AlertMailer v1.04 User Guide

Nanometrics Inc. Kanata, Ontario Canada © 2003 Nanometrics Inc. All Rights Reserved.

AlertMailer v1.04 User Guide

The information in this document has been carefully reviewed and is believed to be reliable. Nanometrics, Inc. reserves the right to make changes at any time without notice to improve the reliability and function of the product.

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Nanometrics Inc.

This product uses the JavaTM 2 Runtime Environment (J2RE) Standard Edition, which includes code licensed from RSA Security, Inc. Some portions licensed from IBM are available at http://oss.software.ibm.com/icu4j/

Nanometrics, Inc. 250 Herzberg Road Kanata, Ontario, Canada K2K 2A1 Tel (613)592-6776 Fax (613)592-5929 Email info@nanometrics.ca

 Part number
 14659R1

 Release date
 2003-01-20

Contents

AlertMailer v1.04	1
1 About AlertMailer. 1 1.1 Message content. 2	1 1
2 Installing AlertMailer 2 2.1 Prerequisites 2 2.1.1 Hardware minimum requirements 2 2.1.2 Operating system 2 2.2 Install Nanometrics Dynamic Link Libraries (DLLs) 2 2.3 Install Java Runtime Environment (JRE) 2 2.4 Install AlertMailer 2	2 2 2 2 2 2 2 2 3
3 Running AlertMailer. 3 3.1 Starting AlertMailer 3 3.2 Starting AlertMailer from NmxWatchdog. 3 3.3 Stopping AlertMailer 3 3.4 Using the AlertMailer command window. 3 3.5 Monitoring acquisition system components 4	3 3 3 3 3
4 Definition of inifile sections and parameters 5 4.1 [Log] 6 4.2 [AlertReceiver] 6 4.3 [SystemMonitor] 6 4.4 [MailSender] 6 4.5 [Thresholds] 6 4.6 [MailGroups] 6 4.7 [Recipients] 10	566890
5 Setting up a translation file 10 5.1 Argument substitution 11 5.2 Message translation example 12 Inifile Example 12) 1 1 2
	J

AlertMailer v1.04

1 About AlertMailer

AlertMailer is a utility program that forwards alert messages from NaqsServer and other Nanometrics programs to a user-defined list of email recipients. Alert messages are generated by various Nanometrics programs and instruments in response to significant system or seismic events. These messages are broadcast via unicast or multicast UDP to a central message handler for translation (optional) and transmission. Alert-Mailer forwards these messages by email.

In addition to forwarding messages from system components, AlertMailer will monitor incoming messages from system components and generate alert messages. This feature can be used to monitor overall acquisition system health.

The system is highly configurable: each message type can be sent to a different list of recipients, and messages which are deemed unimportant can be suppressed altogether. Also, each message can be reformatted or translated into a different language through a user-defined translation file.

1.1 Message content

Each alert message contains the fields described in Table 1. Each message has a default format, but AlertMailer allows the user to reformat messages by specifying a new format string. See also sections "Setting up a translation file" on page 10 and "Argument substitution" on page 11.

Message field	Description
sourceld	The name of the component which generated the message
classId	The specific message type, which indicates its meaning
priority	A numeric measure of the message importance
time	The time in UT at which the message was generated
content	A format string and a list of arguments

Table 1 Alert message fields

1

2 Installing AlertMailer

2.1 Prerequisites

2.1.1 Hardware minimum requirements

- Processor: 166 MHz or faster Intel or compatible processor
- RAM capacity: 128MB or higher
- SVGA monitor: 800x600 resolution
- Hard disk space required (these requirements do not include requirements for the operating system):
 - Nanometrics application code: 5MB
 - Log file capacity: allow 5MB per year (with typical use, and with no deletion of log files)
 - JRE: 50MB

2.1.2 Operating system

- Windows NT 4.0 workstation or server
- Windows 2000/XP
- Windows 95/98/ME
- Solaris

2.2 Install Nanometrics Dynamic Link Libraries (DLLs)



Note If the Nanometrics DLLs have already been installed for another Nanometrics application, you do not need to install them again.

