarms under the pre-processing tab. Alarm messages set to invisible will be managed by the NAS and will be listed under the Status tab, provided that the option *Show Invisible Alarms* (on the Status tab) is selected. However, they will not be visible in the Alarm sub console in Infrastructure Manager and Enterprise Console, unless the ACL the Nimsoft user is associated with allows the user to see the invisible alarms.

Select the visibility option in the section below the action type field:

- Make event invisible
- Make event visible (if invisible).

SMS

Composes and sends a message destined for the SMS gateway. The message field accepts field expansion.

Enter the phone number and message in the section below the action type field.

Action category

Select the action category to be used. The action category is a way to group profiles to ease the administration of multiple profiles. You can create new categories by placing the cursor in the field and typing a new category.

Action mode

On messages arrival

Performs the selected action immediately when the alarms arrive.

Note that this time setting is disabled for some of the actions (close, command, new_alarm and escalate_level), as it is not advisable to perform these actions if the same alarm message (with the same source, sub-system and severity) arrives hundreds of times (see also *Message suppression*).

On overdue age

Performs the selected action when the age of the alarm exceeds the specified threshold. Select one of the predefined values in the list or type another value of your own choice (use the same format as used in the list).

On every AO interval

Performs the selected action on every Auto Operator check interval.

On every interval

Performs the selected action on every interval specified. Select one of the predefined values in the list or type another value of your own choice (use the same format as used in the list).

On trigger

Select this option when the *Trigger* mode is selected in the action category.

The lower portion of this screen displays the options for setting triggers, see Setting Triggers for more information.

You can select one or more triggers. The Auto operator performs the selected action immediately when the trigger specified is true.

Example:

Provided that the profile is not de-activated due to *operating period* and/or *scheduler* settings:

You select *Action type* = script and *Action mode* = trigger. You select a *script* to be executed and a *trigger* to trigger the action.

Imagine that the properties dialog for the selected trigger is set to trigger on *Message string* *Oslo*, the selected script will be run as soon as an alarm message containing the word "Oslo" in the message text appears.

Note that this choice will restrict the number of *Action types* available.

Matching Criteria

This tab will appear only if an Action mode other than Trigger is selected.

Note that *Matching criteria* fields marked with a '*' can be inverted (set to *NOT Expression*) by clicking the small button above each of the fields. The buttons becomes activated as soon as the field contains information.

Note: Pressing **F1** in the Matching Criteria fields (except for the Message field), will list all parameters available. Pressing **F2** in the Matching Criteria fields gives you the option to specify a target string, a pattern/regular expression and test that it works before selecting it. See Using F1 and F2 in Matching Criteria Fields for more information.

Severity level

Select the severity level(s) for the alarms you want to be treated by this profile.

Message Visibility

Select the matching criteria to be valid for:

- Ignore (both visible and invisible alarm messages)
- Visible
- Invisible

* Hostname

Specify the hostname (string matching) sending alarms to be treated by this profile.

* Source

Specify the source (string matching) for alarms to be treated by this profile.

* NMS Domain

Specify the name of the Domain (string matching) the host sending alarms to be treated by this profile belongs to.

* Subsystem ID

Specify the subsystem ID (string matching) for alarms to be treated by this profile.

* Subsystem String

Specify the subsystem (string matching) for alarms to be treated by this profile.

* NMS Hub Name

Specify the name of the Hub (string matching) the host sending alarms to be treated by this profile belongs to.

User Tag 1

User-defined tag in the Controller probe. To be used as a grouping / locating mechanism.

User Tag 2

User-defined tag in the Controller probe. To be used as a grouping / locating mechanism.

*** NMS Robot Name**

Specify the name of the Robot (string matching) sending alarms to be treated by this profile belongs to.

Message assigned to

Select this option to send a message to a specific user.

Specify the user name to which alarms are assigned to be treated by this profile.

Message counter

Specify the number of times an alarm must be received to be treated by this profile. Select *Less than*, *Equal to* or *Greater than* from the drop-down list and insert a number in the value field.

*** NMS Probe Name**

Specify the probe (string matching) sending alarms to be treated by this profile.

Message string

Specify a text string found in alarms (string matching) for alarms to be treated by this profile.

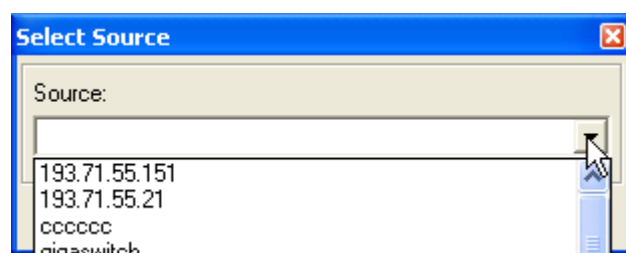
*** NMS Origin**

Specify the origin (string matching) sending alarms to be treated by this profile.

Using F1 and F2 in Matching Criteria Fields

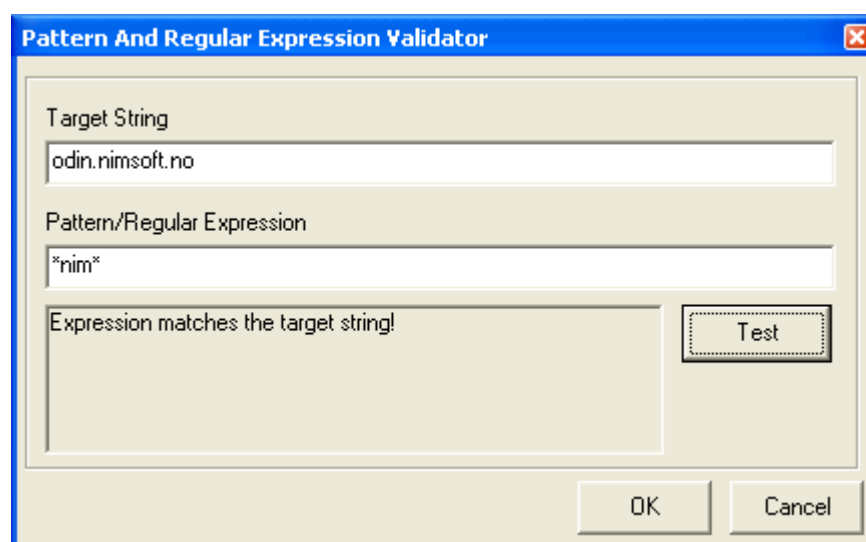
Pressing **F1** in the Matching Criteria fields (except for the Message field) displays the Select Source screen. This screen displays a list of parameters retrieved from the transaction summary table. This table contains entries for alarms (such as hostname, source origin) processed recently.

Select an entry in the list or type another source.



Pressing **F2** in a Matching Criteria field displays the Pattern and Regular Expression Validator screen.

Type the string you want to use as the target string and the pattern/regular expression you want to use. Click the **Test** button and verify that the output is "Expression matches the target string". Clicking the **OK** button, the pattern/regular expression will be inserted into the field.



Note: If creating an AO profile, using *Message assigned to* as matching criteria, the profile will trigger when an alarm has been assigned to a user or a group by another AO profile. Note that profiles using the *Message assigned to* should NOT use Action mode *On arrival* to ensure that the processing acts as expected.

IMPORTANT:

All filters take pattern matching and regular expressions. You may also combine patterns/strings using a comma, for example:

host1 , host2 matches host1 or host2 as two separate patterns

host1 & host2 matches host1 or host2, but as a single pattern

If you specify multiple mask criteria, all of them must be fulfilled for the action to be performed.

Field expansion is made available to some of the action fields. This feature is triggered by typing a dollar sign (\$) and then wait for 1 sec. A list of available field codes is listed, and may be used to create your own mixture.

String matching is accomplished with a mixture of pattern matching rules and/or regular expressions.

Profile Advanced

The options on this tab are available only if an Action mode other than *on trigger* is selected and apply only when *Severity level Clear* is selected as a matching criteria.

Note: The options on this tab are grayed out and cannot be selected if Action type *escalate_level* is selected for the profile.

Ignore User Acknowledgment (Delete)

Alarms acknowledged by a user will be ignored and not trigger any action from the profile.

Expect the previous severity level to be part of the severity filter

When an alarm with severity level *Clear* occurs, the previous severity level of the alarm must also be a part of the matching criteria to trigger an action from the profile.

Example:

An alarm with severity level *Clear* occurs. Severity level *Critical* and *Clear* is selected as matching criteria, and the previous severity level of the alarm was *Warning* (which is not selected as matching criteria). No action will be triggered.

Same example as above, but the previous severity level of the alarm was *Critical* (which is selected as matching criteria).

The selected action will be triggered by the profile.

Execute action on user acknowledgment of open 'clear' messages

When checked, the profile is executed when a user performs alarm acknowledgment of open 'clear' messages.

Skip further profile checks on match

This option is available in case of multiple profiles. Normally all the auto-operator profiles will be checked during the filter validation. Selecting this checkbox will stop the further checks.

Note: It is also a good idea to use "ordering" to ensure that the proper order is maintained for checking the AO profiles.

Profile Operating Period

Defines the time-slots when the profile is active within a week. The Operating Period settings are closely related to the Scheduling settings.

See the section [Setting an Operating Period](#) for more information.

Profile Scheduling

The scheduling profiles available (these are defined and listed under the Auto-Operator > Scheduler tab) can be used to administer the running properties for the Auto-Operator profiles. The scheduler profiles can activate or deactivate Auto-operator profiles for defined time periods.

All scheduling profiles defined will be listed here. Activate the one(s) you want to administer the running properties for the selected Auto-operator profile.

Setting Triggers

This tab will appear only if Action mode *on trigger* is selected.

The Matching criteria tab disappears when Action Mode *on trigger* is selected and is replaced by the Trigger tab. You can select one or more of the triggers defined to act as a matching criteria for the profile.

The screenshot shows a configuration window with three tabs: 'Triggers', 'Operating Period', and 'Scheduling'. The 'Triggers' tab is active. It contains three radio buttons for logical operations: 'Use boolean 'AND'' (selected), 'Use boolean 'OR'', and 'Use result of the boolean expression'. To the right are two checkboxes: 'Activate when state changes.' and 'Activate on change in trigger alarmlist.'. Below these are two list boxes: 'Available triggers' and 'Selected triggers'. The 'Available triggers' list contains three items: 'example.network', 'example.system', and 'example.SLM', each with a small icon to its left. Between the two list boxes are two icons: a plus sign (+) and a minus sign (-) with an 'x' inside. The 'Selected triggers' list is currently empty.

If using more than one trigger, you can use Boolean AND or OR, or you can build your own Boolean expression.

Activate when state changes:

The profile will be executed when the trigger changes its state (false/true).

Please note that the term *state* in the option "Activate when state changes" refers to the state of the *Auto Operator Profile* and not to the selected triggers within the Auto Operator Profile.

Activate on change in trigger alarmlist:

The profile will be executed when the alarmlist (the alarm(s) that sets the trigger to true) changes, for instance when an alarm changes from severity *major* to *critical*.

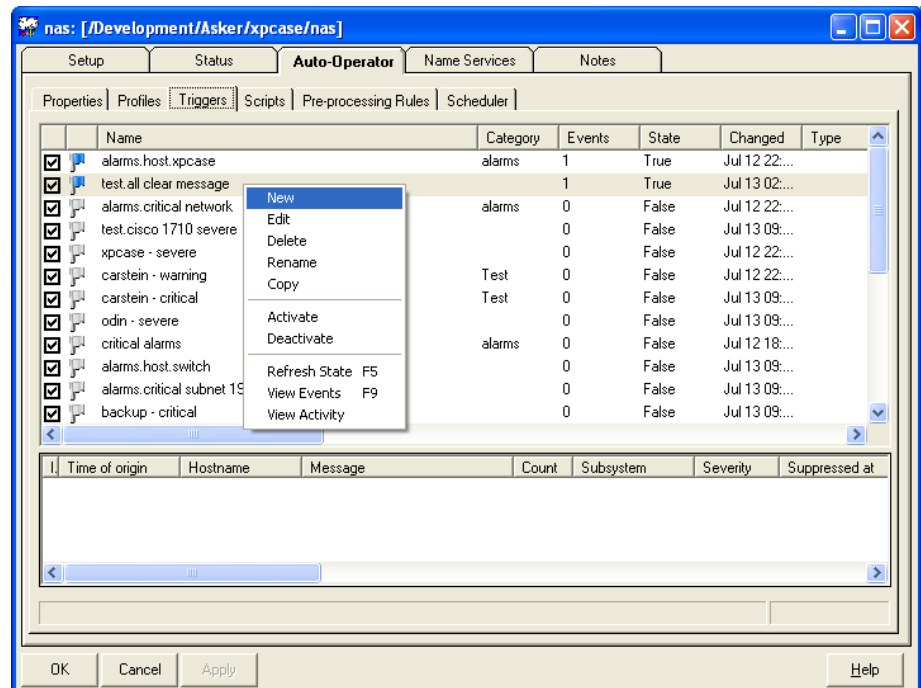


Select the triggers you want to use from the list of available triggers and add them, using the **Add** button.

Triggers

This tab allows you to define triggers to "sort" alarm messages based on properties set for the trigger. That means that alarms matching the criteria defined for the trigger will not be handled by an Auto-operator.

You can define triggers, using matching criteria (such as message text, severity, hostname etc.) and time restrictions (defining the periods when the filter should be active).



Note the following icons



Alarm events have occurred.



Alarm events have NOT occurred.

Right-clicking in the list gives you the following options:

- **New**

Opens the Trigger properties dialog, allowing you to create a new trigger.

- **Edit**

Opens the Trigger properties dialog for the selected trigger, allowing you to edit the trigger properties.

- **Delete**

Deletes the selected trigger.

You are asked to confirm the deletion.

- **Rename**

Allows you to give the selected trigger another name. The current trigger name is highlighted; just type the new name.

- **Copy**

Makes a copy of the selected trigger. Rename the new trigger, using a descriptive name. If you want to modify the properties, you just double-click it and make your modifications.

- **Activate**

Activates the selected trigger.

- **Deactivate**

Deactivates the selected trigger.

- **Refresh State**

Refreshes the list to display the most current contents.

- **View events**

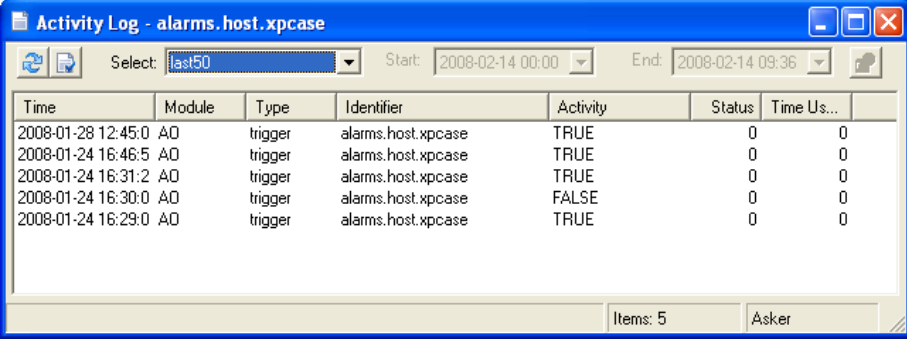
The column Events in the list of triggers shows how many events have occurred for the trigger. When you select this option, the events will be listed in the lower window.

- **View Activity**

Opens the activity log for the selected trigger.

You can select a period, such as last hour, last week, last month etc., or you can specify a time range (start day and end day).

The activity log shows information like: when the trigger was run, the time used, the status etc.



