



[MG-SOFT Corporation](http://www.mg-soft.com)

# Net Inspector

Version 6.6

## INSTALLATION AND CONFIGURATION GUIDE

(Document Version: 1.9.6)

Document published on Monday, 29-June-2009

Copyright © 2000-2009 MG-SOFT Corporation

In order to improve the design or performance characteristics, MG-SOFT reserves the right to make changes in this document or in the software without notice.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of MG-SOFT Corporation. Permission to print one copy is hereby granted if your only means of access is electronic.

Depending on your license, certain functions described in this document may not be available in the version of the software that you are currently using.

Screenshots used in this document may slightly differ from those on your display.

MG-SOFT may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. The furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Copyright © 2000-2009 MG-SOFT Corporation. All rights reserved.

---

## TABLE OF CONTENTS

---

<b>1</b>	<b>Introduction.....</b>	<b>4</b>
<b>2</b>	<b>About Net Inspector .....</b>	<b>5</b>
<b>3</b>	<b>Installing Net Inspector on Windows.....</b>	<b>6</b>
3.1	Requirements .....	6
3.2	Installing Net Inspector for Windows .....	6
<b>4</b>	<b>Installing Net Inspector on Linux.....</b>	<b>15</b>
4.1	Requirements .....	15
4.2	Installing Net Inspector .....	16
4.2.1	<i>Installing Net Inspector on Standalone Computer.....</i>	<i>16</i>
	<i>Fresh Installation.....</i>	<i>16</i>
	<i>Updating Existing Installation .....</i>	<i>17</i>
4.2.2	<i>Installing Net Inspector on Cluster.....</i>	<i>18</i>
4.3	Starting and Stopping Net Inspector Server from Command Prompt.....	19
<b>5</b>	<b>Net Inspector Server Initialization File.....</b>	<b>20</b>
5.1	Section [connection] .....	20
5.2	Section [user].....	21
5.3	Section [config] .....	22
5.4	Section [action] .....	23
5.4.1	<i>Defining Actions.....</i>	<i>23</i>
5.5	Section [event].....	28
5.6	Section [log].....	29
5.7	Section [snmp notifications].....	31
5.8	Section [snmp agent].....	33
<b>6</b>	<b>Net Inspector Server Profiles File .....</b>	<b>34</b>
6.1	Section [poll profile] .....	34
6.2	Section [snmp access profile].....	36
<b>7</b>	<b>Configuring SNMP Notification Destination on SNMP Agents .....</b>	<b>38</b>

---

## 1 INTRODUCTION

---

This guide provides instructions for installing and configuring Net Inspector Server version 6.x for Windows and Linux operating systems.

All command line commands, filenames, paths and examples in this guide are formatted with a fixed width font, e.g., `mkdir /install_niv6`.

The path to Net Inspector installation directory in this guide is specified as `//Engine`. By default, this is equivalent to `C:\Program Files\MG-SOFT\Net Inspector` on Windows and to `/usr/local/mg-soft/netinspector` path on Linux operating system. Example: By default, the `//Engine/workspace` path means the following path: `C:\Program Files\MG-SOFT\Net Inspector\workspace` (Windows) and `/usr/local/mg-soft/netinspector/workspace` (Linux).

The content of this guide is listed in the [Table of Contents](#).

## 2 ABOUT NET INSPECTOR

---

Net Inspector is a fault management application used for monitoring the status of managed objects and viewing and managing alarms on managed objects, i.e., telecommunications and IT devices.

Net Inspector is a client/server application. Net Inspector Server (Engine) runs as a background application without a user interface. It continuously polls managed objects in the supervised network by means of SNMP and ICMP queries and receives SNMP Trap and Inform notifications sent autonomously by managed objects when important events occur. Net Inspector Server translates received SNMP notifications into alarms, stores alarms in a database and dispatches information about alarms to connected Net Inspector Clients. Net Inspector Client, on the other hand, is a Java-based application that connects to the Server and provides a graphical user interface that lets you view and manage alarms on managed objects, monitor the status of managed objects, as well as configure the Client and certain parameters of the Server application.

Net Inspector Server is available for MS Windows operating systems (Windows 2000, Windows XP, Windows Server 2003, Windows Vista and Windows Server 2008) as well as for Red Hat Linux Enterprise operating systems (RHEL 3, 4 and 5 for x86 and x86\_64 architectures), while the Client runs on all operating systems with the Java Runtime Environment (JRE), version 5.0 (a.k.a. 1.5) or later. Currently supported database management systems are MS SQL Server (on Windows) and MySQL (on Linux).

## 3 INSTALLING NET INSPECTOR ON WINDOWS

### 3.1 Requirements

In order to install and use MG-SOFT Net Inspector for Windows version 6.x, the following software needs to be installed on your computer:

- ❑ Windows 2000, Windows XP, Windows Server 2003, Windows Vista or Windows Server 2008
- ❑ Java Runtime Environment (JRE) for Windows, version 5.0 (a.k.a. 1.5) or later, which can be downloaded from the following Web page:  
<http://www.java.com/en/download/manual.jsp>

Additionally, you need to have administrative privileges to successfully install Net Inspector.

### 3.2 Installing Net Inspector for Windows

1. Start the computer and put the MG-SOFT Net Inspector installation CD into your CD or DVD drive.

**Note:** You need to have the administrative user privileges to install the software.

2. Click the Windows taskbar **Start** button and select the **Run** command.
3. The Run dialog box appears (Figure 1).

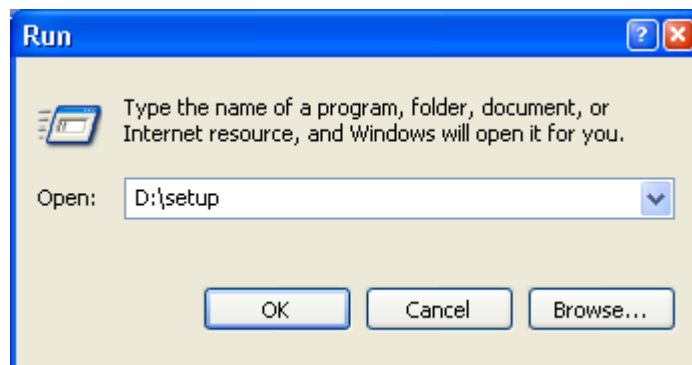


Figure 1: Run dialog box

4. Into the **Open** input line, type `D:\setup` and click the **OK** button.

**Note:** `D` is the letter assigned to the CD or DVD drive. If your CD or DVD has a different letter, type that one instead of `D`.

5. Net Inspector setup routine first checks your system for the presence of Microsoft SQL Server software, which is required for managing the Net Inspector database. If no matching software can be found, installer displays the dialog box offering you the option to install Microsoft SQL Server 2000 Desktop Engine (MSDE 2000), SP3a (Figure 2).

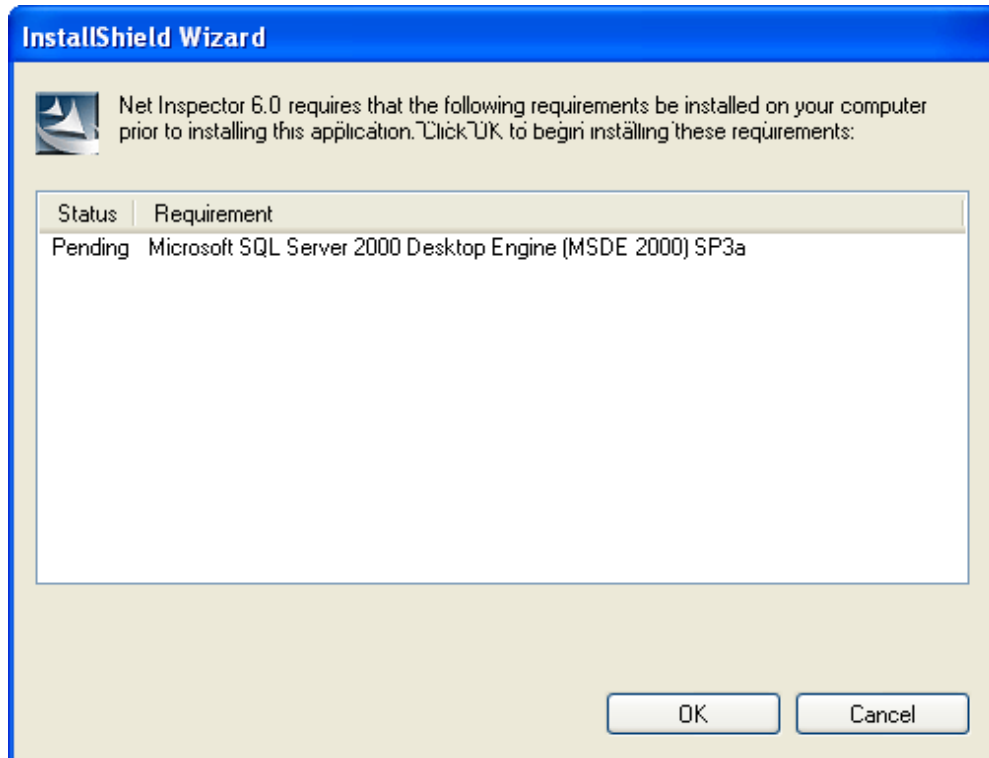


Figure 2: MSDE 2000 installation screen

6. Click the **OK** button to install the MSDE 2000. After successful installation of MSDE 2000, the Net Inspector Installation Wizard welcome screen appears (Figure 3).