- 1. From a command prompt or Windows Explorer, open the installation CD directory Win32\DLL\[version number]
- 2. Run the program install.bat

The install program will create a c:\nmx\bin and a c:\nmx\user directory on your computer, and copy all program and library files into the c:\nmx\bin and all configuration and data files into the c:\nmx\user directory.

2.3 Install Java Runtime Environment (JRE)

Note If JRE has already been installed for another Nanometrics application, you do not need to install it again.

- 1. From a command prompt or Windows Explorer, open the installation CD directory Java\JRE
- 2. Run the program jre1_4_1-win-i

(Note that this program may have been replaced with a newer version after this manual was released.)

3. Follow the instructions given by the setup program.

2.4 Install AlertMailer

- 1. From a command prompt or Windows Explorer, open the installation CD directory Win32\AlertMailer\[version number]
- 2. Run the program install.bat

The install program will copy all program and library files into the c:\nmx\bin directory and all configuration and data files into the c:\nmx\user directory on your computer (these directories were created during DLL installation).

- 3. Copy the files c:\nmx\user\activation.jar and c:\nmx\user\mail.jar to the directory %JRE%\lib\ext, where %JRE% is the installed location of the java runtime environment (typically C:\Program Files\java\j2re1.4.1).
- 4. Check the system path, and add the directory c:\nmx\bin if it is not already included.

3 Running AlertMailer

3.1 Starting AlertMailer

➤ To start AlertMailer from the command line, type either AlertMailer or AlertMailer *inifile* (where *inifile* is the path to the AlertMailer configuration file). If no inifile parameter is specified, AlertMailer looks for the configuration file named alertmailer.ini in its working directory. The format of the configuration file is described in the section "Definition of inifile sections and parameters" on page 5.

3.2 Starting AlertMailer from NmxWatchdog

To set AlertMailer to be started and monitored automatically by the Nanometrics watchdog program, add the following entry to your watchdog.ini file:
 [WatchEntry]
 ProgramTitle = AlertMailer

ProgramPathname = "java -cp c:\nmx\bin\AlertMailer.jar AlertMailer [inifile]" WorkingDirectory = "c:\nmx\user" ExitAction = Restart PingsSemaphore = true StartDelay = 6s

3.3 Stopping AlertMailer



Note It is important that AlertMailer be shut down properly in order for the application to release its system resources.

 To stop AlertMailer properly, in the AlertMailer command window type quit <Enter>

3.4 Using the AlertMailer command window

The AlertMailer command window (Figure 1) displays log messages showing both incoming and outgoing messages. It also supports a limited keyboard interface, with the options described in Table 2.

To do this	Type this command
Display all log messages in the log file; set the log verbosity to DEBUG	D
Suppress debug messages in the log file; set the log verbosity to VERBOSE	V
Suppress debug and verbose messages in the log file; set the log verbosity to INFO	I
Move the log file (close the current log and start a new file)	Μ
Set the mail format to normal; (see also the Format parameter description in Table 7)	FN
Set the mail format to debug; (see also the Format parameter description in Table 7)	FD
Set the mail format to brief; (see also the Format parameter description in Table 7)	FB
Save email messages to file before mailing; turn the save mail option on	S+
Do not save email messages before mailing; turn the save mail option off	S-
Stop AlertMailer and exit	Quit

Table 2 AlertMailer command options



Command Prompt - alertma	iler	
T 2003 01 17 20.10.22	Alant Madler day (A)	Come to file entire DICODIED
1 2003-01-17 20-10-23	AlertMailSender.(6)	Save-to-file option Dishbury
1 2003-01-17 20-10-23	HierthallSender.(7)	Hall format is MUKHHL
I 2003-01-17 20:18:23	AlertMailSender.(1)	Loaded mail properties from NagsMail.
properties		
1 2003-01-17 20:18:23	AlertReceiver(3)	Receiving on multicast group 224.1.1.
101		
I 2003-01-17 20:18:23	AlertReceiver(2)	starting with priority 10
S+		
I 2003-01-17 20:18:30	AlertMailSender.(6)	Save-to-file option ENABLED

3.5 Monitoring acquisition system components

In addition to forwarding alert messages from other programs and instruments, Alert-Mailer can be used to monitor incoming messages from system components and then generate alert messages.