The screenshot shows a window titled "Activity Log - alarms.host.xpcase". It has a toolbar with a refresh icon and a "Select:" dropdown menu showing "last50". To the right of the dropdown are "Start:" and "End:" date/time pickers. Below this is a table with the following data:

Time	Module	Type	Identifier	Activity	Status	Time Us...
2008-01-28 12:45:0	AO	trigger	alarms.host.xpcase	TRUE	0	0
2008-01-24 16:46:5	AO	trigger	alarms.host.xpcase	TRUE	0	0
2008-01-24 16:31:2	AO	trigger	alarms.host.xpcase	TRUE	0	0
2008-01-24 16:30:0	AO	trigger	alarms.host.xpcase	FALSE	0	0
2008-01-24 16:29:0	AO	trigger	alarms.host.xpcase	TRUE	0	0

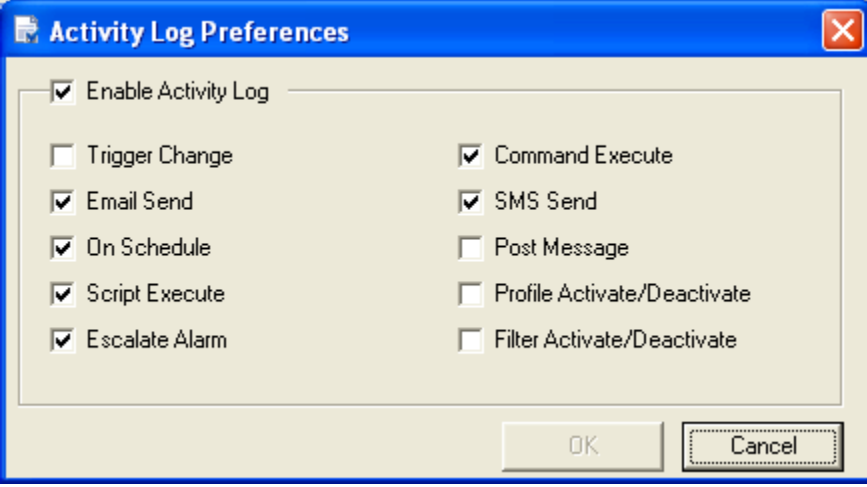
At the bottom right of the window, it says "Items: 5" and "Asker".



Clicking the *Refresh* button refreshes the list to reflect the most updated contents.



Clicking the *Preferences* button opens the *Activity Log Preferences* dialog, enabling you to select additional items to be logged in the Activity log. Click the *Enable Activity Log* option (by default not selected) to enable the options on the dialog. Otherwise the options are deactivated (greyed out).



The screenshot shows a dialog box titled "Activity Log Preferences". It has a checkbox labeled "Enable Activity Log" which is checked. Below this are two columns of checkboxes, some of which are checked and some are greyed out:

<input type="checkbox"/> Trigger Change	<input checked="" type="checkbox"/> Command Execute
<input checked="" type="checkbox"/> Email Send	<input checked="" type="checkbox"/> SMS Send
<input checked="" type="checkbox"/> On Schedule	<input type="checkbox"/> Post Message
<input checked="" type="checkbox"/> Script Execute	<input type="checkbox"/> Profile Activate/Deactivate
<input checked="" type="checkbox"/> Escalate Alarm	<input type="checkbox"/> Filter Activate/Deactivate

At the bottom right are "OK" and "Cancel" buttons.

Adding new triggers

If you right-click in the triggers list and select **New** in the menu, a dialog box appears where you can define a new trigger by filling out some of these fields.

Matching Criteria: Category: example

Severity Level: ☐ ☐ ☐ ☐ ☒ ☒ ☐

Message Visibility: ignore

Hostname:

Source:

NMS Domain:

Subsystem ID.:

Subsystem String:

NMS Hub Name:

User Tag 1:

User Tag 2:

NMS Robot Name:

☐ Message assigned to:

Message Counter:

Value:

NMS Probe Name:

NMS Origin:

Message string:

Hint: F1 for dropdown selection, F2 for pattern/regexp validation.

You can use F1 and F2 in the matching criteria fields, see the Using F1 and F2 in Matching Criteria Fields section for more information.

Note: If more than one of the filtering criteria is specified, all of them must apply for the alarm to be treated by the trigger (logical AND).

The fields are:

Severity Level

Select the severity level(s) for the alarms you want to be treated by this trigger. Only alarms with the specified severity level are collected by this trigger.

Category

Select the category to be used. Category is a way to group triggers to ease the administration in case of many triggers. You may create new categories by placing the cursor in the field and typing a new category.

Hostname

Specify the name of the host (string matching) sending alarms to be treated by this trigger.

Source

Specify the source (string matching) for alarms to be treated by this trigger.

Subsystem ID

Specify the subsystem ID (string matching) for alarms to be treated by this trigger.

Subsystem string

Specify the subsystem string (string matching) for alarms to be treated by this trigger.

User Tag 1

User-defined tag in the Controller probe. To be used as a grouping / locating mechanism.

User Tag 2

User-defined tag in the Controller probe. To be used as a grouping / locating mechanism.

Message assigned to

Select this option and select the Nimsoft user the alarm is assigned to.

Message string

Specify a text string (string matching) found in alarms to be treated by this trigger.

Message Counter

Specify the message counter as a value either less than, equal to or greater than the value specified.

Domain

Specify the name of the Domain (string matching) the host sending alarms to be treated by this trigger belongs to.

Hub name

Specify the name of the Hub (string matching) the host sending alarms to be treated by this trigger belongs to.

Robot name

Specify the robot (string matching) sending alarms to be treated by this trigger.

Probe name

Specify the probe (string matching) sending alarms to be treated by this trigger.

Origin

Specify the origin (string matching) sending alarms to be treated by this trigger. QoS data from probes are tagged with a name to identify the origin of the data.

If specifying an origin name in the controller probe, this name will be used to identify the origin of the data.

If not, the Hub name will be used.

Note however, that the *Origin* field under *Advanced Settings* in the Hub GUI lets you specify an origin name of your own choice to be used, rather than the Hub name.

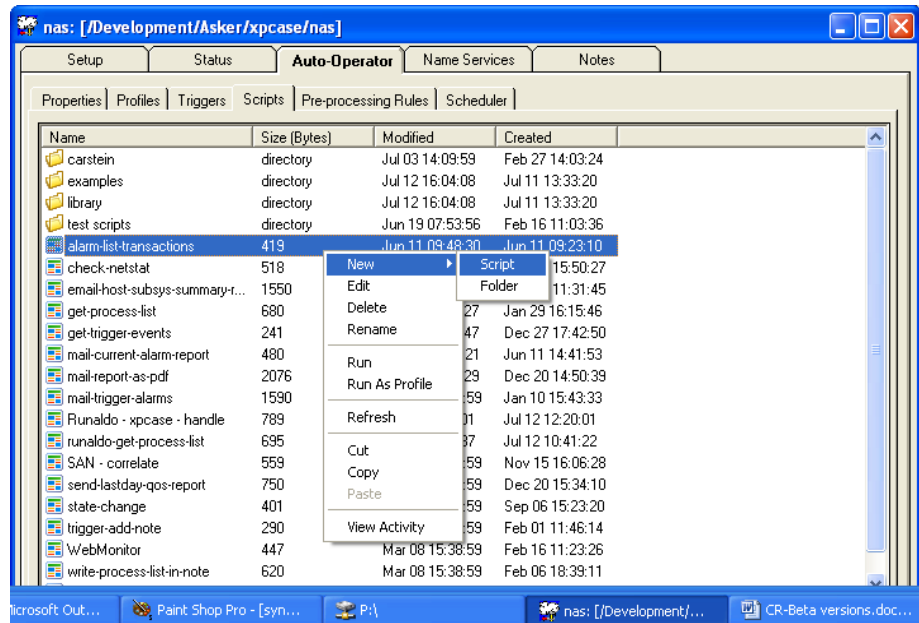
Scripts

Scripts can be used by the Auto Operator when processing alarm messages matching the criteria defined for the Auto Operator profile.

Important! If this tab is not enabled you must go to the distsrv probe configuration on the primary hub, select the 'Forwarding' tab, and set up a new record. The hub where you are trying to set up auto-operator must be selected as the Server field, and the Type set to Licenses. Once this has been applied and saved on the primary hub distsrv, the user must wait approximately five minutes, then restart the NAS on the remote hub. The scripts tab should now be available.

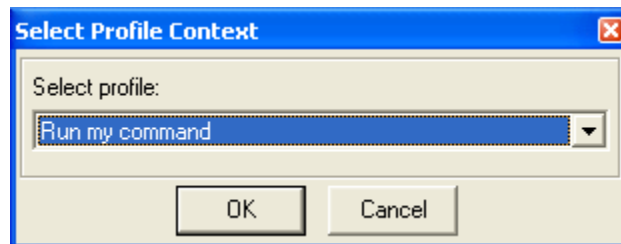
Use scripts when processing alarm messages matching the criteria defined for the Auto Operator profile. Scripts can also be run by the Scheduler and by the pre-processing filters.

Use the Script Editor (see the section The Script Editor) to create and edit these scripts, using the Lua scripting language. The scripts can also be grouped in folders, created by right-clicking in the list, selecting *New > Folder*.



Right-clicking in the list displays the following options:

- **New**
Opens the Script properties dialog, allowing you to create a new script, or a new folder where you can group scripts.
- **Edit**
Opens the properties dialog for the selected script, allowing you to edit the script properties.
- **Delete**
Deletes the selected script.
You are asked to confirm the deletion.
- **Rename**
Allows you to give the selected script another name. The current script name is highlighted; just type the new name.
- **Run**
Executes the selected script. The execute result will appear on the screen. Click the **OK** button to exit the pop-up.
- **Run as Profile**
Allows you run the selected script as a profile. A dialog pops up, letting you select the profile which you want the script to run as.



- **Refresh**

Refreshes the list to reflect the most current content.

- **Cut**

Allows you move the selected script to another location (for example from the root level to a group, from a group to the root level, or from one group to another group).

The script will be copied to the clipboard. Use the **Paste** command to paste the script to the new location.

- **Copy**

Makes a copy of the selected script. The selected script will still be present. The script will be copied to the clipboard. Use the **Paste** command to paste a copy of the script in the list. The name of the copy will be "Copy of <script name>".

- **Paste**

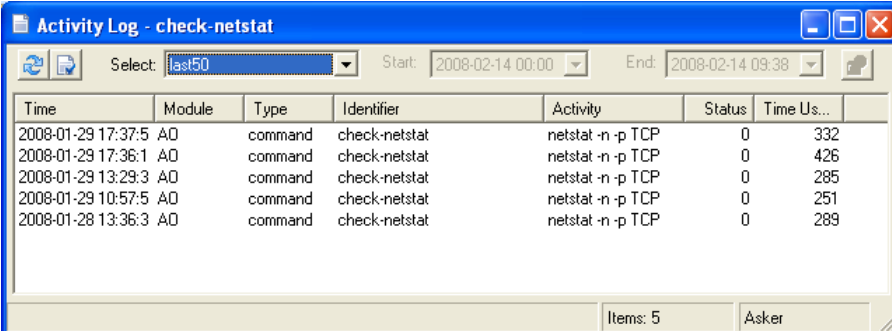
Allows you to paste the contents of the clipboard to the current location.

- **View activity**

Opens the activity log for the selected script.

You can select a period, such as last hour, last week, last month etc., or you can specify a time range (start day and end day).

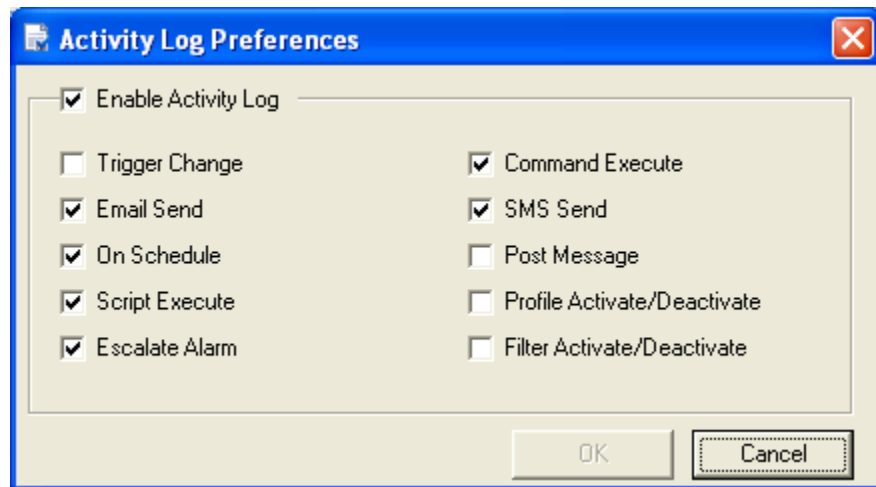
The activity log shows information like: When the script was run, the time used, the status etc.



Time	Module	Type	Identifier	Activity	Status	Time Us...
2008-01-29 17:37:5	AO	command	check-netstat	netstat -n -p TCP	0	332
2008-01-29 17:36:1	AO	command	check-netstat	netstat -n -p TCP	0	426
2008-01-29 13:29:3	AO	command	check-netstat	netstat -n -p TCP	0	285
2008-01-29 10:57:5	AO	command	check-netstat	netstat -n -p TCP	0	251
2008-01-28 13:36:3	AO	command	check-netstat	netstat -n -p TCP	0	289



Clicking the **Preferences** button opens the *Activity Log Preferences* dialog, enabling you to select additional items to be logged in the Activity log. Select the *Enable Activity Log* option (by default not selected) to enable the options on the dialog. Otherwise the options are deactivated (grayed out).

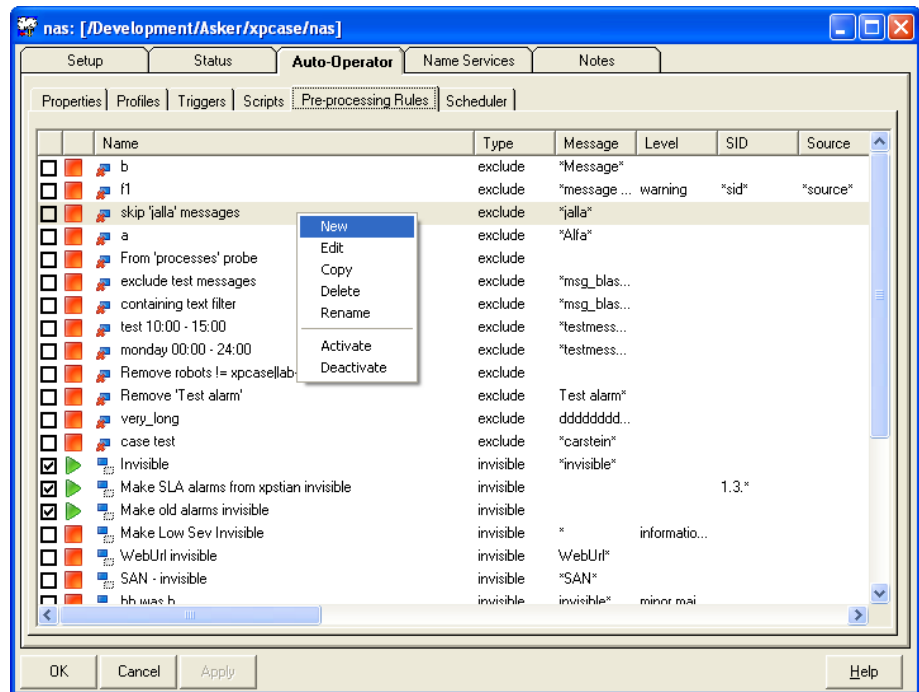


Note that you can move a script file by dragging the file onto a folder, or copy from a remote NAS by dragging the script from one UI to the other.

Pre-processing Rules

A pre-processing rule consists of a filter and a set of rules determining how the NAS will handle alarm messages matching the filter (exclude, set invisible, custom etc).

This is useful if you want to filter out specific alarm messages to be treated a specific way, or want specific alarm messages to be excluded and not managed by the NAS.