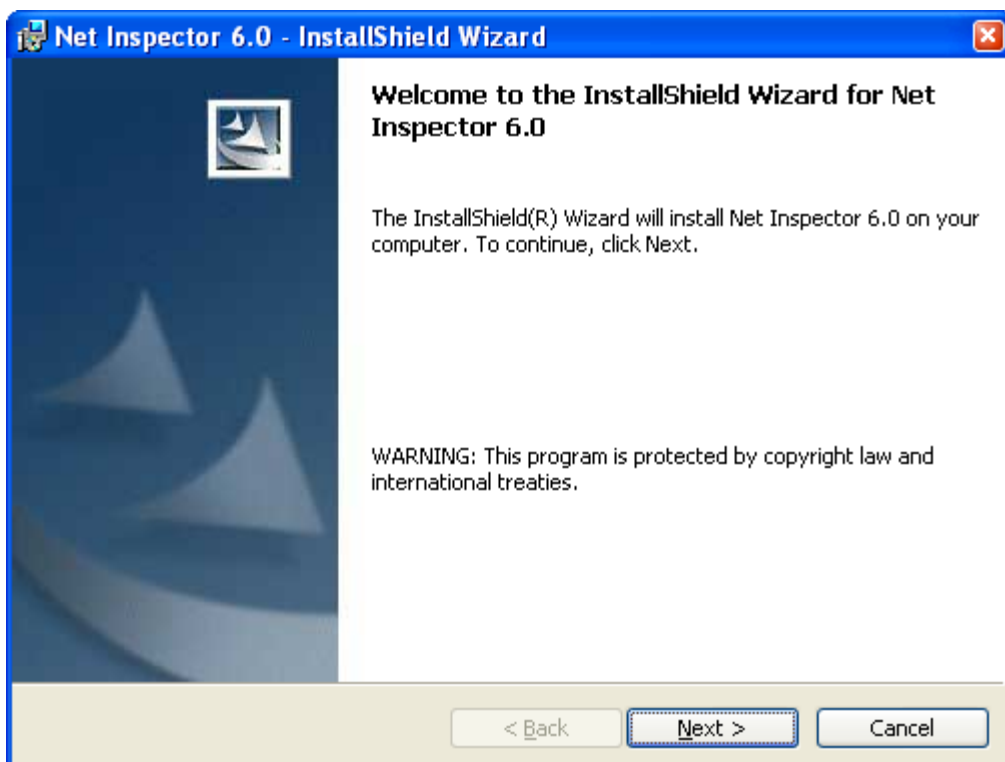


Figure 3: Net Inspector Installation Wizard – Screen 1

7. To proceed with the installation, use the **Next** button at the bottom of the wizard screen.

**Note:** This manual describes only those installation steps that are specific to the MG-SOFT Net Inspector installation process.

8. After passing the standard steps of accepting the license agreement, specifying the license key file location and providing the user information, the Net Inspector Server Settings screen appears (Figure 4).

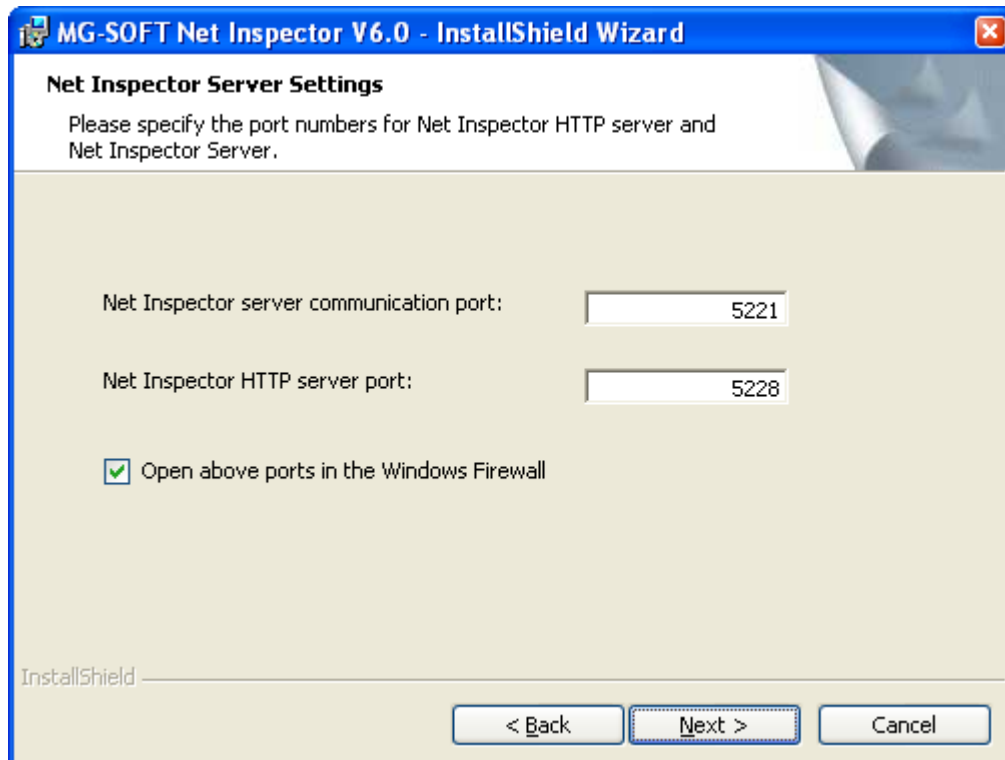


Figure 4: Net Inspector Installation Wizard – Screen 5

- ❑ Into the **Net Inspector Server communication port** input line, enter the number of TCP port on which Net Inspector Server will listen to for incoming Net Inspector Client connections. By default, this port number is 5221.
- ❑ Into the **Net Inspector HTTP Server port** input line, enter the number of TCP port on which Net Inspector HTTP Server will listen to for incoming HTTP connections. By default, this port number is 5228.

**Note:** Net Inspector comes with its own HTTP (Web) server program that installs to the same computer as Net Inspector Server and other components of the package. Net Inspector HTTP Server serves a Web page that enables launching Net Inspector Client by using the Java Web Start framework. The latter enables starting Java applications from anywhere in the network by using a Web browser. Additionally, Net Inspector HTTP Server provides also Web-based access to Net Inspector documentation in electronic form.

- ❑ Leave the **Open above ports in Windows Firewall** checkbox checked if you want the Installation Wizard to open above specified TCP ports in the built-in Windows Firewall (if applicable). If you disable this option, remote users will not be able to connect to Net Inspector Server and Net Inspector HTTP Server.

**Note:** Net Inspector Installation Wizard will open the relevant ports only in the built-in Windows Firewall. If you use third-party firewall software, you need to manually configure it to allow incoming connections on relevant ports. For details, please check the documentation that came with the third party firewall software.

9. Click the **Next** button to proceed to the next step.
10. The Discovery Settings screen appears (Figure 5), where you can configure parameters for the network discovery operation, which is a Net Inspector feature that discovers devices on your network and automatically adds them to Net Inspector configuration, so the software can immediately start monitoring your network.

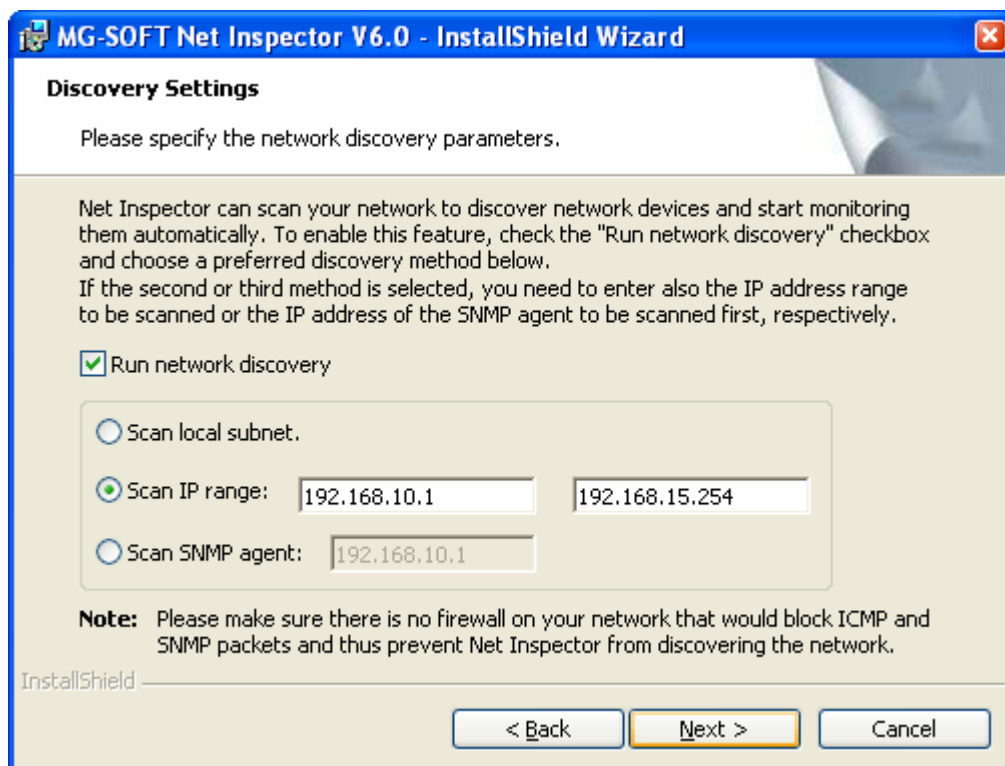


Figure 5: Net Inspector Installation Wizard – Screen 6

If the **Run network discovery** checkbox is checked, Net Inspector Server will automatically run the network discovery operation immediately after the installation and start monitoring discovered devices. If you disable this option, you can run the discovery operation later or add devices to Net Inspector configuration manually. If the network discovery operation is enabled, you can select the desired discovery strategy, as follows:

- ❑ If the **Scan local subnet** radio button is selected (this is the default option), Net Inspector discovers the network (devices and their interconnections) within the local subnet, that is, subnet the PC running Net Inspector Server is a member of.

- ❑ If the **Scan IP range** radio button is selected, Net Inspector performs the discovery operation within the range of IP addresses specified by the user. In this case, enter the IP range start and end address into the accompanying input lines.
  - ❑ If the **Scan SNMP agent** radio button is selected, Net Inspector discovers the network by means of the SNMP-based network scan that progressively examines the routing tables and other relevant data on the scanned objects in your network. If this option is selected, enter the IP address of the SNMP-enabled device that will be scanned first into the accompanying input line.
11. After you have specified the network discovery options, click the **Next** button to proceed to the next screen.

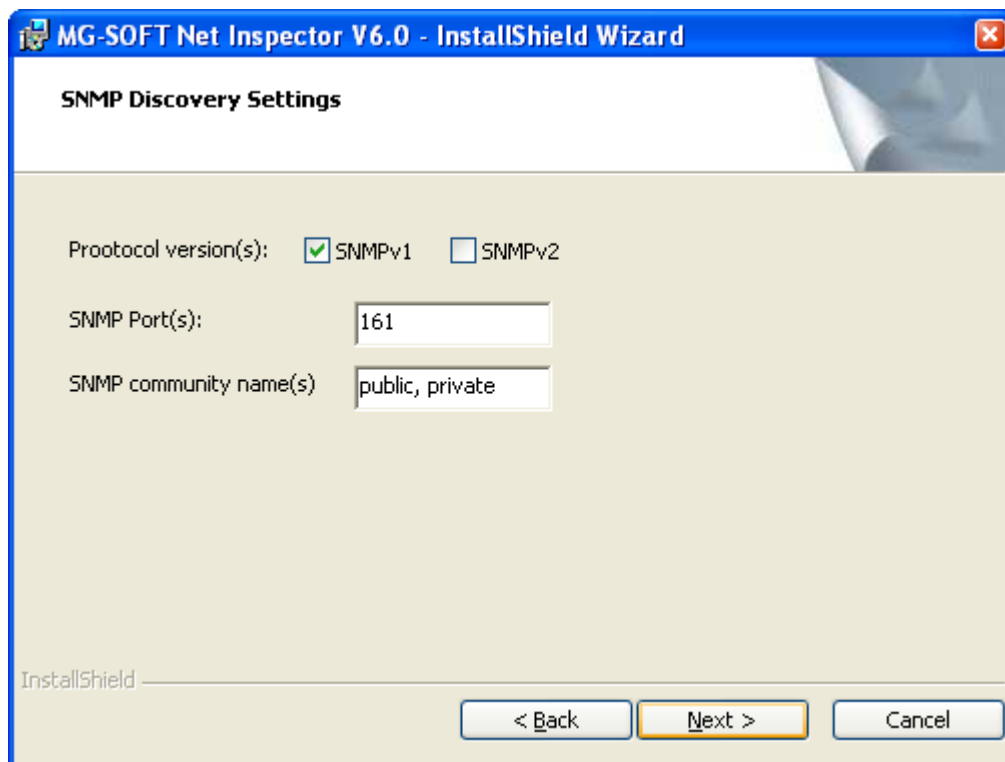


Figure 6: Net Inspector Installation Wizard – Screen 7

12. The SNMP Discovery Settings screen appears (Figure 7), where you can specify the SNMP access parameters for discovering the network:
- ❑ Select the **Protocol version(s)** that the SNMP devices on your network support. Net Inspector Discovery module will search only for the devices that support the selected SNMP protocol version(s). If you specify more than one SNMP version, Net Inspector will search for devices that support any of the specified protocols (starting with the lower SNMP protocol version).
  - ❑ In the **SNMP port(s)** input line, specify the port on which network devices listen to for incoming for SNMP requests. You can specify more than one port number by separating individual items with comma (,).

- ❑ In the **SNMP community name(s)** input line, enter one or more (read) community names accepted by SNMP-enabled network devices. You can specify more than one community name by separating individual items with comma (,).

13. Click the **Next** button to proceed to the next step.

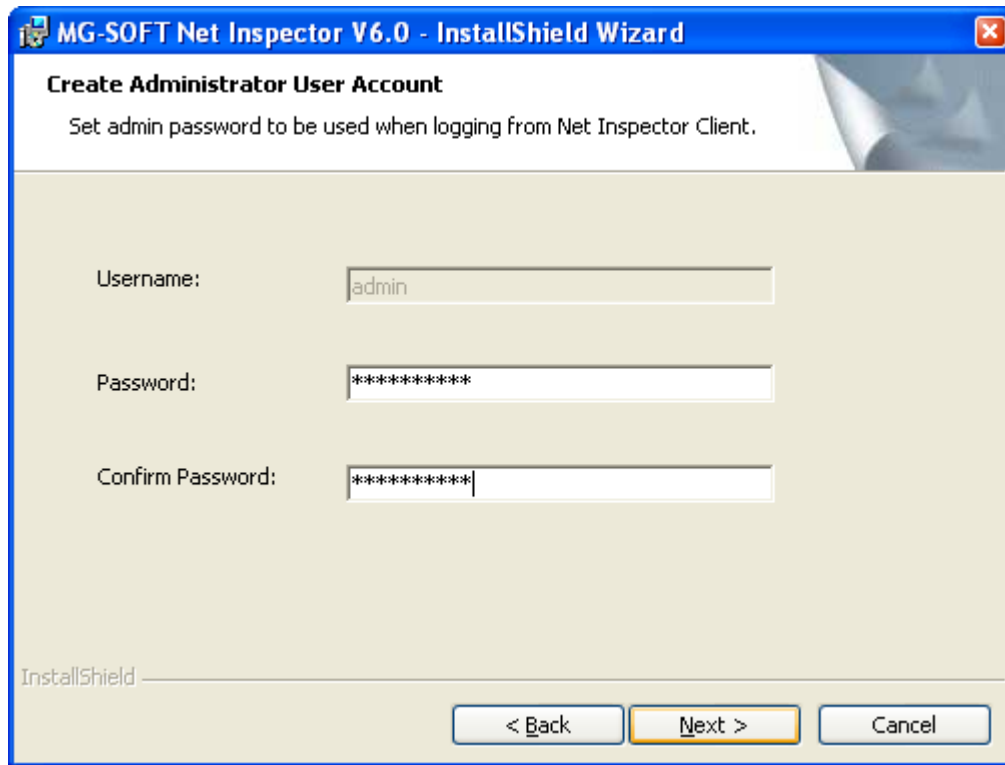


Figure 7: Net Inspector Installation Wizard – Screen 8

14. The Create Administrator User Account screen appears (Figure 7), which prompts you to enter a password for the built-in Net Inspector administrative user account:
- ❑ The **Username** field displays the username of the built-in administrative user account (admin). The username is displayed read-only and cannot be modified.
  - ❑ Enter the password into the **Password** input line and confirm the password by re-entering it into the **Confirm password** input line below.

**Note:** Carefully note the username and password (both are case sensitive!). After installing the software, you will need to log on to Net Inspector Server using this user account in order to create other user accounts and perform other administrative tasks.

15. Click the **Next** button to proceed to the next step. After specifying the program installation path in the Destination Folder screen, click the **Next** button to proceed to the SQL Server Setup screen.