These alert messages can then be sent to users, using distribution and format options defined in the configuration file and the optional translation file. AlertMailer internal message types are described in Table 3.

For example, critical components such as NaqsServer and Carina issue a "ping" message every 30 seconds. If AlertMailer does not receive a "ping" from a monitored component for 5 minutes or more, it will issue an alert to indicate that the component has gone offline. The components to be monitored must be specified in the [SystemMonitor] section of the configuration file alertmailer.ini.

Message	Description
AlertSystemUp	Indicates that AlertMailer has been started, and provides a list of the monitored components which are online and offline. Priority = 1.
AlertSystemDown	Indicates that AlertMailer is being shut down. Priority = 1.
ComponentOffline	Indicates that AlertMailer has not received a message from the given component for a specified time. The first message (priority 5) is issued after 5 minutes; a second message (priority 20) is issued after 20 minutes.
ComponentOnline	Indicates that AlertMailer has started receiving from a component that was previously offline. The priority is equal to the priority of the corresponding ComponentOffline message.
MailErr	Indicates that AlertMailer was unable to send a mail message. The message will include any error message returned by the system.

 Table 3
 AlertMailer internally generated alert messages

4 Definition of inifile sections and parameters

The AlertMailer configuration file—alertmailer.ini—consists of a number of *sections*, each containing several *parameters*; these are described below. In the alertmailer.ini file, sections are identified by a name enclosed in square brackets (e.g. [Log]). Each parameter is defined on a separate line following the section identifier, in the format: [SectionName]

```
ParameterName = Value
ParameterName = Value
```

All entries are required unless specified otherwise; some of the sections and parameters are optional. See "Inifile Example" on page 13 for an example, and for information on editing the file.

4.1 [Log]

The [Log] section defines the location, name, and verbosity of the AlertMailer log file, using the parameters described in Table 4.

Parameter	Definition
LogFilename = AlertMailer.log	The name of the AlertMailer log file.
LogDirectory = logs	The pathname for the directory in which to store the log file. • Do NOT include the trailing slash in this pathname.
Verbosity = Debug	The startup verbosity of the log file.Options are DEBUG, VERBOSE, or INFO; normal value is INFO

Table 4 [Log] section parameters

4.2 [AlertReceiver]

The [AlertReceiver] section defines the configuration for the module which receives alert messages via UDP from other programs, using the parameters described in Table 5.

Parameter	Definition
Port = 31000	 The UDP port used to receive alert messages from NaqsServer and other programs. Alert-generating programs and instruments must be config- ured to send alert messages to this port in order for Alert- Mailer to receive the messages.
MulticastGroup = 226.7.8.9	 Optional entry. The IP address of a multicast group on which the AlertMailer can receive messages. AlertMailer can listen to any number of multicast groups; include one line in the configuration file for each group. If no multicast groups are specified, AlertMailer will still receive alert messages sent via unicast UDP to the correct machine and port.

Table 5 [AlertReceiver] section parameters

4.3 [SystemMonitor]

The [SystemMonitor] section specifies the source name for alert-generating modules to be monitored (see also [AlertReceiver]). Network components that are not included in this section will not be monitored by AlertMailer. This section can contain any number of lines of the form described in Table 6.

Statement	Description
Component = sourceld	 Indicates that AlertMailer should monitor the component specified by sourceld, and issue a message when the component goes offline or comes back online. Permitted Values: Any string (may be an instrument ID or the sourceID of NaqsServer).