Icons used in the pre-processing rules list:



Means that the pre-processing rule is not activated. To activate it, right-click the rule and select *Activate*, or simply click the check box.



Means that the pre-processing rule is activated. To de-activate it, right-click the rule and select *Deactivate*, or simply de-select the check box.



This indicator means that alarm messages matching the filter set for this pre-processing profile will be excluded and not managed by the nas.



This indicator means that alarm messages matching the filter set for this pre-processing profile will be set to invisible. They will still be listed under the Status tab, provided that the option *Show Invisible Alarms* is selected. However, they will not be visible in the Alarm sub console in Infrastructure Manager and Enterprise Console, unless the ACL the Nimsoft user is associated with allows the user to see the invisible alarms there.

Right-clicking in the list gives you a set of options: Create a new rule, edit, copy, delete or rename a rule. In addition you can activate/deactivate a rule.

The properties dialog for a rule is as shown below:

The screenshot shows a 'Rule Properties' dialog box. At the top, there are two dropdown menus: 'Filter type' (set to 'exclude') and 'Category'. Below these are two tabs: 'Matching Criteria' (selected) and 'Operating Period'. The 'Matching Criteria' tab contains a 'Severity Level' section with a row of eight colored squares (green, light green, cyan, blue, dark blue, yellow, orange, red). Below this are several input fields arranged in three columns: 'Hostname', 'Source', 'NMS Domain', 'Subsystem ID', 'Suppression Key', 'NMS Hub Name', 'User Tag 1', 'User Tag 2', 'NMS Robot Name', 'Time of origin older than' (with a dropdown arrow), 'NMS Probe Name', 'Message string', and 'NMS Origin'. At the bottom of the dialog is a 'Custom Script' section with a text area and a dropdown arrow. A hint at the bottom reads: 'Hint: F1 for dropdown selection, F2 for pattern/regexp validation.'

Note: Filtering criteria fields can be inverted (set to *NOT Expression*) by clicking the small button above each of the fields. The buttons gets activated as soon as something is written in the fields.

The fields are:

Filter type

This option describes how the NAS will handle alarm messages matching the rules set in this dialog.

Valid options are *Exclude*, *Invisible* and *Custom*.

Exclude

Exclude alarm messages will not be managed by the NAS and are deleted.

Invisible

Alarm messages set *invisible* will managed by the NAS and will be listed under the Status tab, provided that the option *Show Invisible Alarms* (on the Status tab) is selected. However, they will not be visible in the Alarm sub console in the Infrastructure Manager and Enterprise Console, unless the ACL the Nimsoft user is associated with allows the user to see the invisible alarms there.

Custom

Enables the Custom script field at the bottom of the dialog (otherwise it is not available).

Transaction

Alarms matching the criteria for this filter are not added to the transaction logs. This is useful if the transaction logs are filled up with messages such as heartbeat.

Category

Select the category to be used. Category is a way to group triggers to ease the administration in case of many rules. You may create new categories by placing the cursor in the field and typing a new category.

Severity Level

Select the severity level(s) for the alarms you want to be treated by this filter. Only alarms with the specified severity level are treated.

Hostname

Specify the hostname (string matching) sending alarms to be treated by this filter.

Source

Specify the source (string matching) for alarms to be treated by this filter.

NMS Domain

Specify the name of the Domain (string matching) the host sending alarms to be treated by this filter belongs to.

Subsystem ID

Specify the subsystem ID (string matching) for alarms to be treated by this filter.

Suppression key

A unique ID with which the probe tags the alarms. This is done to avoid that the Alarm Console receives too many instances of the same alarm. A counter indicates the number of times the alarm is sent from the probe.

NMS Hub name

Specify the name of the Hub (string matching) the host sending alarms to be treated by this filter belongs to.

User Tag 1

User-defined tag in the Controller probe. To be used as a grouping / locating mechanism.

User Tag 2

User-defined tag in the Controller probe. To be used as a grouping / locating mechanism.

NMS Robot name

Specify the robot (string matching) sending alarms to be treated by this filter.

Time of origin older than

Only alarms with origin older than the number of days specified here will be treated select one of the pre-defined values or type another value on the format **n days**.

Time of origin tells when the alarm was sent from the Probe.

NMS Probe name

Specify the probe (string matching) sending alarms to be treated by this filter.

Message string

Specify a text string (string matching) found in alarms to be treated by this filter.

NMS Origin

Specify the origin (string matching) sending alarms to be treated by this trigger.

QoS data from probes are tagged with a name to identify the origin of the data.

If specifying an origin name in the controller probe, this name will be used to identify the origin of the data.

If not, the Hub name will be used.

Note however, that the *Origin* field under *Advanced Settings* in the Hub GUI lets you specify an origin name of your own choice to be used, rather than the Hub name.

Custom Script

This field is only enabled if the Filter type is set to custom.

Select a LUA script to pre-process the alarm messages. The menu lists all scripts available. You may create/edit these scripts yourself, using the LUA programming language (see the Scripts tab). Note that only a subset of the Lua methods are available to the pre-processing script. The following classes and methods are not available: exit, sleep, nimbus, pds, trigger, action, database, alarm and note. The trigger.state method through the state method is however available.

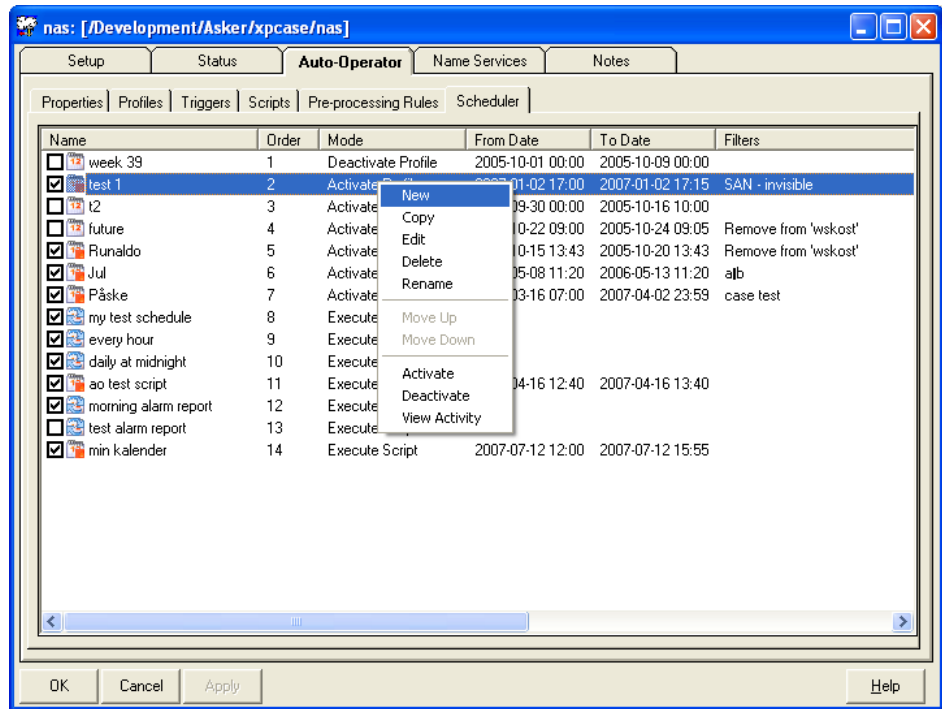
Operating Period tab

Specifies the periods when the filter should be valid.

See the section Setting an Operating Period for information about these settings.

Scheduler

This tab enables you to define *scheduler* profiles. These are used to administer the running properties for the Auto-operator profiles and the Pre-processing Rules. The scheduler profiles can activate or deactivate Auto-operator profiles and/or pre-processing rules for defined time periods.



The scheduler can also execute scripts, independent of Auto-operator profiles.

The Scheduler works in co-operation with the *operating periods* specified for the Auto-operator profiles and the Pre-processing Rules. When a scheduler is selected to *activate* a profile, it is possible to select the scheduler to override the operating period defined for the profile.

Right-clicking in the list gives you the following options:

- **New**

Opens the Schedule properties dialog, allowing you to create a new schedule.

- **Copy**

Copies the selected schedule with a new name and displays it in the list. The new profile will be named "New Calendar", but have the same properties as the schedule you copied. Use the Rename option to give the schedule a descriptive name.

- **Edit**

Opens the properties dialog for the selected schedule, allowing you to edit the schedule properties.

- **Delete**

Deletes the selected schedule.

You are asked to confirm the deletion.

- **Rename**

Allows you to rename the selected schedule. The current schedule name is highlighted; just type the new name.

- **Activate**

Activates the selected schedule.

- **Deactivate**

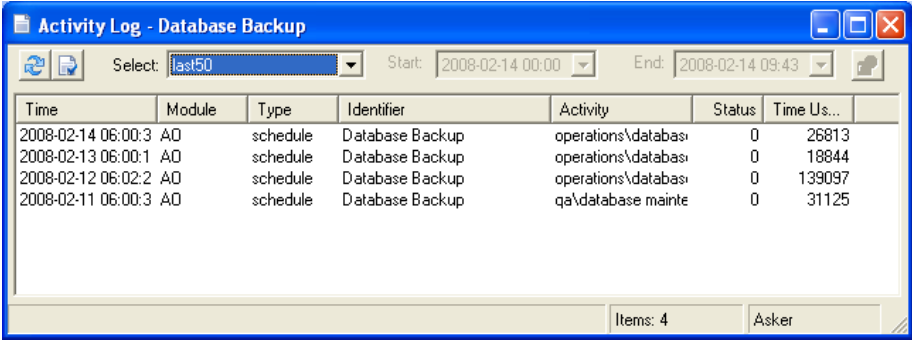
Deactivates the selected schedule.

- **View activity**

Opens the activity log for the selected schedule.


You can select a period, such as last hour, last week, last month etc., or you can specify a time range (start day and end day).

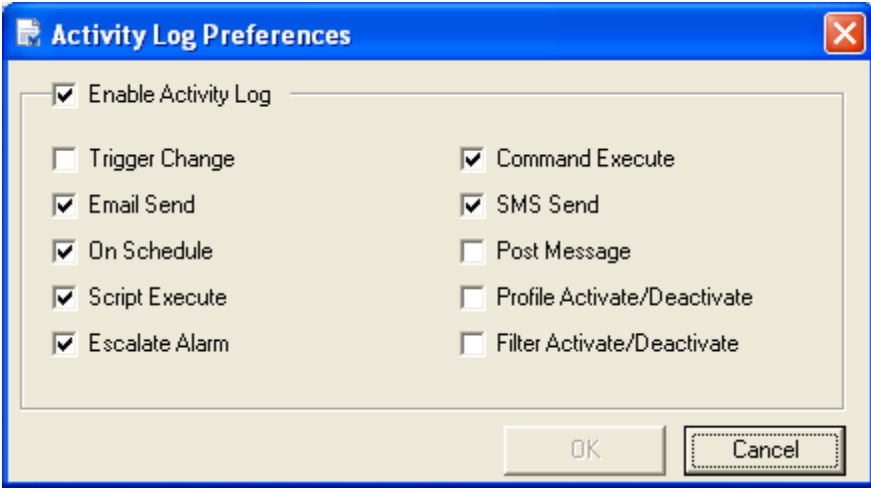
The activity log shows information like: When the schedule was active, the time used, the activity (activating, deactivating or running scripts).



The screenshot shows the 'Activity Log - Database Backup' window. It has a title bar with standard Windows window controls. Below the title bar is a toolbar with icons for refresh, print, and a search icon. To the right of the toolbar is a 'Select:' dropdown menu showing 'last50'. Further right are 'Start:' and 'End:' date/time pickers. The main area is a table with columns: Time, Module, Type, Identifier, Activity, Status, and Time Us... (likely Time Used). The table contains four rows of data, all for 'Database Backup' events. At the bottom right, there are two buttons: 'Items: 4' and 'Asker'.

Time	Module	Type	Identifier	Activity	Status	Time Us...
2008-02-14 06:00:3	AD	schedule	Database Backup	operations\databas	0	26813
2008-02-13 06:00:1	AD	schedule	Database Backup	operations\databas	0	18844
2008-02-12 06:02:2	AD	schedule	Database Backup	operations\databas	0	139097
2008-02-11 06:00:3	AD	schedule	Database Backup	qa\database mainte	0	31125

 Clicking the **Preferences** button opens the *Activity Log Preferences* dialog, enabling you to select additional items to be logged in the Activity log. Select the *Enable Activity Log* option (by default not selected) to enable the options on the dialog. Otherwise the options are deactivated (grayed out).



The screenshot shows the 'Activity Log Preferences' dialog box. It has a title bar with a close button. The main area contains a list of checkboxes. The first checkbox, 'Enable Activity Log', is checked. Below it are two columns of checkboxes. The left column has: 'Trigger Change' (unchecked), 'Email Send' (checked), 'On Schedule' (checked), 'Script Execute' (checked), and 'Escalate Alarm' (checked). The right column has: 'Command Execute' (checked), 'SMS Send' (checked), 'Post Message' (unchecked), 'Profile Activate/Deactivate' (unchecked), and 'Filter Activate/Deactivate' (unchecked). At the bottom right are 'OK' and 'Cancel' buttons.

☒ Enable Activity Log

☐ Trigger Change ☒ Command Execute

☒ Email Send ☒ SMS Send

☒ On Schedule ☐ Post Message

☒ Script Execute ☐ Profile Activate/Deactivate

☒ Escalate Alarm ☐ Filter Activate/Deactivate

OK Cancel

Creating or Editing a Schedule

The Schedule dialog allows you to enter scheduling information for profiles. This screen contains three sections. The top section allows you to select the mode and operation. Depending on the options selected, the lower two sections (Mode: and Operation:) change to reflect the Mode or Operation selected at the top of the screen. This example displays the Mode: By Time and Operation: Deactivate Profile options.

Schedule: [NEW]

Mode By Time **Operation** Deactivate Profile

Mode: By time

Start time: 15:04

Start date: 2007-09-05

End time: 16:04

☐ End date: 2007-09-05

September 2007

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
35	27	28	29	30	31	1	2
36	3	4	5	6	7	8	9
37	10	11	12	13	14	15	16
38	17	18	19	20	21	22	23
39	24	25	26	27	28	29	30
40	1	2	3	4	5	6	7

Today: 05/09/2007

Operation: Activate/Deactivate Profile

Filters Auto-Operator Profiles

Pre-processing filters

Available

- All fields
- bb was b
- Invisible
- Make Low Sev Invisible
- Make old incoming alarms invisible

Selected

ALL ☐

Duration 5m ☐ Ignore Operating-Period

OK Cancel

The fields are:

Mode

Three different modes can be used to specify the time settings for the scheduler. The scheduler will activate or deactivate Auto-Operator profiles and Pre-processing rules treated by this schedule or run the script specified, depending on operation selected.

By Time

Use the drop-down menus to specify a *start date* and *start time*, and also *end time* and an *end date*.

You can also use the calendar to set the start date and end date.

Click on a date to set the start date, and then use <shift> + click to mark the end date.

By Recurring Event

Select the date range for the recurring event and the pattern of the recurrence.

Use the mode By Recurrent Event, you first specify the time period the selected operation should be performed, with a start day and time, and optionally an end day and time.

Mode: By recurring event

Range

☒ Start now

☐ Start at 2012-05-29 11:29

☒ No end date

☐ End after 1 occurrences

☐ End by 2012-05-30 11:29

Pattern

☐ Minutely

☐ Hourly

☒ Daily

☐ Weekly

☐ Monthly

☐ Yearly

☐ Every 1 day(s)

☒ Every weekday

Hour:

Minutes:

E.g 10,12,14

Forecast

You must also specify a pattern, defining how often the selected operation should be performed within the time period specified.

Click the **Forecast** button to list the future occurrences of the selected operation.