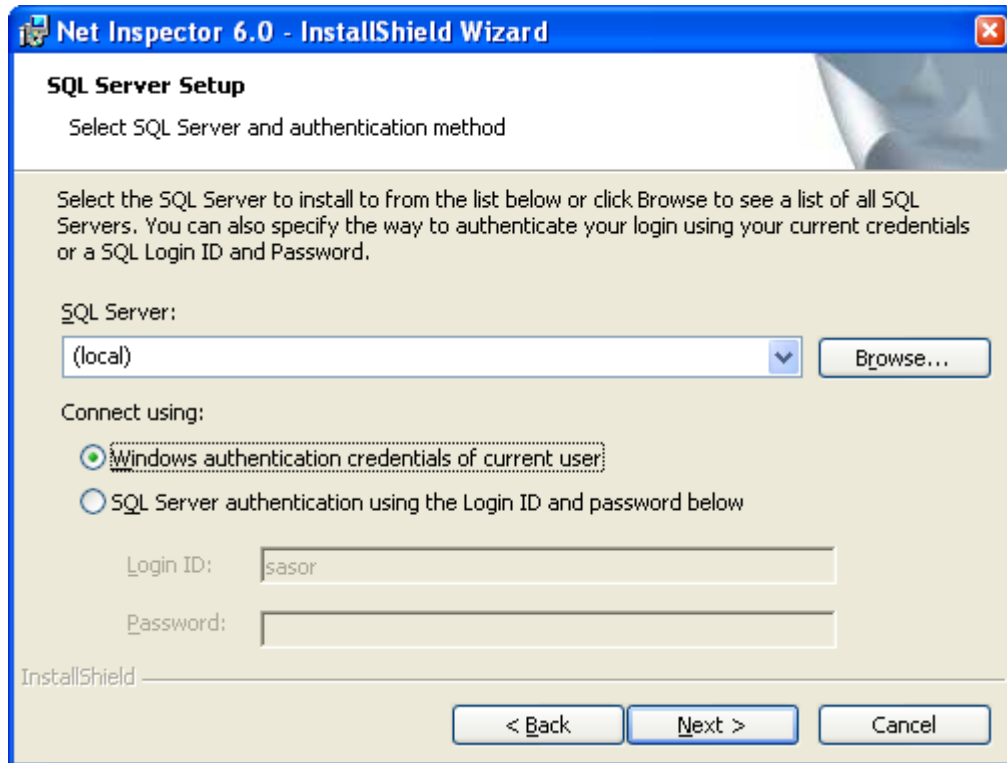


Figure 8: Net Inspector Installation Wizard – Screen 9

16. The SQL Server Setup screen appears (Figure 8).
  - ❑ In the **SQL Server** drop-down list, specify the MS SQL Server you want to use with Net Inspector.
    - ❑ If you want to use a remote SQL Server, enter its name into the **SQL Server** drop-down list. If unsure regarding the name, click the **Browse** button next to this input line to open a dialog box listing the names of all SQL Servers on your network and select the desired one from the list.
    - ❑ To use the SQL Server running on your computer, leave the default value (local) in the **SQL Server** drop-down list.
  - ❑ The **Connect using** options let you specify what authentication mechanism you wish to use to connect to the selected SQL Server (this setting must match the authentication setting on the SQL Server).
    - ❑ To connect to SQL Server using the Windows authentication mechanism and credentials of the currently logged-on user, leave the first radio button selected.
    - ❑ To connect to SQL Server using the SQL Server authentication mechanism, select the second radio button and enter the **Login ID** and the **Password** of the 'sa' user into the corresponding input lines.
17. After you have specified the SQL server options, click the **Next** button to proceed to the next screen (Figure 9).

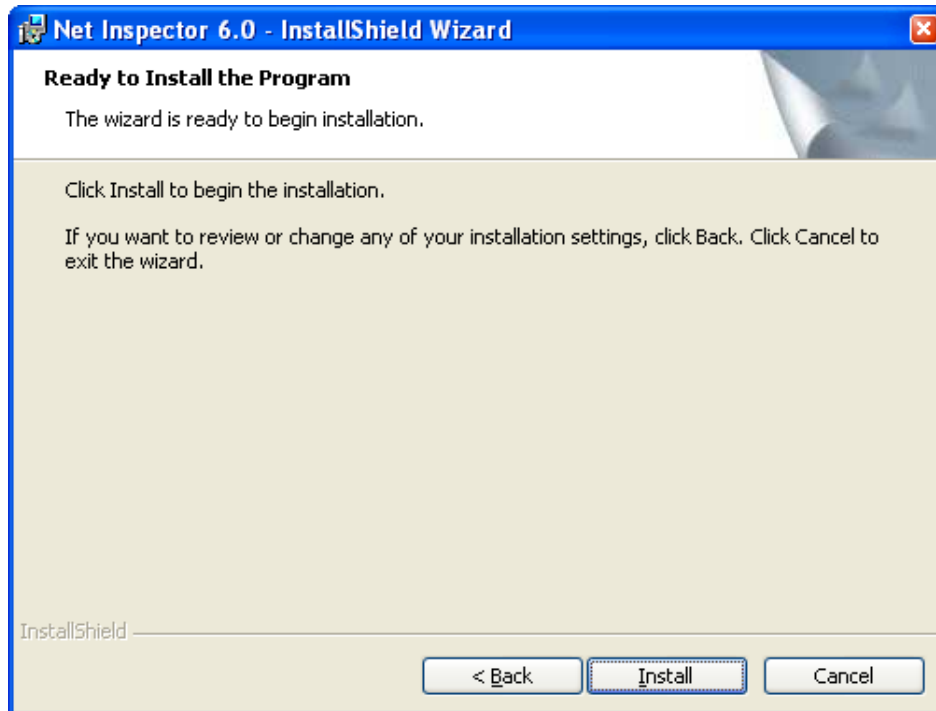


Figure 9: Net Inspector Installation Wizard – Screen 10

18. Click the **Install** button to install the software according to the parameters specified in the previous steps. After copying the required files and setting up necessary registry entries, the final screen of the Net Inspector Installation Wizard appears (Figure 10).

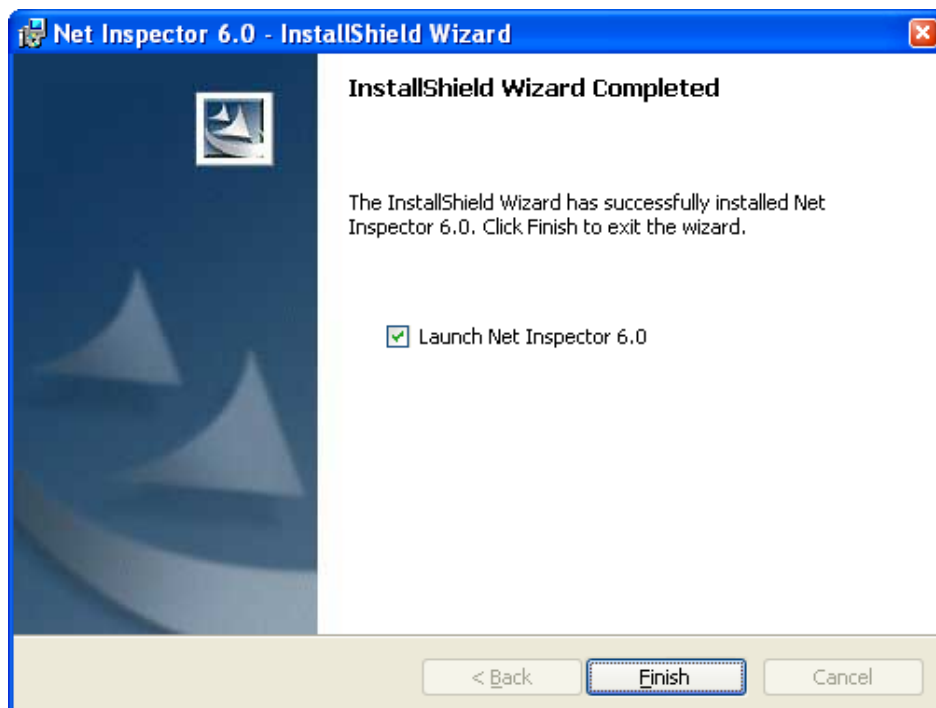


Figure 10: Net Inspector Installation Wizard – Screen 11

19. Click the **Finish** button to end the Net Inspector Installation Wizard. If you leave the **Launch Net Inspector Client** checkbox checked, this operation also displays the Net Inspector Web Start page in your default Web browser, which lets you launch Net Inspector Client (by clicking the **Net Inspector Client** link) and connect it to the Server (using the **File /Connect** command).

---

**Note 1:** Please refer the Net Inspector Client Reference Manual and Net Inspector Client Getting Started Guide for more information on configuring and using Net Inspector.

---

---

**Note 2:** If you received a valid license key file after you had installed the software, you need to copy the `license.key` file to the Net Inspector's Bin folder (i.e., `C:\Program Files\MG-SOFT\Net Inspector\Bin`) and restart the MG-SOFT Net Inspector Server service in the Services window (**Start / Settings / Control Panel / Administrative Tools / Services**).