4.4 [MailSender]

The [MailSender] section defines the configuration of the email-sending module of the AlertMailer program, using the parameters described in Table 7.

Parameter	Definition
Sourceld = AlertMailer	The source name for messages which are generated internally by AlertMailer.
MailServer = mailhost.seismicnet.com	The host name or IP address of the SMTP mail server through which mail is to be sent.

Table 7 [MailSender] section parameters

Parameter	Definition
SenderAddress = naqs@seismicnet.com	 The email address to be used as the "from" address in email messages sent by AlertMAiler. This should normally be a valid email address with a mailbox accessible to the data acquisition systems administrator, since this address will receive notification about undeliverable messages.
ReplyToAddress = naqs@seismicnet.com	 The email address to be used as the "Reply To" address in email messages sent by AlertMailer. This may be the same as the SenderAddress, or it may be an address to which further instructions or queries are to be sent (typically, the email address of the data acquisition system administrator)
BccAddress = naqs@seismicnet.com	The email address to which a copy of every email message should be sent. This feature allows archiving of the alert mail activity in a central location. • To disable this feature, set the value to "none"
TranslationFile = naqsmail.properties	 The pathname of the file containing message-specific reformatting or translation information. Variable definitions and the format of translated messages are summarised in section 5, "Setting up a translation file". See also the naqsmail.properties file. To skip the translation step and forward messages using the default message format, set the value to "none".
SaveToFile = No	Optional entry. It indicates whether every mail message sent should be saved to disk. This is included as a troubleshooting feature, and should normally be set to "No" to avoid cluttering the disk with unnecessary files. This feature may also be toggled on and off from the command window. • Options are Yes, No; normal value is "No"
Format = Normal	 Optional entry. It specifies the formatting convention for the body of each email message. Options are Normal, Debug, and Brief: Normal: The email body will contain the message type, source, time, and severity, followed by the formatted message. For example:
	Subject: AlertMailer:AlertSystemUp - Nmx Alert system is now running Alert type: AlertSystemUp Source: AlertMailer Time: 2003-01-17 19:59:12.167 Priority: 1 Nmx Alert system is now running Components online: CAR044, SatTest Components offline: CAR332, CYG306, CYG326
	 Debug: The email body will contain the message type, source, time and severity, the format string and argument string, and the formatted message. Brief: The email body will contain only the formatted message.

 Table 7
 [MailSender] section parameters (Continued)

4.5 [Thresholds]

The [Thresholds] section is optional. It can be used in combination with the [MailGroups] and [Recipients] sections to provide additional message filtering, typically by setting a priority level for all messages of a given type. This section defines the minimum message priority (severity) to be forwarded for each message type. This section can contain any number of assignment statements of the form described in Table 8.

Message types are described in detail in the naqsmail.properties file; this file can be viewed with any text editor. Table 9 provides a summary of the priority levels for message types that have specific priority options.

Statement	Description
classID = value	 Optional entry. It specifies the minimum message priority to be forwarded for message type classId. Messages with priority < value will not be forwarded. Permitted values: Any non-negative integer.

Table 8 [Thresholds] section statement format

Table 9 Message priority options for setting thresholds

Source	Message type	Description	Priority options
AlertMailer	ComponentOffline	Indicates that AlertMailer is not receiving from the specified component	Number of minutes since last message was received
	ComponentOnline	Indicates that AlertMailer has started receiving from a component that was previously offline	(Same as priority of corresponding ComponentOffline message)
NaqsServer	RbfOpenFail	Issued if one or more ringbuffer cannot be opened properly	Number of ringbuffers which could not be opened
	RbfWriteFail	Issued when there is an error writing to a ringbuffer	Number of consecutive write failures (1, 100, 1000, 10000)
	RbfWriteOk	Issued when a ringbuffer write succeeds after previously being failed	Number of consecutive write failures before success
	NaqsEvent	Issued when the event-detection module detects a seismic event	The average over all triggers of the peak STA/LTA value recorded