Note: If the operation selected is Activate or Deactivate and the mode is Recurrent you can select how long the Auto-Operator profiles and Pre-processing rules treated by this schedule should be activated / deactivated by entering a value in the **Duration** field.

By Calendar

Specify a point in time when the Auto-Operator profiles and Pre-processing rules treated by this schedule should be activated, deactivated, or the specified script executed.

Operation

Three different operations are available for the schedule:

Activate Profile

Activates the Auto-Operator profiles and Pre-processing rules treated by this schedule at the time specifications set. Use the time settings to define how long the Auto-Operator profiles and Pre-processing rules treated by this schedule should be active.

You can also chose to ignore the operating period for activating a profile. Select the **Ignore Operating Period** option in the lower portion of the screen. The Operating Period specified for the Auto-Operator profiles and Pre-processing rules treated by this schedule will be ignored.

Deactivate Profile

Deactivates the Auto-Operator profiles and Pre-processing rules treated by this schedule at the time specifications set. Use the time settings to define how long the Auto-Operator profiles and Pre-processing rules treated by this schedule should be deactivated.

Execute Script

Executes the script specified in the time period indicated.

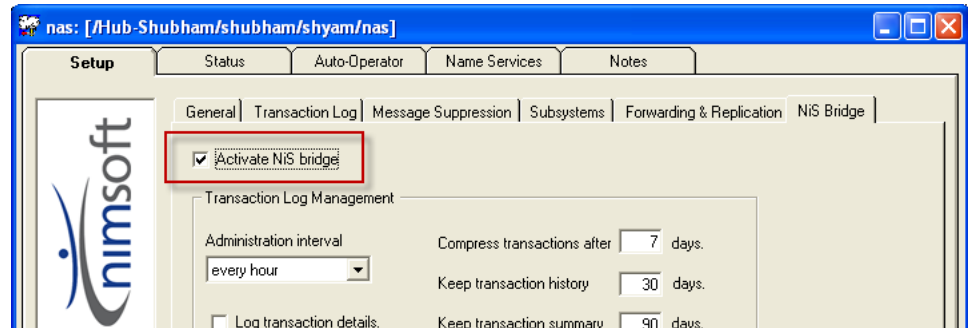
If the time specification mode *By Time* is chosen, you may select how often to execute the script. These options are:

- Run once
- Run on interval every
- Run on Auto Operator interval.

This option does not apply and is deactivated for the *By Recurring Event* and *By Calendar* modes.

Computer State Monitor

The *Computer State Monitor* is enabled only when the NiS Bridge is activated from the *Setup > NiS Bridge* tab.



The NAS attempts to locate the data_engine on its hub and requests the database connectivity information. If this is obtained and is usable, then the user may activate the NIS bridge by activating the checkbox. This is a configuration file parameter, and can be set by the server installation script.

The Computer State Monitor provides the ability to monitor the computer system (CS) table (CM_COMPUTER_SYSTEM) for changes in the state field. This field indicates whether the computer system (or rather the devices referenced by the CS) is in a maintenance/managed mode or not.

The alarms are tagged with the sender's device_id and these are then further mapped to the appropriate CS (over the CM_DEVICE table). A state may be examined and used by the current computer system monitor ruleset. This enables to act against the device regardless if it is monitored locally or from one or more remote locations from different probes.

Note: This functionality requires NMS Server 4.10 or higher and robots compatible with NMS Server 4.10 or higher.

The screenshot shows the 'Auto-Operator' configuration window with the 'Computer State Monitor' tab selected. The window has a top navigation bar with tabs: Setup, Status, Auto-Operator, Name Services, and Notes. Below this is a sub-navigation bar with tabs: Properties, Profiles, Triggers, Scripts, Pre-processing Rules, Scheduler, and Computer State Monitor. The main content area is divided into several sections:

- Activate:** A checkbox labeled 'Activate' is present.
- Polling interval for updates:** A dropdown menu showing '2 min'.
- Action On 'Maintenance' State:** A group box containing four radio buttons: 'None' (selected), 'Filter Message', 'Make Invisible', and 'Run Pre-Processing Script'. Below these is a text input field.
- Action On 'Ignored' State:** A group box containing three radio buttons: 'None' (selected), 'Filter Message', and 'Make Invisible'.
- Action On 'Managed' State:** A group box containing two radio buttons: 'None' (selected) and 'Make Visible'.
- Help Text:** A text area on the right side stating: 'This service will monitor the NIS Computer System table for changes in the state field. The state is changed through the NIS Manager and can be Managed, Maintenance, Ignore or None. This feature is supported by NMS 4.1 compatible components or newer.'

The fields are:

Activate

Activates the computer state monitor.

Polling interval for updates

Choose a time-interval in minutes that nas will check for the changes in the state of the monitored systems.

Action on 'Maintenance' State

Action to be performed for the systems in the "Maintenance" state. The following options are available:

- *None*: perform no action
- *Filter Message*: filter the messages from these systems
- *Make Invisible*: make the systems 'invisible'
- *Run Pre-processing Script*: choose one of available scripts from the drop-down menu.

Action on 'Ignored' State

Action to be performed for the systems in the "Ignored" state. The following options are available:

- *None*: perform no action
- *Filter Message*: filter the messages from these systems
- *Make Invisible*: make the systems 'invisible'

Action on 'Managed' State

Action to be performed for the systems in the "Managed" state. The following options are available:

- *None*: perform no action
- *Make Visible*: make the systems 'visible', in case they were 'invisible'

Pattern Matching in Auto-Operator

In NAS version 3.x, changes were implemented to our pattern-matching to regular-expression conversion library to avoid improper matching.

This impacts the NAS where a 'loose' pattern has been used.

Consider the string '**robot1**' as the robot field in the matching criteria in a preprocessing filter, Auto-Operator profile or a trigger definition.

***robot1** resulted in match on myrobot1 and myrobot100. This was recognized as a bug after the release of NAS 2.75 and was in all prior releases.

All fields will be treated as possible patterns **unless** starting and ending with a '/'.

If no special characters like `*.?(())` appear as the first character in the target-string, the expression will be prefixed with a `^` (signifying the start of string). And likewise for the last character, if the target-string ends with a `*`, then a `$` (dollar: signifying rest of string) is added.

All fields starting and ending with a '/' (slash) will be passed on and treated as a true regular expression.

Also note that '\ ' means escape in the pattern-matching/regex world. If for example using the text string *Average (4 samples) disk free on C:\ is now 93%, which is below the error threshold (95%)* as matching criteria, you should substitute the '\ ' with e.g. a '*' in the text string.

Hence, 'robot1' is internally converted and compiled to the regular expression '^robot1\$'. This will only match 'robot1'.

The user must actively append an '*' (asterix), to make the expression match a string containing robot1 as a substring.

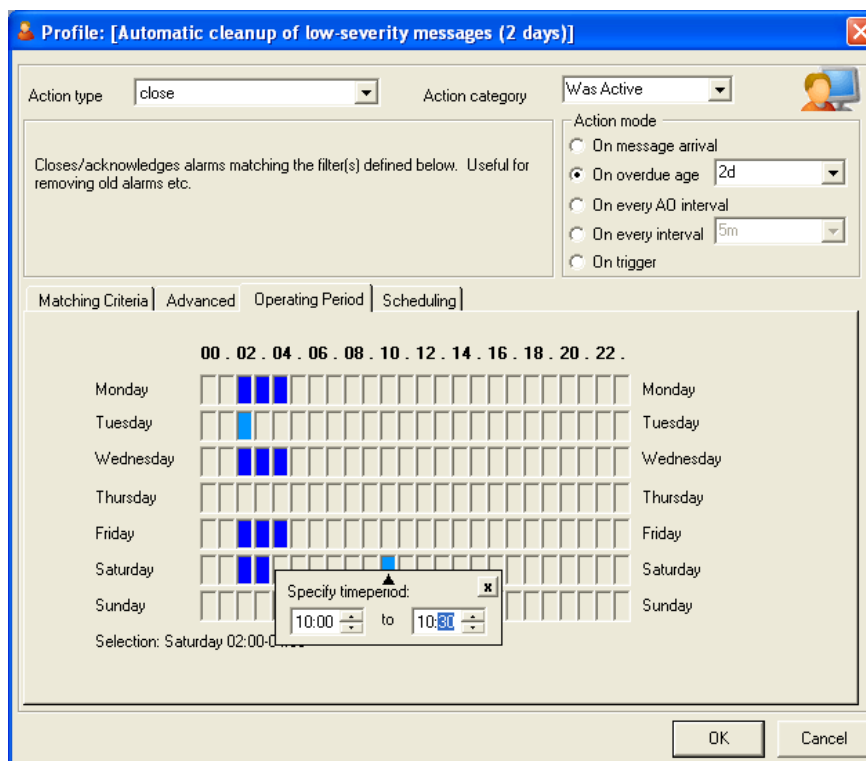
The field may also be comma-separated (**NOTE: Except for the message field!**) with a mixture of regex and patterns e.g. robot1, robot2*, /.robot./.

So, the above match criteria will match:

robot1	the first criteria
robot2	the second criteria
robot222	the second criteria
myrobottest	the third criteria

To select multiple boxes, select the first time box and then hold the shift key and select the last time box.

To select more than an hour, double-click inside a box to open a small dialog that allows you to specify the start and/or end-time within the selected hour in the format hh:mm.

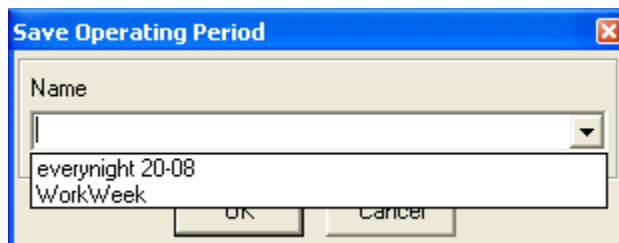


Click the **OK** button to activate the operating period.

Right-click in the operating period dialog to select one of the following options:

- Clear the selected day.
- Clear the whole operating period definition.
- Select one of the predefined operating periods.
- Save As to save your operating period definition.

Specify a name for the operating period, or you can overwrite one of the existing operating periods by selecting them from the list.



The Name Services Tab

If you select the *Enable Name Services* option, the IP-addresses of the sources sending alarms will be resolved to host names according to the rules defined on this screen. The IP-addresses (with corresponding host names) of all sources sending alarms will be listed under the *Address Table* sub-tab. By right-clicking an entry in the list, you may overrule the hostname with a name of your own choice. You may also add your own entries to the list.

The tab contains two sub-tabs:

- Properties
- Address Table

Name Services Properties

Select the *Enable Name Services* option to map the IP-addresses of the sources sending alarms to host names. Otherwise, the IP-address will be entered as the host name in the alarm messages.

You can also select the name resolution rules:

- **Use RobotName as Hostname (if available)**

Applies to alarms generated on/by a Robot (source IP is the same as the robot's IP address).

- **Use Default Name-Resolution**

Applies the default name, however you can modify the default name with the following options:

- **Lowercase Hostname After Resolution**

The host name will be converted to lower case.

- **Ignore Unsuccessful Resolution Attempts**

Lookups that failed will not be recorded, but ignored.

- **Revalidate Name every**

Specifies the interval at which the NAS attempts to validate host names.

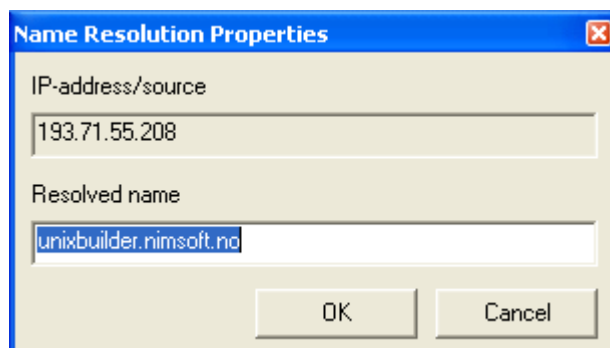
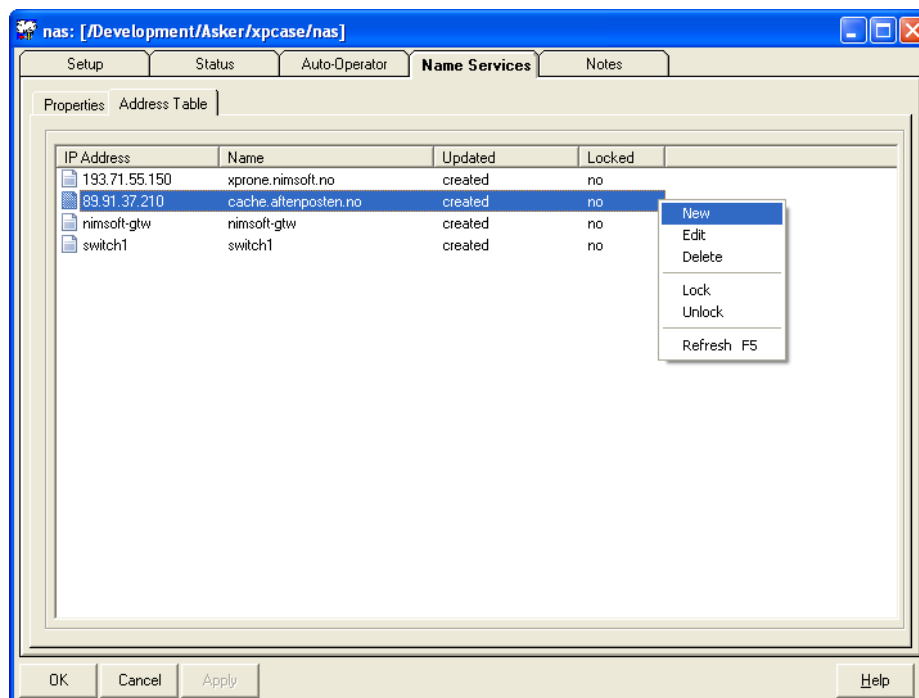
The screenshot shows a software window titled 'Name Services' with tabs for 'Setup', 'Status', 'Auto-Operator', 'Name Services' (selected), and 'Notes'. Inside the 'Name Services' tab, there are two sub-tabs: 'Properties' and 'Address Table'. The 'Properties' sub-tab is active and contains the following settings:

- ☒ Enable Name Services
- ☒ Use NMS RobotName as Hostname (if available).
- ☒ Use Default Name-Resolution. DNS, WINS etc.
- ☒ Lowercase Hostname After Name-Resolution.
- ☐ Ignore Unsuccessfull Resolution Attempts.
- Revalidate Name Every: E.g. 1h30m

Below these settings, a note states: 'The NimBUS Alarm Server will attempt to resolve an IP Address in the 'source' field of an alarm using the options above and in the order from top to bottom.'

Name Services Address Table

Right-clicking in the list allows you to add, edit existing, or delete entries from the list. Selecting *Refresh* (or pressing the **F5** button on your keyboard) refreshes the list to display the most current contents.