---

---

## 4 INSTALLING NET INSPECTOR ON LINUX

---

### 4.1 Requirements

---

In order to install MG-SOFT Net Inspector for Linux version 6.x, the following software needs to be installed on your computer:

- ❑ Red Hat Enterprise Linux (RHEL 3, 4 or 5) for x86 or x86\_64 architecture
- ❑ Compatibility standard C++ libraries (compat-libstdc++-33)
- ❑ unixODBC version 2.2.9 or later, available at: [www.unixodbc.org](http://www.unixodbc.org)
- ❑ MySQL Connector / ODBC driver version 3.51 or later, available at: <http://dev.mysql.com/downloads/connector/odbc/>

Additional software requirements:

- ❑ MySQL database version 4.1 or later (installed locally or remotely on the network). MySQL database is available at: <http://www.mysql.com/>
- ❑ Java Runtime Environment (JRE), version 5.0 (a.k.a. 1.5) or later must be installed on all computers that will run Net Inspector Client. JRE for various operating systems can be downloaded from the following Web page: <http://www.java.com/en/download/manual.jsp>

Furthermore, you need to have administrative (root) privileges on the target PC and MySQL database administrator privileges on the relevant MySQL server to successfully install Net Inspector.

The procedure of installing Net Inspector on a Linux operating system includes several steps, as described in this section.

Throughout this guide it is presumed that the contents of the Net Inspector installation CD is accessible at `/mnt/cdrom`.

The entire installation procedure is carried out from a temporary directory. To create the temporary directory, execute the following command at a command prompt:

```
mkdir /install_niv6
```

Then change to this directory, with the following command:

```
cd /install_niv6
```

Once the installation is successfully completed, remove the temporary directory with the following command:

```
rm -Rf /install_niv6
```

---

## 4.2 Installing Net Inspector

---

### 4.2.1 Installing Net Inspector on Standalone Computer

---

#### Fresh Installation

---

Before installing Net Inspector, make sure to install the required software, as specified in the [Requirements](#) section. For detailed installation instructions, please refer to the “Installing the Software” section of the `READ_ME.TXT` file bundled with the installer.

#### Installing the RPM package

If no previous Net Inspector installation exists on the target machine, install Net Inspector from the installation CD by running the following command:

```
rpm -ihv /mnt/cdrom/<name of the Net Inspector RPM package>
```

This will install Net Inspector binaries on your computer and create Net Inspector menu entries. The installer also searches the local drives for PDF file viewer software (required for viewing Net Inspector documentation in electronic form) and Java Runtime Environment (required to run Net Inspector Client locally).

#### Post-installation procedure

After successfully installing the RPM package, run the following command to perform the post-installation configuration required for using Net Inspector:

```
nipostinst
```

This runs a post-installation script that performs the following:

- ❑ Creates Net Inspector database and user account for accessing this database on the local or remote MySQL server
- ❑ Registers ODBC DSN (data source name) for accessing the Net Inspector database
- ❑ Sets the Net Inspector administrator password (saved in the `//Engine/workspace/userconfig.txt` file)
- ❑ Sets the TCP port on which the Net Inspector Server listens to for incoming client connections (default is 5221) and the port on which Net Inspector HTTP Server listens to for incoming HTTP connections (default is 5228)
- ❑ Sets the hostname, fully qualified domain name (FQDN) or IP address of the computer serving the Java Web Start web page used for launching Net Inspector Client remotely
- ❑ Copies a valid `license.key` file to `//Engine/bin` directory
- ❑ Starts MG-SOFT SNMP Trap daemon (`mgtrapd`)
- ❑ Starts MG-SOFT Net Inspector Server (`niengine`) and HTTP daemon (`mghttpd`)

---

**Note:** Make sure that TCP ports, which Net Inspector Server and HTTP Server listen to, are open in the firewall (if applicable).

---

After successfully finishing the post-installation procedure, you can run Net Inspector Client from the start menu and connect it to Net Inspector Server. For detailed instructions, please refer to Net Inspector Client Getting Started Guide.

Once the entire installation process is completed, you can delete the temporary directory from which you installed the software by issuing the following commands:

```
cd ..  
rm -Rf /install_niv6
```

## Updating Existing Installation

---

If you want to update an existing Net Inspector installation, first stop Net Inspector Server (niengine) and MG-SOFT SNMP Trap daemon (mgtrapd), as follows:

```
/etc/init.d/niengine stop  
/etc/init.d/mgtrapd stop
```

Then, remove the existing installation by running the following command:

```
rpm -e netinspector
```

---

**Note:** Net Inspector workspace (user views, devices) and all the settings will be preserved.

---

Next, run the following command to install the new version of Net Inspector:

```
rpm -ihv /mnt/cdrom/<name of the Net Inspector RPM package>
```

Finally, run the following command to make post-installation configuration settings and run the MG-SOFT SNMP Trap daemon (mgtrapd) and Net Inspector Server (niengine) services:

```
nipostinst
```

Once the update process is completed, remove the temporary directory from which you performed the installation by issuing the following commands:

```
cd ..  
rm -Rf /install_niv6
```

## 4.2.2 Installing Net Inspector on Cluster

---

To install or update Net Inspector on a cluster system, run the [install](#) or the [update](#) commands described in previous sections on both (all) cluster nodes.

After installing or updating Net Inspector, you need to run the following script on both (all) clusters nodes in order to set up the Net Inspector workspace files for use in the cluster:

```
netinspector-cluster-setup
```

The above script will prompt you to enter the name of the shared storage, which contains the Net Inspector workspace files, e.g., `/dev/sdc1` and its mount point, e.g., `/usr/local`.

### Required Cluster Software Settings

You need to make the following settings in your cluster management software in order for it to properly “fail over” Net Inspector Server:

1. Configure the cluster software so that it will mount the shared storage to the directory that has been specified above as the mount point, e.g., `/usr/local`
2. Configure the cluster software so that it will run the Net Inspector Server initialization script at failover event, i.e.: `/etc/init.d/niengine`

For detailed instructions on configuring the above settings, please refer to your cluster software documentation.

---

**Note:** When using Net Inspector in a clustered environment, you must start and stop Net Inspector Server from the cluster management software.

---

## 4.3 Starting and Stopping Net Inspector Server from Command Prompt

---

Before starting Net Inspector Server, you should configure parameters in the [Net Inspector Server initialization file](#) and the [Net Inspector Server profiles file](#).

You need the “root” privileges to successfully start and stop Net Inspector Server from the command prompt.

- To start the Net Inspector Server (niengine), run the following command from the command prompt:

```
mg-netinspector-start.sh startengine
```

or

```
/etc/init.d/niengine start
```

- To stop the Net Inspector Server (niengine), run the following command from the command prompt:

```
mg-netinspector-start.sh stopengine
```

or

```
/etc/init.d/niengine stop
```

---

**Note:** If using Net Inspector in a clustered environment, you must start and stop Net Inspector Server from the cluster management software.

---

## 5 NET INSPECTOR SERVER INITIALIZATION FILE

---

Net Inspector Server initialization parameters are specified in the `niengine.ini` file. This initialization file should be located in the `//Engine/workspace` directory. When Net Inspector Server starts up, it reads the initialization parameters from the `niengine.ini` file, and initializes itself accordingly. If the `niengine.ini` file is not present in the `//Engine/workspace` directory, the default initialization parameters are used.

The Net Inspector Server initialization file (`niengine.ini`) is a plain ASCII file that can be edited in any text editor. It has the following format:

```
[section1]
; optional comment
parameter1 = value1
parameter2 = value2
parameter3 = value3

[section2]
; optional comment
parameter1 = value1
parameter2 = value2

...
```

The initialization file contains several sections. Every section contains one or more parameter. Supported sections and corresponding parameters are described below.

### 5.1 Section [connection]

---

The `[connection]` section contains the connection parameters.

The `port` parameter value determines the TCP port number on which Net Inspector Server listens to for Client connections. The default value of this parameter is 5221. The `timeout` parameter specifies how long (in seconds) the Client will wait for the Server response (or vice-versa), before generating the timeout signal. The `retries` parameter specifies how many times the program re-sends the connection request after the first timeout.

The `extension_port` parameter value specifies the TCP port number on which Net Inspector Server listens to for incoming connections initiated by its program extensions (e.g., Net Inspector mgmail extension, etc.).

The `soap_port` parameter value specifies the TCP port number on which Net Inspector Server listens to for incoming connections initiated by Net Inspector Configuration Browser application. The default value of this parameter is 8080.

Example:

```
[connection]
port = 5221
timeout = 10
retries = 3
extension_port = 5223
soap_port = 8081
```

## 5.2 Section [user]

---

The [user] section specifies where the Net Inspector users are defined.

The type parameter determines the source of the user data. Available values are:

- txt – the data about users is in a text file.

The dsn parameter specifies the location of the user data.

The default username is admin with the password admin.

Example:

```
[user]
; available type is: txt.
type = txt
dsn = //Engine/workspace/user_config.txt
```

Example of a text file defining users:

```
[user]
username=admin
passwd=admin
access=admin

[user1]
username=operator
passwd=operator
access=operator

[user2]
username=guest
passwd=guest
access=guest
```

## 5.3 Section [config]

**Note:** To be able to effectively monitor alarms on managed objects, you need to configure the SNMP agents on managed devices to send SNMP notifications to Net Inspector Server. Otherwise, Net Inspector will not receive those notifications and consequently will not display and notify you about the corresponding alarms.