Source	Message type	Description	Priority options
Each Carina	VSatTxOutage	Issued when Carina has not transmitted for N minutes	Number of minutes of outage (N = 2, 5, or 20)
	VSatTxOk	Issued when Carina transmission resumes	Same as priority of last VSatTxOutage message
	VSatSelfRxOutage	Issued when Carina has not received its own transmission for N minutes	Number of minutes of outage (N = 2, 5, or 20)
	VSatSelfRxOk	Issued when Carina self- reception resumes	(Same as priority of last VSatSelfRxOutage message)
	VSatRxOutage	Issued when Carina has not received from VSat <i>x</i> for N minutes	Number of minutes of outage (N = 2, 5, or 20)
	VSatRxOk	Issued when Carina self- reception resumes	(Same as priority of last VSatRxOutage message)
Each Cygnus, Janus, or Europa	PowerWarn	Issued when supply voltage enters RED zone (using thresholds from user interface)	1
	PowerOk	Issued when supply voltage enters GREEN zone	0
	SohWarn	Issued when SOH enters RED zone	1
	SohOk	Issued when SOH enters GREEN zone	0

 Table 9 Message priority options for setting thresholds (Continued)

A less typical application of thresholds is to provide a message-specific verbosity control through which low-importance messages can be suppressed. Setting the priority level of a message type to a value greater than the highest value provided in the range of options will suppress all messages of this type; these messages will not be forwarded to any mail group or recipient. For example, setting VsatRxOutage to 30 will suppress all VsatRxOutage messages. If a message type has a single priority option (for example, PowerWarn has a priority option of 1), setting the priority to a greater value (for example, setting PowerWarn to 2) will suppress all of messages of this type. Generally, this sort of filtering would more typically be managed through mail groups.

4.6 [MailGroups]

The [MailGroups] section is optional. It assigns message types to recipient groups. Every message of type classID will be forwarded to each member of the specified group.

Each message type can be sent to any number of groups, and any number of message types can be sent to any given group. This section can contain any number of assignment statements of the form described in Table 10.

Statement	Description
classID = groupID	Assigns message types to recipient groups, where classID is a specific message class type, and groupID is the name of a user-defined group to which that message should be sent. The email addresses of recipients belonging to each group are identified in the [Recipients] section.

Table 10 [MailGroups] section statements

4.7 [Recipients]

The [Recipients] section is optional. It defines recipients belonging to each mail group. It can contain any number of assignment statements of the types described in Table 11.

Table 11 [Recipients] section statements

Statement	Description
groupId = recipient	 Assigns recipients to a mail group (see also [MailGroups]). groupId is the name of a user-defined mail group, and recipient is the email address of a recipient belonging to that group. There should be a separate line for each recipient in each group. Each group may have any number of recipients, and each recipient may belong to any number of groups
classID = recipient	Indicates that all messages of type classId should be sent to the specified recipient. This provides a way to specify individual recipients for individual message types without defining recipient groups.
all = recipient	Indicates that all messages should be sent to the specified recipient, in addition to any other recipients specified for specific message types.
default = recipient	Specifies a destination for all messages with no other recipient.

5 Setting up a translation file

AlertMailer will optionally reformat messages before sending, using information from a user-defined translation file. The name of this file is specified in the alertmailer.ini file [MailSender] section; typically, it is the naqsmail.properties file. (To disable the translation/reformatting option, specify "none" for the file name.)

The translation file is similar in format to a standard Nanometrics inifile, and typically contains statements of the form "name = value", where name is a parameter name, and value is a string to be assigned to that parameter. The file can contain comments if they are preceded by the # character.



Note It is necessary to provide format strings only for messages which you want to reformat. Other messages will be sent using the default format.