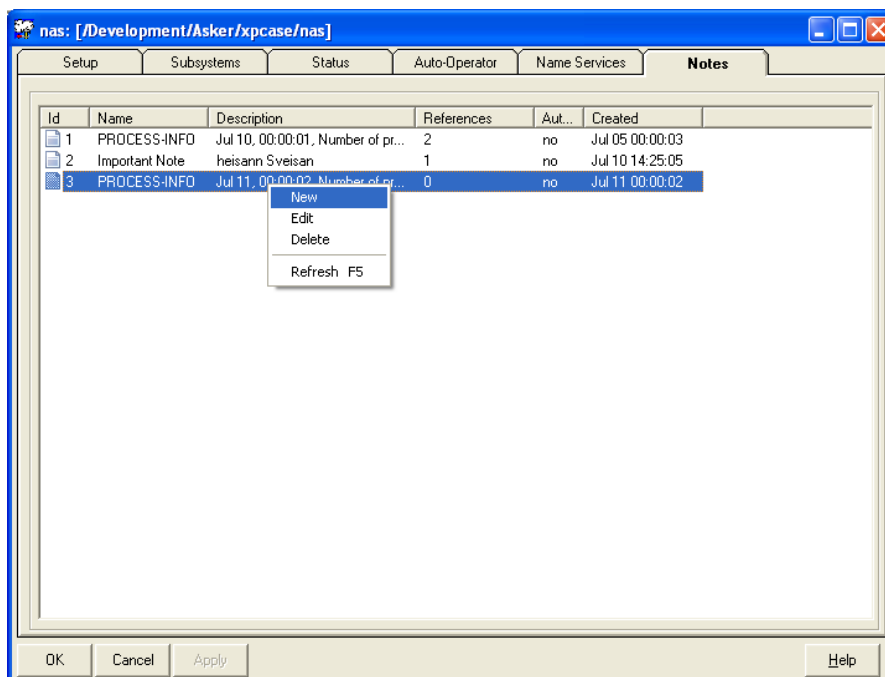


The *Lock* option lets you protect the entry from being modified.

The *Unlock* option removes the lock protection from a locked entry.

The Notes Tab

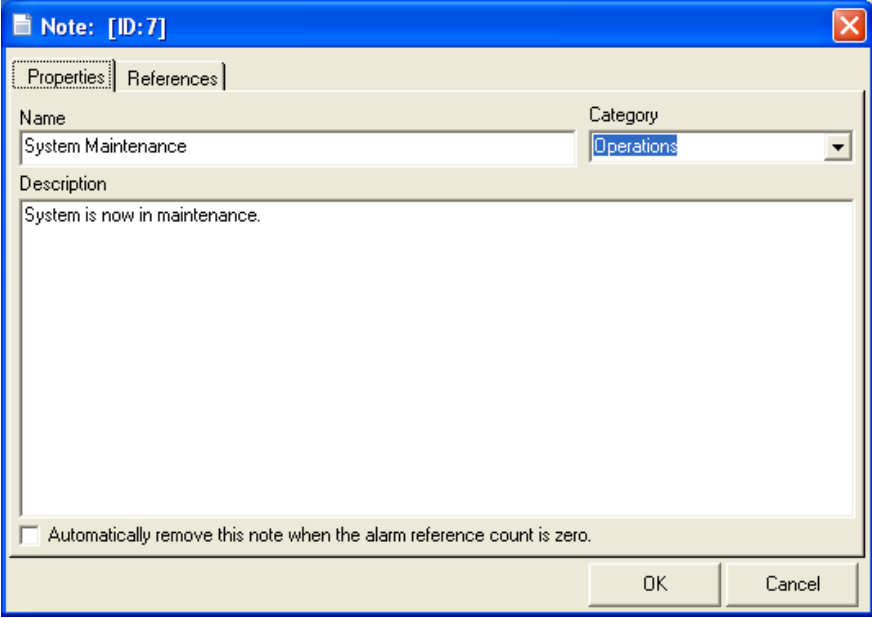
Under this tab you can add text notes. These notes can be attached to Alarms. See The Alarm List section for more information on how to attach a note to an alarm.



Right-click in the list to add, edit existing, or delete notes. Select Refresh (or pressing the F5 button on your keyboard) to display the most current contents.

The Note dialog contains two sub-tabs:

- Properties
- References



The screenshot shows a Windows-style dialog box titled "Note: [ID: 7]". It has two tabs: "Properties" (selected) and "References". In the "Properties" tab, there is a "Name" text box containing "System Maintenance", a "Category" dropdown menu with "Operations" selected, and a "Description" text area containing "System is now in maintenance.". At the bottom of the dialog, there is a checkbox labeled "Automatically remove this note when the alarm reference count is zero." which is currently unchecked. To the right of the checkbox are "OK" and "Cancel" buttons.

The fields are:

Name

Descriptive name for the note.

Description

Enter the text of the note. These notes can be attached to Alarms.

Automatic removal

The note will automatically be deleted when the alarm to which the note is attached is deleted.

References tab

Displays all alarms that this note is attached to. If no alarms are linked to the note, the tab will be grayed out.

You can create notes from other sections of this GUI:

- Right-click an alarm under the Status tab, selecting *Create > Note* also lets you create a new note. This note will be attached to the alarm, and will appear under the Notes tab as soon as you right-click in the list, selecting Refresh (or pressing the F5 button on your keyboard).
- Auto Operator profiles can be configured to add notes to alarm messages.

Chapter 5: The Script Editor

Use this editor to create and edit scripts, using the **Lua** scripting language. These scripts can be selected to be used by the Auto Operator when processing alarm messages matching the criteria defined for the Auto Operator profile. Scripts can also be run by the Scheduler.

What is Lua?

Lua is a powerful, light-weight programming language designed for extending applications. Lua is also frequently used as a general-purpose, stand-alone language. It is dynamically typed, interpreted from opcodes, great facility to handle strings and other kinds of data with dynamic size, and has automatic memory management with garbage collection, making it ideal for configuration, scripting, and rapid prototyping.

Lua is easily extended not only with software written in Lua itself, but also with software written in other languages, such as C and C++. Lua is also a glue language. Lua supports a component-based approach to software development, where we create an application by gluing together existing high-level components, written in a compiled, statically typed language, such as C or C++; Lua is the glue that we use to compose and connect those components. However, unlike other glue technologies, Lua is a full-fledged language as well. Therefore, we can use Lua not only to glue components, but also to adapt and reshape them, or even to create whole new components.

Lua features

Lua is not the only scripting language around. There are other languages that you can use for more or less the same purposes, such as Perl, Tcl, Ruby, Forth, and Python. The following features set Lua apart from these languages; although other languages share some of these features with Lua, no other language offers a similar profile:

Extensibility: Lua's extensibility is so remarkable that many people regard Lua not as a language, but as a kit for building domain-specific languages. Lua has been designed from scratch to be extended, both through Lua code and through external C code. As a proof of concept, it implements most of its own basic functionality through external libraries. It is really easy to interface Lua with C/C++ and other languages, such as Fortran, Java, Smalltalk, Ada, and even with other scripting languages.

Simplicity: Lua is a simple and small language. It has few (but powerful) concepts. This simplicity makes Lua easy to learn and contributes for a small implementation. Its complete distribution (source code, manual, plus binaries for some platforms) fits comfortably in a floppy disk.

Efficiency: Lua has a quite efficient implementation. Independent benchmarks show Lua as one of the fastest languages in the realm of scripting (interpreted) languages.

Portability: When we talk about portability, we are not talking about running Lua both on Windows and on Unix platforms. We are talking about running Lua on all platforms we have ever heard about: NextStep, OS/2, PlayStation II (Sony), Mac OS-9 and OS X, BeOS, MS-DOS, IBM mainframes, EPOC, PalmOS, MCF5206eLITE Evaluation Board, RISC OS, plus of course all flavors of Unix and Windows. The source code for each of these platforms is virtually the same. Lua does not use conditional compilation to adapt its code to different machines; instead, it sticks to the standard ANSI (ISO) C. That way, usually you do not need to adapt it to a new environment: If you have an ANSI C compiler, you just have to compile Lua, out of the box.

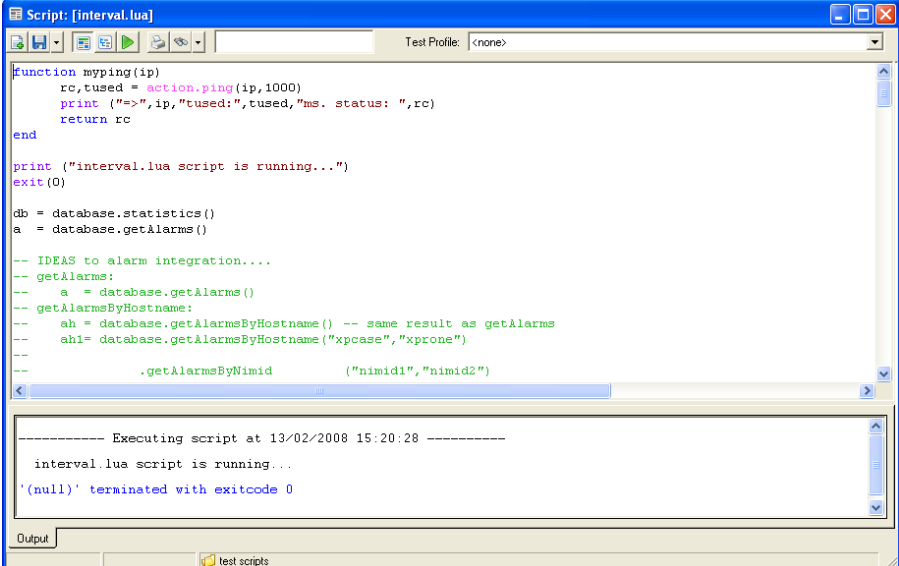
References:

See the sites for more information on the Lua scripting language:

http://www.scratchprojects.com/2007/08/introduction_to_lua_programming_p02.php
http://www.scratchprojects.com/2007/08/introduction_to_lua_programming_p02.php
http://www.scratchprojects.com/2007/08/introduction_to_lua_programming_p02.php

<http://lua-users.org/wiki/LuaTutorial> <http://lua-users.org/wiki/luatutorial>

<http://lua-users.org/wiki/SampleCode> <http://lua-users.org/wiki/samplecode>



```
Script: [interval.lua]
Test Profile: <none>

function mypping(ip)
    rc,tused = action.ping(ip,1000)
    print ("=>",ip,"tused:",tused,"ms. status: ",rc)
    return rc
end

print ("interval.lua script is running...")
exit(0)

db = database.statistics()
a = database.getAlarms()

-- IDEAS to alarm integration...
-- getAlarms:
--   a = database.getAlarms()
-- getAlarmsByHostname:
--   ah = database.getAlarmsByHostname() -- same result as getAlarms
--   ah1= database.getAlarmsByHostname("xpcase","xprone")
--
--   .getAlarmsByNimid      ("nimid1","nimid2")

----- Executing script at 13/02/2008 15:20:28 -----
interval.lua script is running...
'(null)' terminated with exitcode 0
```

The script editor window consists of the following parts:

- A row of tool buttons. These tool buttons offer the functionality needed to create and save a script.
- The workspace (main window) where the code is written.
- A syntax helper in the right part of the window. You can select this syntax helper to be shown or hidden, clicking the Show/hide syntax helper button.



The syntax helper appears as a tree-structure. Expanding the tree-structure, you will find elements that you can use in the script.

- An output field located at the bottom of the window. This field will show the output generated when executing a script. Error messages reporting syntax error will appear as red text.

Notes:

- Placing the cursor in the script window, pressing F1 on your keyboard, will bring up a short explanation of the keyboards shortcuts available.
- Marking an elements in the script, pressing F1 on your keyboard, will bring up a description of the NAS extensions to LUA.

The tool buttons



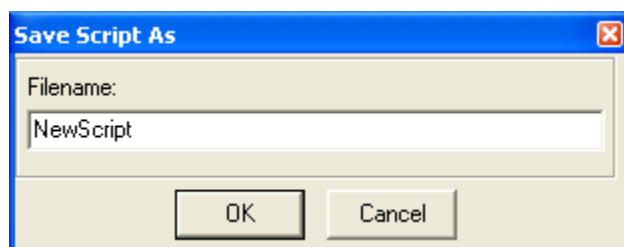
New

Click this button to create a new script.



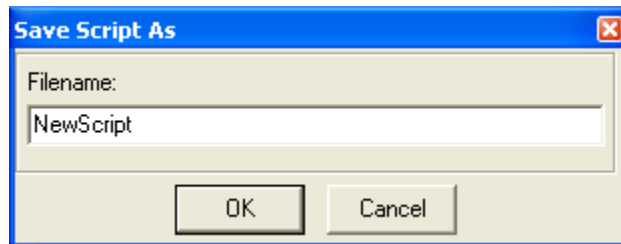
Save

Click this button to save a script. If the script is new and never has been saved before, you will be prompted for a script name.

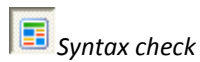




Click this button to save new scripts (that never has been saved before). You will be prompted for a script name.



Enter a name and click the **OK** button. The script can now be found, listed under the *Scripts* tab.



Clicking this button toggles the syntax check mode on and off. With the syntax check turned on, your code will continuously be checked as you write.

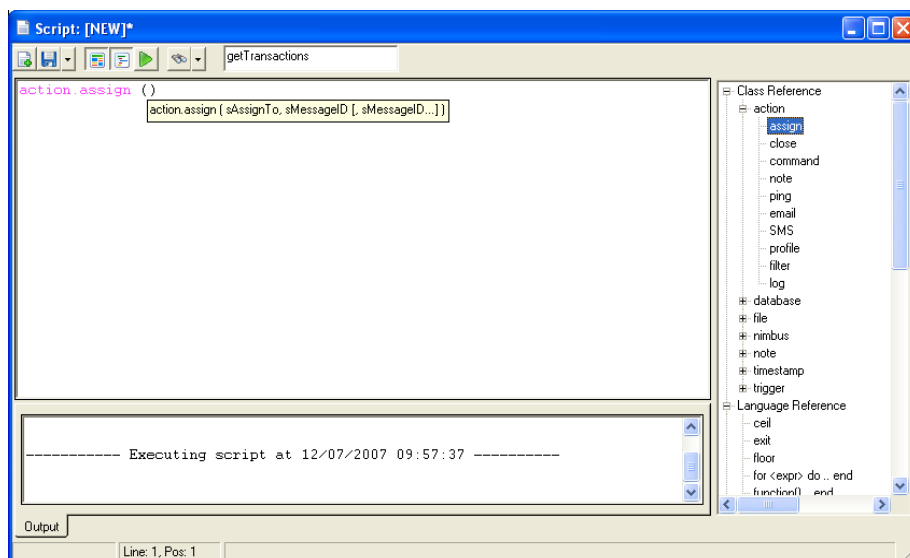
Note: When working with large scripts, it is advisable to turn of syntax check if the response is slow.



Show/hide syntax helper

Clicking this button toggles the syntax helper on and off. With the syntax helper turned on, the syntax helper will appear in the right part of the window, appearing as a tree-structure.

Expanding the tree-structure, you will find elements that you can use in the script. Double-clicking an element, the element will appear in the script where the cursor is placed, and a help text (in a yellow frame) will show you the correct syntax.



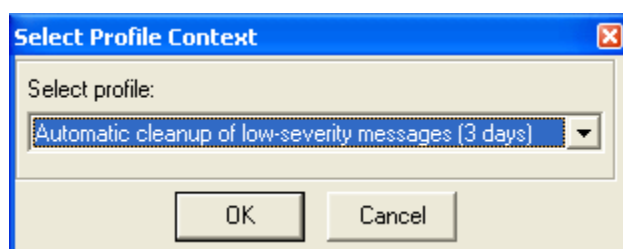
Validate and execute script

Clicking this button, the script in the window will be validated and executed.

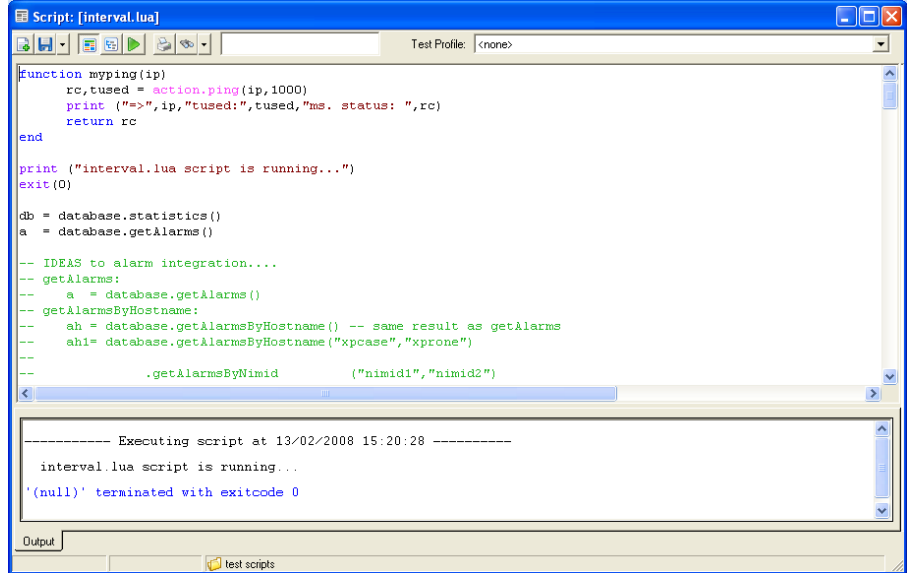
If you have modified the script, you must save it before executing it.