The [config] section provides information about the Net Inspector configuration database, which contains data about the managed objects. The `niengine.ini` file can contain more than one configuration section. In this case, sections must be named according to the following scheme: [config], [config1], [config2], ...[configN], where  $N \leq 63$ .

The value of the `type` parameter specifies the type of the configuration database. Valid values are:

- `txt` – the configuration information is stored in a text file,

The `dsn` parameter specifies the location of the configuration database. The `user` and `passwd` parameters are used for specifying the username and password for accessing the database (if required).

The `passive_resync` parameter is used for enabling or disabling the passive resynchronization of event records maintained by Net Inspector Server and the SNMP agents on managed objects. The default value of this parameter is `false`. If the passive resynchronization is enabled, the resynchronization will occur only if initiated by the user and immediately after the managed object starts responding to Net Inspector Server queries; while there will be no resynchronization in case the SNMP notifications are lost.

The `change_gateway` parameter controls if Net Inspector Server should change the gateway for accessing duplicated objects when their state turns from active to passive. The default value of this parameter is `true`.

Examples of different configurations:

```
[config]
type = txt
dsn = //Engine/workspace/simpleconfig1.txt
passive_resync = true
change_gateway = true
```

```
[config1]
type = txt
dsn = //Engine/workspace/simpleconfig2.txt
passive_resync = false
change_gateway = false
```

Example of configuration information specified in a text file:

```
[device1]
hostname = local
ipaddr = 127.0.0.1
type = IP

[device2]
hostname = test
ipaddr = 192.168.69.10
type = IP
```

The `type` parameter above specifies the type of the managed device. The valid value is:

- IP

## 5.4 Section [action]

---

The `[action]` section specifies where the Net Inspector actions are configured.

The `type` parameter determines the source of the actions information. Available values are:

- `txt` – the actions information is in a text file.

The `dsn` parameter specifies the location of the actions information.

Example:

```
[action]
; available type is: txt.
type = txt
dsn = //Engine/workspace/actionconfig.txt
```

### 5.4.1 Defining Actions

---

A text file that defines Net Inspector actions (e.g., `actionconfig.txt`) should be present in the `//Engine/workspace` directory.

An action is defined with the `[actionX]` section. If more than one action is defined, the action sections should be named according to the following scheme: `[action1]`, `[action2]`, ..., `[actionN]`.

The action section contains the following parameters:

The `mstate` parameter controls if the action functionality (e.g., e-mail sending) is enabled or not. Valid values are 1 (enabled) and 0 (disabled).

The `name` parameter specifies the name of the action (action object).

The `type` parameter above specifies the type of the action (action object). Valid values are:

- `CMD` - command object
- `SMS` - SMS object
- `MAIL` - mail object

The `desc` parameter is used for describing the action (action object).

The `filter_name` parameter specifies the name of the action filter applied to the action object.

The remaining parameters depend on type of the action, as follows:

#### **CMD:**

---

<code>cmd_line</code>	The system command or the path to the program or script to be executed on events.
<code>cmd_args</code>	The switches and parameters to be appended to the command line specified above. It supports using the <a href="#">reserved words</a> .
<code>auto_term_enable</code>	Controls if Net Inspector will check if started processes still run and automatically terminate such processes. Valid values are <code>true</code> and <code>false</code> .
<code>auto_term_timeout</code>	Controls how long (in seconds) Net Inspector waits before it checks if started processes still run. This setting is effective only if the <code>auto_term_enable</code> parameter value is set to <code>true</code> .

#### **SMS:**

---

<code>port</code>	The serial port to which the mobile phone for sending SMS messages is connected.
<code>speed</code>	The baud rate (speed in bits per second) for communication with the mobile phone.
<code>data_bits</code>	The number of data bits used for each character that is transmitted and received.

---

parity	The communication parity setting..
stop_bits	The communication stop bits setting.
phone	The phone number of the SMS message recipient. The phone number must include the international country code, the area code or mobile network code (without the leading zero), and the actual mobile phone number. Do <b>not</b> prefix the number with the international direct-dial prefix (which is 00 in most countries (011 in North America) and sometimes substituted with the plus (+) sign).
msg	Specifies the contents of the SMS messages. You can specify the contents of the SMS message by combining regular text with <a href="#">reserved words</a> .

**MAIL:**

---

mail_server_addr	The name or IP address of the SMTP mail server used for sending e-mails.
mail_server_port	The TCP port on which the SMTP server listens to for incoming connections.
mail_server_timeout	The timeout value (in seconds) for connecting to the SMTP server.
user_name	The name of the e-mail sender (e.g., the name of the e-mail account holder).
user_org	The sender's organization name.
user_email	The sender's e-mail address.
user_reply_to	The "reply to" e-mail address.
send_to	The e-mail address of the recipient.
msg_custom_subject	Specifies the contents of the e-mail subject section by combining arbitrary text and <a href="#">reserved words</a> , which let you include desired information about events into the e-mail subject.
msg_body_file	The file in which the contents of the e-mail body section is specified (e.g., <code>//Engine/workspace/action3_msg_body.txt</code> )
msg_as_html	Enables or disables sending e-mails in HTML format. Valid values are <code>true</code> and <code>false</code> .

<code>msg_merge_max</code>	Specifies up to how many events triggered within a particular time frame (defined with the <code>msg_merge_interval</code> ) will be included into one e-mail message.
<code>msg_merge_interval</code>	Specifies the time frame for merging e-mails.

Example of the contents of the message body section definition file (e.g., `//Engine/workspace/action3_msg_body.txt`):

```
$FOR_EACH_BEGIN
  '$SEVERITY' alarm from '$SOURCE_NAME' at '$DATE_TIME'
  ('$MESSAGE', cause: '$CAUSE')
$FOR_EACH_END
```

### **Reserved Words**

a) The following reserved words are available for all events:

<code>\$\$SEVERITY</code>	Event severity level (e.g., critical, major,...)
<code>\$\$SEVERITY_ID</code>	Event severity ID number (2=normal, 4=informational, 8=warning, 16=minor, 32=major, 64=critical)
<code>\$\$SOURCE_ID</code>	ID number of event source object (e.g., 65595)
<code>\$\$SOURCE_NAME</code>	Name of event source object (e.g., MyServer)
<code>\$\$SOURCE_INFO</code>	Additional information about the source of event (e.g., Physical Memory)
<code>\$\$SOURCE_TYPE</code>	Type of source object (e.g., IP)
<code>\$\$MESSAGE</code>	Event message (e.g., Threshold value for storage usage exceeded)
<code>\$\$MESSAGE_ID</code>	Event message ID number (e.g., 11007)
<code>\$\$CAUSE</code>	Event cause (e.g., Threshold Crossed)
<code>\$\$CAUSE_ID</code>	Event cause number (e.g., 549)
<code>\$\$EVENT_TYPE</code>	Event type (e.g., Equipment)
<code>\$\$EVENT_TYPE_ID</code>	Event type ID number (e.g., 5)
<code>\$\$DATE_TIME</code>	Date and time of event (e.g., Thu 19 Oct 2006 01:50:28 PM CEST)
<code>\$\$THRESHOLD</code>	Value in case of a threshold event (e.g., 86.774)

b) The following additional reserved words are available for events generated from received SNMP Trap and SNMP Inform notifications, which are not included in the built-in notification-to-event mapping table:

<code>\$\$NOTIFICATION</code>	Identity (name) of SNMP notification
<code>\$\$TIME_STAMP</code>	Notification's time stamp value
<code>\$\$AGENT_ADDRESS</code>	Address of notification sender
<code>\$\$V1AGENT_ADDRESS</code>	SNMPv1 agent address (from SNMPv1 Trap)
<code>\$\$PROTOCOL</code>	SNMP protocol version of notification
<code>\$\$ENTERPRISE</code>	Enterprise associated with notification
<code>\$\$COMMUNITY</code>	SNMPv1/v2c community string
<code>\$\$TRANSPORT</code>	Notification's transport protocol
<code>\$\$PORT</code>	UDP port of notification receiver
<code>\$\$VBCOUNT</code>	Total number of variable bindings in notification
<code>\$\$VB(E)</code>	Log E bindings. E can be individual bindings from the variable bindings list (1,3,19), ranges of bindings (3-6), or both (1,3-6,19).
<code>\$\$VBALL</code>	Log all bindings

```
$SEC_USER_NAME      SNMPv3 security user name
$SEC_AUTH_PROTOCOL  SNMPv3 authentication protocol
$SEC_PRIV_PROTOCOL  SNMPv3 privacy protocol
$SEC_CONTEXT        SNMPv3 context name
```

c) The following for-each loop reserved words are available:

```
$FOR_EACH_BEGIN      Starts the for-each loop
$FOR_EACH_END        Ends the for-each loop
```

Every reserved word inside the for-each loop (i.e., between the `$FOR_EACH_BEGIN` and `$FOR_EACH_END` reserved words) is expanded repeatedly for each event that is merged and sent in one message.