For each message that is to be translated, the translation file must contain one or both of the following lines:

```
classID.subject = string1
classID.text = string2
```

where string1 is the format string defining the subject line and string2 is the format string defining the body text for an email message reporting an alert message of type classID. Long strings can be continued over several lines by placing a backslash character ($\langle \rangle$) at the end of each line to be continued. Leading blanks in continuation lines are ignored.

➤ To use the basic options provided in the naqsmail.properties file, "uncomment" the classID.text statement you want to use (delete the initial #), and customise the statement if you wish, using the method described in section 5.1.

5.1 Argument substitution

Each alert message type may contain a number of instance-specific arguments indicating, for example, a station name, earthquake magnitude, or other information. Alert-Mailer provides a simple but powerful formatting language which allows these arguments to be included anywhere, in any order, in the reformatted message. Arguments are indicated by special character sequences beginning with %, with defined tags as described in Table 12.

This method is very flexible, since it allows each argument to be referenced an arbitrary number of times in any order. This allows constructing of both complete and abbreviated messages, and accommodating different grammatical conventions which may be associated with different languages.

Tag [*]	Description
%1, %2,%9	These represent the corresponding element from the argument list;
%s	The sourceID of the module which generated the message
%m	The message type, or classID
%р	The message priority or severity
%a	The entire argument string, concatenated together, separated by a delimiting character (usually /)
%t	The time at which the message was generated
%r	A carriage return
%%	The % sign

Table 12 Argument tags (see the naqsmail.properties file for specific descriptions)

* Unrecognized or invalid tags will be displayed in the formatted message as "*".

5.2 Message translation example

Given a message type SohOk from source Cygnus 149, with argument string /VaultDoor/0.32, and the format string:

SohOk.text = Instrument %s SOH %1 reading OK (%2)

The resulting reformatted message is: Instrument CYG149 SOH VaultDoor reading OK (0.32) Appendix

Inifile Example

This section contains an example AlertMailer configuration file (inifile). On installation, the alertmailer.ini file is placed in the c:\nmx\user directory. You can edit the file in any text editor.

The inifile reader ignores white space and blank lines, so white space can be added anywhere within the file if desired to improve readability. Also, the inifile reader recognizes the double-slash "//" as a comment delimiter, so comments can be added anywhere in the file. Comments are useful for adding descriptive information to the file, or for temporarily removing parameters or sections from the file.

For example:

// This is a full line comment. A blank line has been inserted before the section name, and // extra spaces have been inserted between the section name and comment.

[SystemMonitor] Component = Na //Component = C	iqs-A YG001	 // a comment can follow a section header // a comment can follow a parameter definition // this parameter has been temporarily converted to a // comment, and can be reactivated by deleting the initial // 	
<pre>// AlertMailer.ini // Config file for</pre>	the A	AlertMailer program	
[Log] LogFilename = AlertMailer.log LogDirectory = logs Verbosity = Info			
[AlertReceiver] Port = 31000 MulticastGroup = 226.7.8.9 MulticastGroup = 226.9.8.7			
[SystemMonitor]			
Component = Naqs-A Component = CAR126	5		
[MailSender]			
SourceId	= Aler	tMailer	
MailServer	= mail	host.seismicnet.com	
SenderAddress	= naqs	@seismicnet.com	
ReplyToAddress = naqs@seismicnet.com		@seismicnet.com	

```
BccAddress
                 = naqs@seismicnet.com
TranslationFile = NaqsMail.properties
Format = Normal
SaveToFile = No
[ Thresholds ]
                   // forward only if priority >= 5
NaqsStatus = 5
[ MailGroups ]
NaqsEvent = seismic
                      // these messages go to seismic group
ComponentOffline = seismic
ComponentOffline = operations // these messages go to operations group
NaqsStatus = operations
[ Recipients ]
seismic = joe@seismicnet.com
                               // gets seismic group messages
seismic = quincy@quake.uwo.edu
operations = joe@seismicnet.com // gets operations group messages
operations = bill@othernet.ca
all = sysop@seismicnet.com
                               // gets all messages
```