When clicking the **Validate and execute script** button, you will be prompted for a profile name.

Select in which profile context you want to execute the script, or select *None*.



The output will be shown at the bottom of the window. If syntax errors are found during the validation, an error message describing the error will appear as red text in the output field.



Print

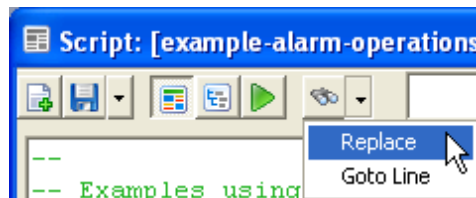
Click this button to make a print-out of your script



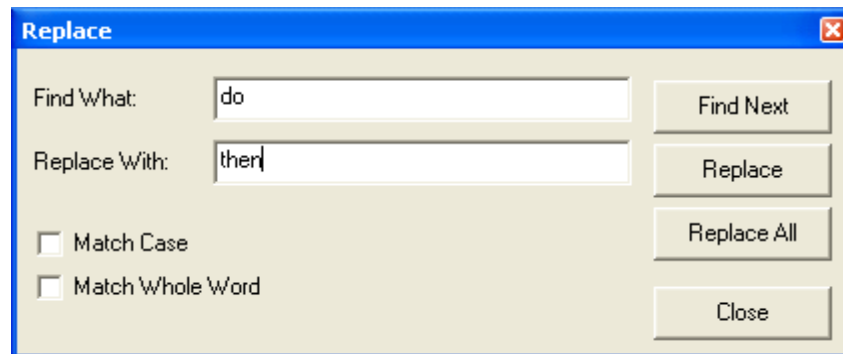
Find

Clicking this option will search through the script for the string entered in the text field. If found, the string will be highlighted.

Clicking the down arrow next to the *Find* button, you can select either to find and replace a specified text string or to go to a specified line in the script.



Replace



Enter the text string you want to replace and the text string with which you want to replace it.

Click the **Find Next** button to find the first instance of the text string. When found, the text string will be highlighted. Click *Replace* if you want to replace it. Otherwise click **Find Next** to continue searching through the script.

Click the **Replace All** button if you want to replace all instances of the text string found.

Go to

Entering a line number and clicking the **OK** button will take you to the specified line in the script, placing the cursor at the start of the line.

This section contains the following topics:

[Keyboard shortcuts](#) (see page 100)

Keyboard shortcuts

TAB – indent the selected text

F# - Find (next or first)

F5 - Execute script

F8 - Uppercase selected text

SHIFT + F8 - Lowercase selected text

CTRL + F - Find text

CTRL + G - Go to line

CTRL + P - Print script

CTRL + W – Save window preferences

CTRL + S – Save script

Chapter 6: The Alarm List

On the **Status** tab, the Alarm list will display showing the current alarms.

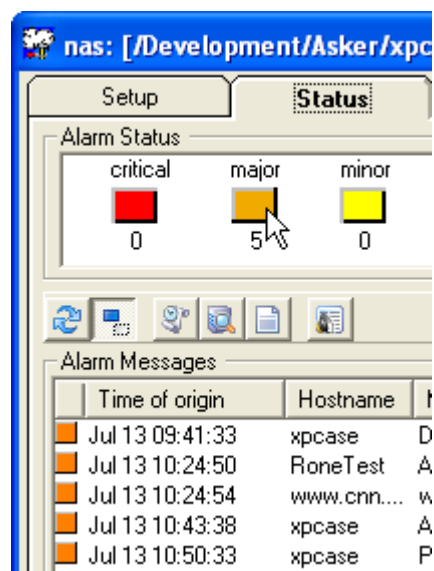
The screenshot shows a window titled 'nas: [/Development/Asker/xpcase/nas]' with tabs for Setup, Status, Auto-Operator, Name Services, and Notes. The 'Status' tab is active, showing an 'Alarm Status' section with color-coded buttons for critical (0), major (2), minor (1), warning (2), information (0), and clear (8). It also shows 'Total alarms: 13', 'Last alarm: Jul 16 14:31:14', and 'Last event: Jul 16 14:49:56'. Below this is an 'Alarm Messages' table with columns for Time of origin, Hostname, Message, Count, Subsystem, and Severity.

Time of origin	Hostname	Message	Count	Subsystem	Severity
Jul 14 01:43:02	cisco 1710	RTR 3: The 'http' probe configuration is now available.	1	Network	clear
Jul 14 20:31:02	cisco 1710	RTR 3 (http:nimsoft.no): The 'http' probe is performing as exp...	1	Network	clear
Jul 15 03:27:02	cisco 1710	RTR 2: The 'http' probe configuration is now available.	1	Network	clear
Jul 15 21:31:11	xpcase	Timed probe group_server not finished at next start time, rest...	3	Controller	warning
Jul 16 04:33:07	xpcase	Userenv(1030): Windows cannot query for the list of Group P...	7	Application	minor
Jul 16 06:31:02	cisco 1710	RTR 1: The 'jitter' probe configuration is now available.	1	Network	clear
Jul 16 10:41:55	xpcase	Queue 'TIL-xproar' failed to connect to hub /Development/xp...	18	Message...	warning
Jul 16 12:14:54	www.cnn...	www.cnn.com: Connection to www.cnn.com [ping] failed	26	Network	major
Jul 16 12:24:51	RoneTest	Average (4 samples) disk free on C:\ is now 2%, which is belo...	8	Disk	major
Jul 16 13:09:51	RoneTest	Processor queue length clear: Average(5)=2.00, Last=2	1	CPU	clear
Jul 16 13:39:55	RoneTest	URL response for 'Nimsoft' is checked and ok	1	Application	clear
Jul 16 13:39:56	RoneTest	URL response for 'Slashdot' is checked and ok	1	Application	clear
Jul 16 14:31:13	nimsoft-gtw	The inbound traffic on interface 'Nimsoft-Internet' is checked,...	1	Network	clear

Version version 3.00.3, Jul 12 2007. Started Jul 12 16:04:08

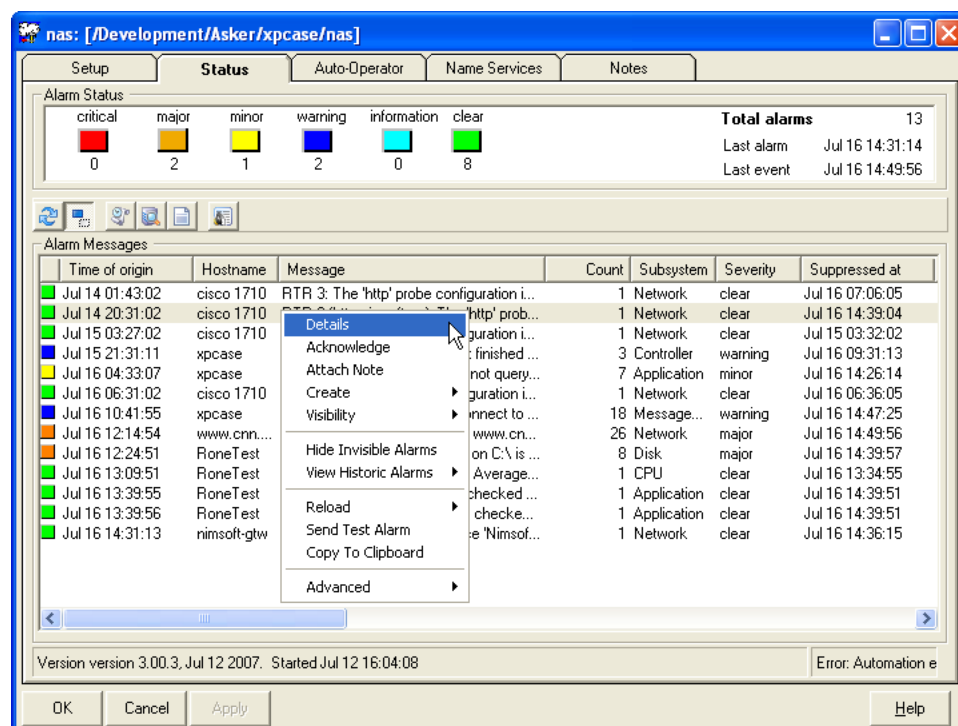
Buttons: OK, Cancel, Apply, Help

The *Alarm status* window (in the upper part of the window) shows the total number of alarms, and the number of alarms for each severity level. Double-click one of the icons in the window, such as **major**, and only alarms with severity level major will be listed in the Alarm messages list (main window).



Managing the alarms

The main window displays the current alarms in the NAS database.



A command menu is available when you right-click in the alarm-messages list. You can perform various administrative tasks from this menu, such as sending test-alarms, viewing alarm details, acknowledging alarms, viewing historic events etc. The *count* column in the alarm-messages list indicates the number of times the alarm has been received.

Right-clicking in the list provides you the following options:

Details

Displays the complete information about the alarm, including the entire life cycle of the alarm (if transaction-logging is enabled) and any notes attached to the alarm.

Alarm Details: FV15138525-00016

Details | Transaction log | Notes

Address details

NMS Domain	NMS Hub	Origin
doc-demo	doc-demo-hub	doc-demo-hub
NMS Robot	NMS Probe	
se-w2k8-4	hub	

Assignment and suppression details

Assigned to		Suppression count	18
Assign time		Suppression time	Sep 02 15:26:49
Assigned by		Suppression key	

Message details

Origin time	May 23 10:25:06	Source	138.42.229.144
Arrival time	May 23 10:25:10	Hostname	se-w2k8-4
Subsystem ID	1.2.1	Subsystem	Hub

warning

Message text

HUB license will expire in 29 days on /doc-demo/doc-demo-hub/se-w2k8-4/hub

Close

Acknowledge

All new alarm messages received by a NAS are initially considered un-acknowledged and presented to an operator. When the operator has verified whether there was a problem and possibly fixed it, they can acknowledge the message, indicating that the problem no longer exists. The message is then deleted from the NAS database, but a copy is kept in the history database.

Attach Note

Attach a note defined under the Notes tab (if any) to the alarm. See The Notes Tab section for more information.

Create

Allows you to create the following:

Filter Definition

Create a pre-processing rule for the NAS (see the Pre-processing Rules Tab). The Pre-processing rules consist of a filter and a set of rules determining how the NAS will handle alarm messages matching the filter (exclude, set invisible etc).

Auto-Operator Profile

Create an Auto-Operator profile for the NAS (see the Profiles section). The Profile dialog box displays the matching criteria based on the alarm information of the selected alarm.

These profiles describe how to handle the alarms (send SMS, e-mail etc.).

Trigger Definition

Create a Trigger profile for the NAS (see also the Triggers section). This tab allows you to define triggers to "sort" alarm messages based on properties set for the trigger. Alarms matching the criteria defined for the trigger will not be handled by an Auto-operator. You can define triggers, using matching criteria (such as message text, severity, hostname etc.) and time restrictions (defining the periods when the filter should be active).

Note

Create notes that can be attached to alarms. The notes created will be listed under the Notes tab, and can be attached to an alarm by right-clicking the alarm, selecting *Attach Note*.

Visibility

Sets the selected alarm to either visible or invisible. This mode is a filter type for incoming alarms. Alarm messages set to invisible will be managed by the NAS and still be listed in the alarm list (but with gray text), provided that the option Show Invisible Alarms is selected.

Set Invisible

Sets the selected alarm as invisible.

Set Visible

Sets the selected alarm as visible.

Hide Invisible Alarms/ Show invisible Alarms

Hides alarms set to invisible. This option will change to Show Invisible Alarms, letting you show any alarms set to invisible in the list again.

View Historic Alarms

When an alarm message is acknowledged, it is deleted from the NAS database, but it is still kept in a history database.

This option displays the alarms from the selected time frame (today, last hour, last day, and last month).

The alarms will be opened in a separate window.




Clicking the *Filter* icon opens a filter dialog, which allows you to filter the list. Note that the fields in this filter do not support pattern matching or regular expression.

You may, however, use an asterisk (*) as wildcard. This will be interpreted as a "%" which will be built into a database statement.

Comma (",") can also be used to set up two or more criteria.

A screenshot of the 'Selection Filter' dialog box. It has a title bar with a magnifying glass icon and a close button. Inside, there's a 'Severity Level' section with eight color-coded checkboxes (Green, Cyan, Blue, Yellow, Orange, Red, etc.). Below this are several input fields arranged in two columns: 'Hostname', 'Subsystem', 'NMS Domain', 'NMS Robot Name', 'Message string' on the left; and 'Source', 'Subsystem ID', 'NMS Hub Name', 'NMS Probe Name' on the right. Each field has a small square icon to its right. At the bottom right are 'OK' and 'Cancel' buttons.

To activate the filter, you must activate the  **Enable/disable filter** button.

Clicking the  **Refresh** button will refresh the list to show the current content.

Reload

Reloads the alarms from the NAS database

Now

Reloads the status window immediately.

Automatic Reload

Reloads the status window automatically at a given interval.

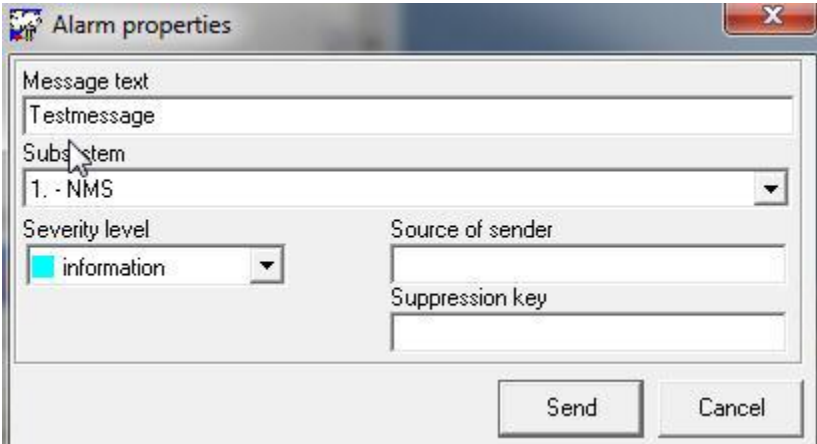
Set Timer

Set the interval for an automatic reload.

Send Test Alarm

Opens the Alarm properties dialog, enabling you to create a test alarm to be sent to the NAS.

Set the properties for the test alarm and click the **Send** button.



The image shows a screenshot of the 'Alarm properties' dialog box. It has a title bar with a close button. Inside, there are several fields: 'Message text' with the value 'Testmessage', 'Subsystem' with a dropdown menu showing '1. - NMS', 'Severity level' with a dropdown menu showing 'information' (highlighted in cyan), 'Source of sender' (empty), and 'Suppression key' (empty). At the bottom right, there are 'Send' and 'Cancel' buttons.

Then left-click in the alarm list and press the F5 key on your keyboard (or click the Reload button) to refresh the alarm list. The test alarm sent should now appear in the alarm list.

Copy To Clipboard

Copies the alarm text to clipboard, enabling you to paste the text to a document, worksheet etc.