**Example of a text file defining actions (e.g., `actionconfig.txt`):**

```
[action1]
mstate = 0
name = dummy_cmd
type = CMD
desc = command test1
filter_name = critical_events
cmd_line = //Engine/mycmd.sh
cmd_args = $SEVERITY $SOURCE_NAME $MESSAGE
auto_term_enable = true
auto_term_timeout = 30

[action2]
mstate = 1
name = sms test
type = SMS
desc = sms test1
filter_name = critical_events
port = /dev/ttyS0
speed = 19200
data_bits = 8
parity = 0
stop_bits = 1
phone = 38641222333
msg = NIE '$SEVERITY' alarm: '$MESSAGE' from '$SOURCE_NAME' at
'$DATE_TIME'

[action3]
mstate = 0
name = Test mail
type = MAIL
desc = test mail
filter_name = my_filter
mail_server_addr = mail.my-company.com
mail_server_port = 25
mail_server_timeout = 10
user_name = ni@my-company.com
```

```
user_org = my-company
user_email = ni@my-company.com
user_reply_to = ni@my-company.com
send_to = admin@my-company.com
msg_custom_subject = Net Inspector Engine '$SEVERITY' alarm:
'$MESSAGE' from '$SOURCE_NAME' at '$DATE_TIME'
msg_body_file = //Engine/workspace/action3_msg_body.txt
msg_as_html = false
msg_merge_max = 10
msg_merge_interval = 5
```

## 5.5 Section [event]

The [event] section provides the event database information. The `type` parameter specifies the type of the event database used. Valid values are:

- ❑ `odbc` – event database is accessible via the ODBC interface,

The `dsn` parameter value is the ODBC *data source name* of the event database.

The `user` and `passwd` parameters are used for specifying the username and password for accessing the database (if required).

The `write` parameter determines what events are logged in the event database. Valid values are:

- ❑ `all` – alarms and events are logged,
- ❑ `alarm` – only alarms are logged.

The `maintenance_type` and `maintenance_value` parameters control the database maintenance operation. The `maintenance_type` parameter specifies whether the maintenance operation is enabled, and if yes, how the database size is controlled (i.e., by the number of events, or by the age of events stored in the database). The `maintenance_value` parameter specifies the max. number of events or the max. age of events to be kept in the database. The `maintenance_type` value also determines the units in which the `maintenance_value` is expressed. Valid values of the `maintenance_type` parameter are:

- ❑ `days` or `hours` or `minutes` or `seconds` – limits the age of events in the database,
- ❑ `count` – limits the number of events in the database,
- ❑ `none` – disables the database maintenance.

Every hour Net Inspector Server checks if the condition for performing the database maintenance is met. If the condition is met, it carries out the maintenance operation. If the `maintenance_type` value is `days` or `hours` or `minutes` or `seconds`, the maintenance operation deletes all events older than specified by the

`maintenance_value` parameter from the events database. If the `maintenance_type` value is `count`, the oldest events are deleted from the database (when required) in order to keep the total number of events below or at the value of the `maintenance_value` parameter.

The `statistics` parameter determines what statistics about database records is kept. Valid values of this parameter are:

- ❑ `count` – the number of events is kept track of,
- ❑ `count_time` – the number of events and their timestamps are kept track of,
- ❑ `none` – statistics is disabled.

Example:

```
[event]
; available type is: odbc
type = odbc
dsn = Net Inspector DataBase
user = NI6
passwd = NI6
; available write types are: all, alarm
write = alarm
; available maintenance types are: days, hours, minutes,
seconds, count, none
maintenance_type = count
maintenance_value = 10000
; available statistics types are: count, count_time, none
statistics = count_time
```

## 5.6 Section [log]

The `[log]` section contains parameters that control the Net Inspector Server logging behavior.

Net Inspector Server logs messages to the following log files located in the `//Engine/log` directory:

- ❑ `niengine.log`,
- ❑ `niengine_action.log`,
- ❑ `niengine_stat.log`,
- ❑ `niengine_trap.log`.

The `system` parameter controls what messages related to Net Inspector Server functioning are logged. Valid values are:

- ❑ `debug` – all messages are logged,

- ❑ `notice` – all normal (but relevant) messages, warning and error messages are logged,
- ❑ `warning` – only warning and error messages are logged,
- ❑ `error` – only error messages are logged.

The `system_size` parameter controls the size of the `niengine.log` file. The maximum value of this parameter is 2 GB, while the default value is 10 MB. The value of this parameter must be specified in bytes; for example, 10 MB (=10485760 bytes) needs to be entered as 10485760.

The `action` parameter controls the logging of Net Inspector Server actions. Valid values are:

- ❑ `admin` – all actions performed by the users with admin access rights are logged,
- ❑ `none` – actions are not logged.

The `default_size` parameter controls the size of the `niengine_action.log` and `niengine_stat.log` log files. The maximum and default value of this parameter is 2 MB. The value must be specified in bytes.

The `stat` parameter controls the logging of Net Inspector Server operating statistics. Valid values are:

- ❑ `all` – statistics on Net Inspector Server functioning is logged,
- ❑ `none` – statistics on Net Inspector Server functioning is not logged.

The `stat_interval` parameter value (in minutes) specifies the interval for statistics logging.

The `trap` parameter controls the logging of received SNMP notifications. Valid values are:

- ❑ `all` – all received SNMP notifications are logged,
- ❑ `none` – SNMP notifications are not logged.

The `trap_size` parameter controls the size of the `niengine_trap.log` file. The maximum value of this parameter is 2 GB, while the default value is 10 MB. The value must be specified in bytes.

The `trap_format` parameters specifies which details of SNMP notifications are logged and in what format. This is achieved by using the reserved words, which are:

<code>\$NOTIFICATION</code>	The identity (name) of the SNMP notification
<code>\$TIME_STAMP</code>	The notification's time stamp value
<code>\$AGENT_ADDRESS</code>	The address of the notification sender
<code>\$V1AGENT_ADDRESS</code>	The SNMPv1 agent address from the SNMPv1 Trap
<code>\$PROTOCOL</code>	The SNMP protocol version of the notification
<code>\$ENTERPRISE</code>	The enterprise associated with notification
<code>\$COMMUNITY</code>	The SNMPv1/v2c community string
<code>\$TRANSPORT</code>	The notification's transport protocol

\$PORT	The UDP port of notification receiver
\$VBCOUNT	The total number of variable bindings in the notification
\$VB(E)	Log E bindings. E can be individual bindings from the variable bindings list (1,3,19), ranges of bindings (3-6), or both (1,3-6,19).
\$VBALL	Log all bindings
\$SEC_USER_NAME	SNMPv3 security user name
\$SEC_AUTH_PROTOCOL	SNMPv3 authentication protocol
\$SEC_PRIV_PROTOCOL	SNMPv3 privacy protocol
\$SEC_CONTEXT	SNMPv3 context name

**Example:**

```
[log]
; system log types are: debug, notice, warning, error
system = notice
system_size = 50000000
; action log types are: admin, none
action = admin
default_size = 1200000
; trap types are: all, none
trap = all
trap_size = 70000000
trap_format = $NOTIFICATION($PROTOCOL) $AGENT_ADDRESS $COMMUNITY
$VB(1-3)
; stat types are: all, none
stat = all
; interval is in minutes
stat_interval = 5
```

## 5.7 Section [snmp notifications]

The [snmp notifications] section controls the SNMP notification reception.

The `port` parameter specifies on which UDP port(s) Net Inspector Server listens to for incoming SNMP notifications. More than one port can be specified, using the following notation:

```
port = 6162
port1 = 7000
...
portN = 8000
```

The `assign_to_object` parameter controls whether the received SNMP notification messages are assigned to managed objects or not.

Valid values of this parameter are `true` and `false`. If the value of this parameter is `true`, Net Inspector Server checks the address from which the generic SNMP notification has been sent and tries to assign the received SNMP notification to the

managed object with the same address. If the managed object with the matching address exists in Net Inspector, its name is displayed in the “Source” field of the alarm or event that has been created from the notification. If the managed object with the matching address does not exist in Net Inspector, the generic SNMP notification is either assigned to the `SNMP notification system` object or silently discarded, depending on the value of the `ignore_unassigned` parameter. If the value of this parameter is `false`, Net Inspector Server does not assign received SNMP notifications to managed objects. Whether notifications in this case will be discarded or converted to events/alarms and displayed depends on the value of the `ignore_unassigned` parameter.

The `ignore_unassigned` parameter controls whether the received SNMP notification messages that were not assigned to managed objects (because no such managed objects exist in Net Inspector or because the `assign_to_object` value is set to `false`) are ignored or not. Valid values are `true` and `false`.

The `unknown_to_alarm` parameter controls whether the “unknown” SNMP notifications are mapped to alarms and thus logged and displayed by Net Inspector or not. “Unknown” notifications are those SNMP notifications for which neither built-in nor user-defined trap-to-alarm rules exist in Net Inspector. Note that Net Inspector comes with a built-in set of rules for mapping the generic SNMP notifications (`coldStart`, `warmStart`, `linkDown`, `linkUp`, `authenticationFailure`, `egpNeighborLoss`) to alarms/events. Therefore, the generic SNMP notifications are “known” notifications. Additionally, users can define their own trap-to-alarm mapping rules for enterprise specific SNMP notifications and thus make those types of notifications “known” to Net Inspector.