Advanced

The Advanced menu option contains the following five options:

Cold-start Alarm Server

Performs a cold-start of the nas.

You will receive a confirmation message. Click **Yes** if you want to continue, otherwise click **Cancel**.

Drop/Delete All Alarms

Deletes all alarms from the NAS database.

You will receive a confirmation message. Click **Yes** if you want to continue, otherwise click **Cancel**.

Reorganize Database

Reorganizes the NAS database. It clears all empty space (deleted alarms leaves empty space).

You will receive a confirmation message. Click **Yes** if you want to continue, otherwise click **Cancel**.

Hold Incoming Alarms (5 min)

Incoming alarms will be put on hold for five minutes before they are processed. This option can be used when the traffic is heavy.

Get Queue Statistics

Displays the current status for the different queues.

Appendix A: The NAS Extentions to Lua

This section describes the nas extensions to the Lua language.

This section contains the following topics:

[Alarm](#) (see page 110)

[Database](#) (see page 112)

[Action](#) (see page 113)

[Nimsoft](#) (see page 114)

[Note](#) (see page 115)

[Trigger](#) (see page 115)

[File](#) (see page 116)

[Timestamp](#) (see page 117)

[PDS](#) (see page 119)

[Language Extension](#) (see page 121)

Alarm

alarm.get ([Nimid])

Returns a table of alarm data for the given *nimid*. If used without the *nimid* it will return the current alarm data, and is only used in conjunction with the *on-arrival* AO profile method.

alarm.list ([Field, Value [, Value...]])

Returns an array of table elements containing alarm data. Will, if used with the field and value(s) parameters, filter the result set according to the user criteria. Use the column name for your field and one or more match strings. The % is used as the wildcard character. E.g alarm.list ("hostname","%xp%") returns alarms for all hostnames with 'xp' in them. These records are extracted from the NAS_ALARMS table.

alarm.transactions (Nimid)

Returns an array of table elements containing alarm transaction information for the given nimid. These records are extracted from the NAS_TRANSACTION_LOG table, in the transaction log database.

alarm.statistics ([ShowAll [, Field, Value]])

Returns a table containing the following items:

level_information	-	number of informational alarms.
level_warning	-	warnings
level_minor	-	minor
level_major	-	major
level_critical	-	critical
alarm_count	-	total number of alarms.

ShowAll = **true** will list all visible and invisible alarms. You may use the *Field* and *Value* parameters to selectively choose statistics. You may choose one of *origin*, *hostname*, *source*, *subsystem*, *sid*.

alarm.history (Selector [, Option])

Returns an array of table elements containing alarm summary records for the selected time period.

The valid selectors are the ones found in the *history browser*, namely: *today*, *lasthour*, *lastweek*, *lastmonth*, *last24hours*, *last7days*, *last3days*, *date*. The Option parameter when used with one of the selectors mentioned, can be one of *time*, *closed* or *creted*. The Option parameter when used with the selector "where" is a valid SQL WHERE statement (without the WHERE).

The date selector is on the form date,yyyy-mm-dd [HH [:MM [:SS]]] , yyyy-mm-dd [HH [:MM [:SS]]]

For example: date,2007-10-18, 2007-10-22 08:00

alarm.query (SQL-Query [, Token])

Runs the SQL-Query in the NAS database unless Token is specified. Token may currently be "transactionlog". Returns the result of the SQL-Query. Please take caution using this function. No checks are performed, and the caller may corrupt the database or database model by running bad queries.

alarm.set (AlarmTable)

Updates the existing alarm denoted by the nimid element of the AlarmTable with a set of supported fields in the alarm. The fields are message, level or severity, sid, user_tag1, user_tag2, visible and escalated. The visible and escalated take 0 (false) and 1 (true) as values. The message, severity, sid, user_tag1, user_tag2 take strings as values.

For example:

```
e = {}
```

```
e.nimid = "TY25224233-56765"
```

```
e.message = "This is a modified text."
```

```
e.level = NIML_WARNING
```

```
e.sid = "1.1.3"
```

```
e.visible = 0
```

```
alarm.set (e)
```

Database

database.open ([FileName | ConnectionString])

Opens a database handle to the specified file or database. Subsequent *database* operations will now be reference through this handle, until it is closed using the *database.close* or through an implicit close when opening another database using *database.open*. The default database is called *user.db*.

For example: `database.open("myprivate.db")`

database.query (SQL-Query)

Performs the provided SQL in the current open database. If no previous *database.open* has been performed then the *user.db* is used. The SQL statement must be supported by the underlying database.

database.close ()

Closes the current database.

database.setvariable (Name, Value)

Creates (or modifies) the persistent variable *Name* in the current database. The variable name should be a unique name to avoid collisions.

database.getvariable (Name)

Retrieves the persistent variable *Name*. The function returns *nil* when the variable is non-existent.

Action

action.assign (AssignTo, NimId | NimId-List [, AssignedBy])

Assigns a user to one or more alarms, using the *nimid* (or a comma separated list of *nimids*).

action.close (NimId | NimId-List)

Closes the open alarm referenced by the single *nimid* or the list of alarm ids.

action.command (CommandLine)

Executes the provided command-line string, and places the output (if any) into a table of lines.

action.note (NoteName, NoteDescription, NimId [, Overwrite])

Create and attach a note to the alarm message referenced by the *nimid*.

action.ping (HostName [, Timeout])

Returns the status (true or false) and the time-used (in milliseconds) when issuing a ping (ICMP ECHO) to the provided hostname or ip-address.

action.email (ReceiverAddress, Subject [, Body])

Generates an email-message targeted for the Nimsoft Email Gateway.

action.SMS (PhoneNumber, MessageText)

Generates an SMSI-message targeted for the Nimsoft SMS Gateway.

action.profile (Name, RunState [,Persistent])

Activates or deactivates the named Auto-Operator profile. A persistent change will affect the configuration file. Note that `action.profile()` returns a table of all filters with status information.

action.filter (Name, RunState [,Persistent])

Activates or deactivates the named Auto-Operator pre-processing filter. A persistent change will affect the configuration file. Note that `action.filter()` returns a table of all filters with status information.

action.log (Activity [, Status [,TimeUsed [,Module [, Identifier]]]])

Adds activity information to the activity logger. Describe it as precise as possible, and use the status information to flag different states, or results of operations.

action.visibility (Visible, NimId | NimID-List)

Set alarm visibility to **true** or **false** on one or more alarms.

action.escalate (SeverityLevel, NimId | NimID-List)

Raises the severity level to according to the SeverityLevel parameter. Only alarms with a current severity level lower than SeverityLevel will be modified.

Nimsoft

nimbus.alarm (SeverityLevel, MessageText [, SuppressionKey [,SubsystemId]])

Generates a Nimsoft alarm message with the severity level (1-5) and a message-text. Use the suppression-key to create a stateful alarm.

nimbus.post (Subject, PDSHandle)

Posts a Nimsoft Message onto the Nimsoft using the Subject.

Returns a message-id string if successful or **nil**

nimbus.request (NimsoftAddress, Command, Arguments [, Wait [, ReturnAsPDS]])

Returns the result of the command targeted for the provided Nimsoft component. The command-arguments are expected to be a PDS (returned by pds.create). The result is placed into a table unless the ReturnAsPDS parameter is set to *true*.

Please note that this is an associative table (not indexed), meaning that a PDS sections will be referenced by its section-name.

controller = Nimsoft.request ("controller","get_info")

printf ("controller robot: %s", controller.robotname)

nimbus.qos_definition (QosName, QosGroup, Description, Unit, UnitAbbreviation, HasMax [, IsAsynch])

Creates a QoS definition named *QosName*. Unless the flag *IsAsynch* is *true*, an interval based QoS is created. Please note that subsequent definitions on the same name will not recreate or alter an existing QoS definition. The *HasMax* flag set requires that all qos data (issued by *Nimsoft.qos*) referring to this *QoSName* is issued with a *MaxValue*.

nimbus.qos (QosName, Source, Target, Value, Interval | QOS_ASYNC [MaxValue])

Will send an interval based QoS message when *Interval* is greater than zero, and a asynchronous QoS message when called with *QOS_ASYNC*. Please note that no QoS data will be recorded unless a valid QoS definition has been sent prior to this request. Remember to set the *MaxValue* if definition was created using *HasMax=true*.

Note

note.create (Name, Description)

Creates a note with the provided name and description fields set, and returns the note identification number (Noteld).

note.append (Name | Noteld, Description [, Overwrite])

Appends the descriptive text to an existing note defined by the name or the id. A new note will be created when no matches are found. Returns status.

note.delete (Name | Noteld)

Deletes a note with the provided name or id. Returns status.

note.find (Name)

Returns the Noteld of the named note, and the note description as the second return parameter, or **nil** when nothing matches the provided Name.

note.attach (Name | Noteld, Nimld [, Nimld...])

Attaches the note to one or more alarms specified as Nimlds.

Trigger

trigger.alarms (TriggerName)

Returns an array of table elements containing alarm data matching the criteria for the named trigger.

trigger.count (TriggerName)

Returns the number of alarm events currently matching the trigger criteria.

trigger.state (TriggerName)

Returns the state (raised or not raised) of the named trigger.

trigger.timestamp (TriggerName)

Returns the UTC timestamp when the trigger last changed state.

File

file.copy (Source, Destination)

Creates a file using the complete *Path* and writes *Buffer* into the file if provided.

file.create (Path [, Buffer])

Creates a file using the complete *Path* and writes *Buffer* into the file if provided.

file.delete (Path)

Deletes the file named *Path*.

file.read (Path [,Mode])

Returns a buffer with the file contents, and the number of bytes read as a second return parameter. The optional *mode* parameter allows for controlling the open-mode. (see fopen man-pages, default: "rb")

file.write (Path , Buffer)

Appends *Buffer* the file *Path*, and returns **true** if success

file.stat (Path)

Returns a table containing the following statistics: *mtime*, *ctime*, *atime*, *mode* and *size*.

file.rename (OldName , NewName)

Renames the file OldName to NewName.

Timestamp

timestamp.now ()

Returns the number of seconds elapsed since Jan. 1 1970, 00:00:00

timestamp.diff (StartTimeStamp [, Format [, EndTimeStamp]])

Returns the difference (seconds, minutes, hours or days) between the EndTimeStamp (or *now* if not provided) and the StartTimeStamp using the Format specifier (seconds, minutes, hours, day)

timestamp.newer (TimeStamp, TimeSpecification)

Returns **true** if the TimeStamp is newer than specified by the TimeSpecification. The TimeSpecification format is built using a combination of numbers and the tokens: seconds, minutes, hours, days. E.g. 10h30min, 5hrs, 30m, 3 days

timestamp.older (TimeStamp, TimeSpecification)

Returns **true** if the TimeStamp is older than specified by the TimeSpecification. The TimeSpecification format is built using a combination of numbers and the tokens: seconds, minutes, hours, days. E.g. 10h30min, 5hrs, 30m, 3 days

timestamp.data ([TimeStamp])

Uses '*now*' if no parameter is provided. Returns a table with the following self-explanatory members:
year, month, day, hour, minute, second, yearofday, weekday and isdst (1 if daylight savings time).

timestamp.fromISO (ISOdatestring)

Returns a timestamp and a timestamp data table (see *timestamp.data*).

timestamp.format (TimeStamp [, Format])

Returns a formatted timestring using the Format specifier (default: %b %d, %H:%M:%S).

specifier	Replaced by	Example
%a	Abbreviated weekday name *	Thu
%A	Full weekday name *	Thursday
%b	Abbreviated month name *	Aug
%B	Full month name *	August
%c	Date and time representation *	Thu Aug 23 14:55:02 2001
%d	Day of the month (01-31)	23
%H	Hour in 24h format (00-23)	14
%I	Hour in 12h format (01-12)	02

%j	Day of the year (001-366)	235
%m	Month as a decimal number (01-12)	08
%M	Minute (00-59)	55
%p	AM or PM designation	PM
%S	Second (00-61)	02
%U	Week number with the first Sunday as the first day of week one (00-53)	33
%w	Weekday as a decimal number with Sunday as 0 (0-6)	4
%W	Week number with the first Monday as the first day of week one (00-53)	34
%x	Date representation *	08/23/01
%X	Time representation *	14:55:02
%y	Year, last two digits (00-99)	01
%Y	Year	2001
%Z	Timezone name or abbreviation	CDT
%%	A % sign	%

* The specifiers whose description is marked with an asterisk (*) are locale-dependent.

PDS

The PDS (Portable Data Stream) format is used heavily within the Nimsoft to exchange data between various processes on all platforms supported by Nimsoft. This format allows users to build nested datastructures that may be passed between different languages and different hardware platforms.

pds.create ()

Returns a reference handle to a PDS structure.

pds.delete (pdsHandle)

Deletes the PDS structure and data.

pds.convert (pdsHandle)

Returns a LUA table. This function converts the PDS structure to a LUA table containing the same key/value pairs and sub-tables (if any).

pds.putInt (pdsHandle, Key, Value)

Stores an integer value in the provided PDS structure using the Key as the reference to the Value.

Note that an existing element with the same Key will be replaced.

pds.putString (pdsHandle, Key, Value)

Stores a string in the provided PDS structure using the Key as the reference to the Value. Note that an existing element with the same Key will be replaced.

pds.putDouble (pdsHandle, Key, Value)

Stores a double value in the provided PDS structure using the Key as the reference to the Value. Note that an existing element with the same Key will be replaced.

pds.putPDS (pdsHandle, Key, Value)

Stores a PDS in the provided PDS structure using the Key as the reference to the Value.

Note that an existing element with the same Key will be replaced.

pds.getInt (pdsHandle, Key)

Returns the number extracted associated by Key from the provided PDS structure (or **nil** if non-existent).

pds.getString (pdsHandle, Key)

Returns the string value extracted associated by Key from the provided PDS structure (or **nil** if non-existent).

pds.getDouble(pdsHandle, Key)

Returns the number extracted associated by Key from the provided PDS structure (or **nil** if non-existent).

pds.getPDS (pdsHandle, Key)

Returns the PDS handle extracted associated by Key from the provided PDS structure (or **nil** if non-existent).

pds.getNext (pdsHandle)

Returns the next Key, Type, DataSize, Data from the provided PDS structure (or **nil** if non-existent).

Language Extension

sprintf (Format [,Par1 [,Par2 [...]]])

Returns a string buffer with the formatted string.

printf (Format [,Par1 [,Par2 [...]]])

Logs the formatted string to the output window (if in the editor) or the NAS logfile.

print (Par1[,Par2 [...]])

Logs the string to the output window (if in the editor) or the NAS logfile. Used primarily for simple unformatted printing and debug output.

left (String, Length)

Returns *Length* characters from the *String*, starting from the left.

right (String, Length)

Returns *Length* characters from the *String*, starting from the right.

mid (String, Start [, Length])

Returns *Length* characters from the *String*, starting from *Start*. If no *Length* is specified, the rest of the string will be returned.

substr (String, Substring)

Returns **true** if the *Substring* is found, as well as the starting *Position* of the substring.

split (String [, Separators])

Returns a table of substrings separated by one or more of the Separator characters. The default separator is whitespace.

trim (String [, Mode])

Removes leading and/or trailing white space from String.

The Mode may be 0: leading and trailing, 1: leading, 2: trailing. The default value is 0.

regexp (String, Expression)

Returns **true** if the regular (or pattern matching) expression matches *String*.

setvariable (Name, Value)

Stores the non-persistent variable *Name*. The value is retrievable until a cold-start of the NAS clears the

Non-persistent data store. Use the equivalent **database.setvariable** for a persistent store.

getvariable (Name)

Returns the non-persistent named variable *Name* or **nil** if non-existent.

exit (ExitCode)

Terminates the script execution with an *ExitCode*. Non-zero *ExitCodes* will be recorded in the NAS activity-log.

tonumber (Value)

Converts *Value* into a number.

tostring (Value)

Converts *Value* into a string.

type (Value)

Returns the variable type as a string.

state (TriggerName)

This is a shortcut for the **trigger.state** function.

sleep (MilliSeconds)

Suspends execution for a given time.