The `unknown_to_event` parameters controls whether the “unknown” SNMP notifications mapped to events or not. If the value of the `unknown_to_alarm` parameter is `true`, then the value of this parameter must also be `true`.

The `check_community` parameter controls if the community names included in received SNMP notification messages should be compared with the trap community names configured for the managed objects the notifications are being assigned to. Valid values are `true` and `false`.

Example:

```
[snmp notifications]
port = 6162
port1 = 7000
unknown_to_alarm = true
unknown_to_event = true
assign_to_object = true
ignore_unassigned = false
check_community = false
```

---

## 5.8 Section [snmp agent]

---

The `[snmp agent]` section controls the connection with the SNMP agent extension.

The `supported` parameter controls if the connection between Net Inspector Server and the Net Inspector SNMP agent extension is enabled or disabled. Valid values are `true` and `false`.

The `ipaddr` parameter specifies the IP address of the PC running the SNMP agent extension application.

The `downscaleid` parameter controls whether the device indices should be downscaled from 32 to 16 bits. If this parameter is not present or if its value is 0, the parameter is ignored. If the value of this parameter is in the range 1-6, the index is downscaled so that the upper 16 bits, which represent the configuration number (0-63), are copied to the upper N bits of the 16-bit index, where the N is the value of the `downscaleid` parameter.

Example:

```
[snmp agent]
supported = true
ipaddr = 127.0.0.1
downscaleid = 4
```

## 6 NET INSPECTOR SERVER PROFILES FILE

---

The profiles used by Net Inspector Server to poll managed objects can be specified in the `nieprofiles.ini` file. This initialization file should be stored in the `//Engine/workspace` directory. When Net Inspector Server starts up, it reads the profiles from the `nieprofiles.ini` file, and initializes itself accordingly. If the `nieprofiles.ini` file is not present in the `//Engine/workspace` directory, the default profile parameters are used.

The Net Inspector profiles file (`nieprofiles.ini`) is a plain ASCII file that can be edited in any text editor. It can contain two types of profiles (sections):

- ❑ `poll profile` - contains parameters for polling managed objects by means of ICMP and SNMP protocols (e.g., polling intervals, monitored OID groups, etc.)
- ❑ `snmp access profile` - contains SNMP access parameters used for polling SNMP agents on managed objects (SNMP version, community names, etc.)

### 6.1 Section [poll profile]

---

The `[poll profile]` section includes a set of parameters for polling managed objects by means of ICMP and SNMP protocols (e.g., polling intervals, monitored OID groups, etc.).

The `nieprofiles.ini` file can contain more than one polling profile section. In this case, sections must be named according to the following scheme: `[poll profile]`, `[poll profile1]`, `[poll profile2]`, ...`[poll profileN]`, where N is a unique polling profile number.

The polling profile sections contain the following parameters:

- ❑ `name` - the name of the polling profile,
- ❑ `polling_plan` - specifies what protocols are used for monitoring (ICMP, SNMP) and what OID groups are monitored (when SNMP monitoring is enabled). Valid values are (more than one value can be specified):
  - ❑ `icmp_ping` - enables the ICMP Ping polling,
  - ❑ `snmp_ping` - enables the SNMP Ping polling,
  - ❑ `snmp_if` - enables monitoring network interfaces on managed objects via SNMP,
  - ❑ `snmp_resources` - enables monitoring the managed object system resources, like the memory consumption, CPU load, etc. via SNMP,
  - ❑ `snmp_storage` - enables monitoring the data storage units, like the disk capacity utilization etc. via SNMP,
- ❑ `timeout` - specifies the timeout value for ICMP and SNMP queries (in seconds),

- ❑ `retries` – specifies how many times the request will be retransmitted after the first timeout occurs (applies to both, SNMP and ICMP queries),
- ❑ `ttl` – specifies the TTL (Time To Live) value for ICMP packets,
- ❑ `ping_poll_interval` – specifies the ICMP in SNMP Ping polling interval (in seconds),
- ❑ `stat_poll_interval` – specifies the polling interval for collecting statistics via SNMP (in seconds),
- ❑ `resync_interval` – sets the alarm resynchronization interval (in seconds). The alarm resynchronization occurs if the managed object does not respond to queries within this interval.

#### Threshold parameters:

Valid value for threshold parameters consists of three numbers separated by comma (,). The first number controls if the threshold is enabled (1) or disabled (0), the second and third numbers specify the threshold alarm raise and clear values. The following threshold parameters exist:

- ❑ `if_inutil_threshold` – Controls the interface inbound utilization threshold values.
- ❑ `if_oututil_threshold` – Controls the interface outbound utilization threshold values.
- ❑ `if_inerrorrate_threshold` – Controls the interface inbound error rate threshold values.
- ❑ `if_outerrorrate_threshold` – Controls the interface outbound error rate threshold values.
- ❑ `if_status_threshold` – Controls the interface status threshold values.
- ❑ `hr_memoryused_threshold` – Controls the memory usage threshold values.
- ❑ `hr_processorload_threshold` – Controls the processor load threshold values.
- ❑ `hr_storageused_threshold` – Controls the storage usage threshold values.

#### Example:

```
[poll profile]
name = default
polling_plan = icmp_ping,snmp_ping,snmp_if,
timeout = 3
retries = 2
ping_poll_interval = 30
stat_poll_interval = 60
ttl = 64
resync_interval = 120
if_inutil_threshold = 1,80,70
```

```
if_oututil_threshold = 1,80,70
if_inerrorate_threshold = 1,20,10
if_outerrorate_threshold = 1,20,10
if_status_threshold = 1,1,0
hr_memoryused_threshold = 1,20,10
hr_processorload_threshold = 1,5,2
hr_storageused_threshold = 1,20,10

[poll profile1]
name = fast_test
polling_plan = icmp_ping,snmp_ping
timeout = 10
retries = 3
ping_poll_interval = 10
stat_poll_interval = 15
ttl = 64
resync_interval = 0
```

## 6.2 Section [snmp access profile]

The [snmp access profile] section includes parameters for accessing the SNMP agents on managed objects. It also specifies the community name included in SNMP notifications sent by SNMP agents.

The `nieprofiles.ini` file can contain more than one SNMP access profile section. In this case, sections must be named according to the following scheme: [snmp access profile], [snmp access profile1], ...[snmp access profileN], where N is a unique SNMP access profile number.

The SNMP access profile sections contain the following parameters:

- ❑ `name` - the name of the profile,
- ❑ `version` – specifies the version of SNMP protocol used for querying SNMP agents on managed objects. Valid values are:
  - ❑ `snmpv1`
  - ❑ `snmpv2c`
  - ❑ `snmpv3`
- ❑ `port` – specifies the UDP port number on which SNMP agents on managed objects listen to for incoming SNMP requests,
- ❑ `read_context` – specifies the community name expected by the SNMP agents for SNMP Get, GetNext and GetBulk operations,
- ❑ `set_context` – specifies the community name expected by the SNMP agents for SNMP Set operation,

- ❑ `trap_context` - specifies the community name included in SNMP Trap messages sent by the SNMP agents on managed devices. If this parameter is not specified or if its value is missing, this community name is not checked.

**SNMPv3 access profile parameters:**

- ❑ `v3_user_name` - The name of the SNMPv3 USM user.
- ❑ `v3_context_name` - The SNMPv3 USM context name.
- ❑ `v3_not_localize_keys` - Controls if the software uses localized or non-localized authentication and privacy keys. Valid values are true and false (default).
- ❑ `v3_auth_proto` - The SNMPv3 authentication protocol (HMAC-MD5 or HMAC-SHA).
- ❑ `v3_auth_key` - The SNMPv3 authentication security key (hex).
- ❑ `v3_priv_proto` - The SNMPv3 privacy protocol (CBC-DES or CFB-AES-128).
- ❑ `v3_priv_key` - The SNMPv3 privacy security key (hex).

**Example:**

```
[snmp access profile]
name = default
version = snmpv1
port = 161
read_context = public
set_context =
trap_context =
```

```
[snmp access profile1]
name = snmpv3_profile
version = snmpv3
port = 161
read_context = public
set_context = private
trap_context = SNMP_trap
v3_user_name = MD5_DES_User
v3_context_name = public
v3_not_localize_keys = false
v3_auth_proto = hmac-md5
v3_auth_key = B65EDE1E0371C43BDFDBB0F189096F15
v3_priv_proto = cbc-des
v3_priv_key = A634AEB72FB4BA9C331FA6BE766311CB
```

## **7 CONFIGURING SNMP NOTIFICATION DESTINATION ON SNMP AGENTS**

---

To be able to effectively monitor alarms on managed objects with Net Inspector, you need to configure the SNMP agents on managed devices to send SNMP notifications to Net Inspector Server. Otherwise, Net Inspector will not receive those notifications and consequently will not display and notify you about the corresponding alarms. For details on configuring SNMP agents on managed objects, kindly refer to user manuals of the relevant network elements.