Constants

NIML_CLEAR	= 0
NIML_INFORMATION	= 1
NIML_WARNING	= 2
NIML_MINOR	= 3
NIML_MAJOR	= 4
NIML_CRITICAL	= 5
QOS_ASYNC	= -1
NAS_AO_INTERVAL	= from the current NAS configuration.
NAS_NAME	= the name of the current NAS.
NAS_ADDRESS	= the Nimsoft address of the current NAS.
SCRIPT_NAME	= the name of the executing script.
SCRIPT_FILE	= the filename of the executing script.
PROFILE_NAME	= the AO profile that executed the script (if any).
PROFILE_STATE	= the state of the profile when using the <i>on_trigger</i> method.

Table structures

As returned from `alarm.list()`, `alarm.get()`:

.nimid	- unique Nimsoft Id
.nimts	- timestamp when the alarm was created (at source)
.source	- source of the alarm (typically ip-address)
.hostname	- resolved name (robotname or ip-address to name resolution)
.level	- severity level (0-5)
.severity	- textual representation of the severity level.
.supptime	- timestamp of last suppression.
.sid	- subsystem identification.
.subsys	- subsystem string resolved from <i>sid</i> .
.message	- alarm message text.
.suppcount	- number of times event has been suppressed.

.supp_key	- suppression identification key.
.origin	- origin of the alarm (stamped by nearest hub, or in some cases the robot.)
.domain	- name of originating Domain.
.robot	- name of the sending robot.
.hub	- name of the nearest hub to the sending robot.
.nas	- name of originating alarm server.
.prid	- name of probe issuing the alarm.
.user_tag1	- user tag 1 (as set by robot).
.user_tag2	- user tag 2 (as set by robot).
.visible	- flag for visibility (1 = visible)
.aots	- AO timestamp
.arrival	- timestamp when alarm arrived at NAS.
.time_arrival	- datetime of <i>arrival</i> .
.time_supp	- datetime of <i>supptime</i> .
.time_origin	- datetime of <i>nimts</i> .
.assigned_at	- datetime at assignment.
.assigned_to	- user alarm is assigned to.
.assigned_by	- the user who assigned the alarm.
.tz_offset	- timezone offset (seconds from GMT)
.supp_id	- checksum of suppression information.
.change_id	- checksum of message, severity and subsystem.

As returned by alarm.transactions(), alarm.history():

.source	- source of the alarm (typically ip-address)
.hostname	- resolved name (robotname or ip-address to name resolution)
.level	- severity level (0-5)
.severity	- textual representation of the severity <i>level</i> .
.time	- datetime of event.
.sid	- subsystem identification.
.subsys	- subsystem string resolved from <i>sid</i> .
.message	- alarm message text.
.suppcount	- number of times event has been suppressed.
.origin	- origin of the alarm (stamped by nearest hub, or in some cases the robot.)
.domain	- name of originating Domain.
.robot	- name of the sending robot.
.hub	- name of the nearest hub to the sending robot.
.nas	- name of originating alarm server.
.prid	- name of probe issuing the alarm.
.user_tag1	- user tag 1 (as set by robot).
.user_tag2	- user tag 2 (as set by robot).
.visible	- flag for visibility (1 = visible)
.assigned_to	- user alarm is assigned to.
.assigned_by	- the user who assigned the alarm.
.acknowledged_by	- the user who acknowledged the alarm.
.tz_offset	- timezone offset (seconds from GMT)

type - transaction type (New, Suppressed major/minor, Acknowledged, Assigned, Closed) only returned by **alarm.transactions()**.

As returned by alarm.statistics():

- .level_clear - number of open alarms with severity level *clear*.
- .level_information - number of open alarms with severity level *information*.
- .level_warning - number of open alarms with severity level *warning*.
- .level_minor - number of open alarms with severity level *minor*.
- .level_major - number of open alarms with severity level *major*.
- .level_critical - number of open alarms with severity level *critical*.
- .alarm_count - number of open alarms.
- .oldest_alarm - timestamp of the oldest open alarm.
- .newest_alarm - timestamp of the newest open alarm.

Custom Pre-Processing

The event table is placed into the LUA context prior to executing the "custom" pre-processing rule. You may alter (launder) the event by setting the fields message, sid, source, hostname, user_tag1, user_tag2, visible and origin. The following fields are present for the script to use:

- .source - source of the alarm (typically ip-address)
- .hostname - resolved name (robotname or ip-address to name resolution)
- .level - severity level (0-5)
- .sid - subsystem identification.
- .message - alarm message text.
- .origin - origin of the alarm (stamped by nearest hub, or in some cases the robot.)
- .domain - name of originating Nimsoft domain.
- .robot - name of the sending robot.
- .hub - name of the nearest hub to the sending robot.
- .prid - name of probe issuing the alarm.
- .user_tag1 - user tag 1 (as set by robot).
- .user_tag2 - user tag 2 (as set by robot).
- .supp_key - suppression identification key.
- .visible - flag for visibility (true = visible)

The script is expected to return the event (modified or not) or nil. A nil indicates that the event is to be skipped.

Note that the user_tag1 and user_tag2 fields will be stored in the database when the inbound alarm translates into a new event.

Note that all pre-processing handling will, by nature, slow down the processing of inbound alarms.

Note that only a subset of the lua methods are available to the pre-processing script. The trigger.state method, through the state method, is available. These classes and methods are not available:

- exit
- sleep
- Nimsoft
- pds
- trigger
- action
- database
- alarm
- note.

Appendix B: The NAS Command Interface

This section describes the NAS command interfaces. All commands return a status value such as NIME_OK (0), NIME_ERROR (1) or NIME_INVALID (7). Details are documented under each command. The 'list' type commands yields a PDS data structure.

This section contains the following topics:

[assign_alarms](#) (see page 127)
[close_alarms](#) (see page 128)
[date_forecast](#) (see page 128)
[db_query](#) (see page 128)
[get_alarms](#) (see page 129)
[get_ao_status](#) (see page 129)
[get_info](#) (see page 130)
[get_sid](#) (see page 130)
[host_summary](#) (see page 130)
[nameservice_create](#) (see page 130)
[nameservice_delete](#) (see page 131)
[nameservice_list](#) (see page 131)
[nameservice_lookup](#) (see page 131)
[nameservice_setlock](#) (see page 131)
[nameservice_update](#) (see page 132)
[note_attach](#) (see page 132)
[note_create](#) (see page 132)
[note_delete](#) (see page 133)
[note_detach](#) (see page 133)
[note_list](#) (see page 133)
[Reorganize](#) (see page 133)
[repl_queue_post](#) (see page 133)
[repl_queue_info](#) (see page 134)
[script_delete](#) (see page 134)
[script_rename](#) (see page 134)
[script_list](#) (see page 134)
[script_run](#) (see page 134)
[script_validate](#) (see page 135)
[set_loglevel](#) (see page 135)
[set_visible](#) (see page 135)
[transaction_list](#) (see page 135)
[trigger_list](#) (see page 136)

assign_alarms

Parameter	Type	Req	Description
-----------	------	-----	-------------

by	string	*	<i>specifies who assigned the alarm(s)</i>
to	string	*	<i>specifies to whom the alarm is assigned to</i>
nimid	string	*	<i>alarm message-id</i>
nimids	array		<i>a string table of message-ids</i>

close_alarms

Parameter	Type	Req	Description
by	string	*	<i>specifies who closed the alarm(s)</i>
nimid	string	*	<i>alarm message-id</i>
nimids	array		<i>a string table of message-ids</i>

date_forecast

Parameter	Type	Req	Description
specification	string	*	<i>RFC-2445 compliant string</i>
startdate	string		<i>ISO starting date of forecast. yyyy-mm-dd hh:mm:ss</i>
nitems	number		<i>number of dates in forecast.</i>
format	string		<i>strftime format specifiers.</i>

This command returns a string array with dates, and the number of dates in the forecast. The current time is used as the default startdate.

db_query

Parameter	Type	Req	Description
sql	string	*	<i>SQL-92 conformant statement</i>
db	string	*	<i>database</i>

get_alarms

Parameter	Type	Req	Description
show_all	number		<i>flag showing all alarms visible and invisible</i>
origin	string		<i>filter alarms using origin field</i>
hostname	string		<i>filter alarms using hostname field</i>
source	string		<i>filter alarms using source field</i>
severity	string		<i>filter alarms using severity field</i>
subsystem	string		<i>filter alarms using subsystem field</i>
assigned_to	string		<i>filter alarms using assigned_to field.</i>

The show_all parameter takes the following values:

0: show only visible alarms.

1: show all alarms with visibility flag in alarm record.

All filter items are *italic* and may be used together. The following syntax is assumed:

[not] [like] value [,value [...]] | null

e.g assigned_to = not null

assigned_to = administrator

assigned_to = null

hostname = like %xp%

Compatibility note:

The mask parameter (used by e.g the Alarm Notifier) is supported as a non-public variable, hence not being visible.

get_ao_status

Parameter	Type	Req	Description
mode	string	*	<i>a combination of triggers, profiles, schedules and filters.</i>
detail	number	*	

get_info

Parameter	Type	Req	Description
detail	number		<i>shows current connections if set to 1.</i>
show_all	number		<i>flag showing all alarms visible and invisible</i>
origin	string		<i>filter alarms using origin field</i>
hostname	string		<i>filter alarms using hostname field</i>
source	string		<i>filter alarms using source field</i>
severity	string		<i>filter alarms using severity field</i>
subsystem	string		<i>filter alarms using subsystem field</i>
assigned_to	string		<i>filter alarms using assigned_to field.</i>

Please see get_alarms for the parameter settings.

get_sid

Parameter	Type	Req	Description
sid	string		<i>specific subsystem identifier e.g. 1.1.1</i>

Returns all subsystem names configured or the one specified by *sid*.

host_summary

Parameter	Type	Req	Description
mode	string		<i>one of: today, lasthour, last24hours, last3days, lastmonth and date=ISO-startdate,ISO-enddate. E.g. date=2007-08-24,2007-08-27</i>

Returns a list of hosts that has alarms in the period specified by the *mode*.

nameservice_create

Parameter	Type	Req	Description
-----------	------	-----	-------------

<i>ip</i>	string	*	<i>ip-address to be used as lookup key.</i>
<i>name</i>	string	*	<i>name to be used in name-resolution</i>
<i>lock</i>	<i>number</i>		<i>specifies if this should be locked (1=locked)</i>

Adds a nameservice record.

nameservice_delete

Parameter	Type	Req	Description
<i>ip</i>	string	*	<i>ip-address to be removed.</i>

nameservice_list

Returns a PDS table (named *table*) with records containing *ip,name,ts* and *time*.

nameservice_lookup

Parameter	Type	Req	Description
<i>ip</i>	string		<i>ip-address to resolve.</i>
<i>name</i>	string		<i>hostname to resolve.</i>

Returns the result of the nameservice lookup, note that either *ip* or *name* must be set.

nameservice_setlock

Parameter	Type	Req	Description
<i>ip</i>	string	*	<i>ip-address to lock/unlock.</i>
<i>lock</i>	number		<i>locks (1) or unlocks (0) the name-ip mapping.</i>
<i>ips</i>	array		<i>Array of ip-addresses to lock/unlock</i>

This command expects *ip* or *ips* to be set.

nameservice_update

Parameter	Type	Req	Description
ip	string		<i>ip-address to modify.</i>
name	string	*	<i>name to be used in the name-resolution.</i>
lock	number		<i>locked (1) or unlocked(0)</i>

This command expects *ip* or *ips* to be set.

note_attach

Parameter	Type	Req	Description
note_id	number	*	<i>id of existing note to attach to alarm (nimid), or zero (0) if a create+attach is performed.</i>
nimid	string	*	<i>alarm message-id that we want to attach note to</i>
description	string		<i>note description (if create)</i>
body	string		<i>note body (if create)</i>
category	string		<i>note category (if create)</i>
nimids	array		<i>a table of alarm message-ids</i>

This command primarily attaches an existing note to one or more alarms. However, it can also perform a "create and attach" in a single operation. Specify this operation by setting note_id = 0 (zero).

note_create

Parameter	Type	Req	Description
note_id	number	*	<i>id of existing note (if edit) or zero (0) if a new note is created.</i>
description	string	*	<i>note description</i>
body	string		<i>note body</i>
category	string		<i>note category</i>
autoremove	number		<i>auto-remove when last alarm reference is cleared.</i>

note_delete

Parameter	Type	Req	Description
<i>note_id</i>	<i>number</i>	*	<i>note id to delete</i>

note_detach

Parameter	Type	Req	Description
<i>note_id</i>	<i>number</i>	*	<i>id of note to remove from the alarm.</i>
<i>nimid</i>	<i>string</i>	*	<i>alarm message-id that we want to remove note from.</i>
<i>nimids</i>	<i>array</i>		<i>a table of alarm message-ids</i>

note_list

Parameter	Type	Req	Description
<i>nimid</i>	<i>string</i>		<i>alarm message-id that we list notes for.</i>

Reorganize

Parameter	Type	Req	Description
<i>by</i>	<i>string</i>	*	<i>specifies who requested the database reorganize.</i>

The reorganize command will take the NAS into maintenance mode, stopping all service modules and performs a VACUUM of the *database.db* and the *transactionlog.db*. All services are started upon completion.

repl_queue_post

Parameter	Type	Req	Description
<i>name</i>	<i>string</i>	*	<i>specifies which NAS posts replication data.</i>

This is a private interface used by the NAS replication service.

repl_queue_info

Parameter	Type	Req	Description
<i>name</i>	<i>string</i>		<i>specifies which queue to get information about.</i>

script_delete

Parameter	Type	Req	Description
<i>name</i>	<i>string</i>	*	<i>specifies the script to delete (including directory path)</i>

script_rename

Parameter	Type	Req	Description
<i>directory</i>	<i>string</i>	*	<i>specifies the path where the scripts resides.</i>
<i>from</i>	<i>string</i>	*	<i>old name</i>
<i>to</i>	<i>string</i>	*	<i>new name</i>

script_list

Returns a PDS containing a string table with all scripts (including path), as well as the actual script root directory.

script_run

Parameter	Type	Req	Description
<i>name</i>	<i>string</i>	*	<i>specifies the script run</i>
<i>profile</i>	<i>string</i>		<i>if script is to be executed in a AO-profile context</i>

script_validate

Parameter	Type	Req	Description
name	string	*	<i>specifies the script run</i>
profile	string		<i>if script is to be validated in a ao-profile context</i>
code	string	*	<i>lua code to be validated</i>
evaluate	number		<i>get returnvalue from script.</i>

set_loglevel

Parameter	Type	Req	Description
level	number	*	<i>specifies the nas loglevel.</i>

set_visible

Parameter	Type	Req	Description
visible	number	*	<i>sets the alarms visible (1) or invisible(0)</i>
nimid	string	*	<i>alarm message-id</i>
nimids	array		<i>a string table of message-ids</i>

Either *nimid* or *nimids* is required.

transaction_list

Parameter	Type	Req	Description
mode	string	*	<i>one of: today, lasthour, last24hours, last3days, lastmonth and date=ISO-startdate,ISO-enddate. E.g. date=2007-08-24,2007-08-27</i>
where	string		<i>valid SQL-92 conformant WHERE clause.</i>
nimid	string		<i>alarm message-id</i>

Returns a list of alarms that occurred in the period specified by *mode*. If *nimid* is specified then the events for that particular alarm-id are listed.

trigger_list

Parameter	Type	Req	Description
name	string		<i>name of trigger to list</i>
detail	string		<i>detail level, no-detail is zero(0), show events is 1.</i>