

KASPERSKY LAB

---

# Kaspersky Anti-Virus 5.6 for Linux Mail Server

ADMINISTRATOR'S  
GUIDE

KASPERSKY ANTI-VIRUS 5.6 FOR LINUX MAIL SERVER

---

# Administrator's Guide

© Kaspersky Lab  
<http://www.kaspersky.com>

Revision date: November, 2008

# Contents

CHAPTER 1. INTRODUCTION .....	7
1.1. What's new .....	8
1.2. Product requirements .....	9
1.3. Service for registered users .....	10
CHAPTER 2. THE STRUCTURE AND OPERATING ALGORITHM OF THE APPLICATION .....	11
CHAPTER 3. INSTALLING AND UNINSTALLING THE APPLICATION .....	14
3.1. Installing the application on a server running Linux .....	14
3.2. Installing the application on a server running FreeBSD .....	15
3.3. Location of application files .....	16
3.3.1. Location of files on a server running Linux .....	16
3.3.2. Location of files on a server running FreeBSD .....	18
3.4. Post-installation setup .....	19
3.5. Configuration of permission rules in SELinux and AppArmor systems .....	22
3.6. Installing the Webmin module to manage Kaspersky Anti-Virus .....	24
3.7. Application removal .....	26
CHAPTER 4. INTEGRATION WITH MTA .....	28
4.1. Integration with Exim .....	29
4.1.1. Post-queue integration using modification of routers .....	29
4.1.2. Pre-queue integration using dynamically loaded library .....	32
4.2. Integration with Postfix .....	34
4.2.1. Post-queue integration .....	35
4.2.2. Pre-queue integration .....	37
4.2.3. Integration with Militer .....	39
4.3. Integration with qmail .....	41
4.4. Integration with Sendmail .....	42
4.4.1. Integration with Sendmail using <i>.cf</i> file .....	43
4.4.2. Integration with Sendmail using <i>.mc</i> file .....	44

---

CHAPTER 5. ANTI-VIRUS PROTECTION OF E-MAIL .....	46
5.1. Setting up groups .....	46
5.2. Definition of e-mail analysis policy .....	48
5.3. E-mail scanning mode.....	48
5.3.1. Anti-virus scanning .....	49
5.3.2. Content filtering .....	50
5.4. Actions over objects .....	52
5.5. Predefined security profiles .....	53
5.5.1. <i>Recommended</i> profile .....	54
5.5.2. <i>Maximum protection</i> profile .....	54
5.5.3. <i>Maximum performance</i> profile .....	55
5.6. Backup .....	56
5.7. Notifications .....	57
5.7.1. Setting up notifications.....	57
5.7.2. Notification templates .....	59
5.7.3. Customizing notification templates.....	62
CHAPTER 6. ANTI-VIRUS PROTECTION OF FILE SYSTEMS .....	70
6.1. Scan scope .....	71
6.2. Object scan and disinfection mode.....	72
6.3. Actions to be performed on objects .....	72
6.4. On-demand scan of an individual directory .....	74
6.5. Scheduled scan .....	74
6.6. Sending notifications to the administrator.....	75
CHAPTER 7. UPDATING THE ANTI-VIRUS DATABASES .....	76
7.1. Automatically updating the anti-virus database.....	77
7.2. On-demand updating of the anti-virus database .....	78
7.3. Creating a network directory to store the updates.....	79
CHAPTER 8. KEY MANAGEMENT .....	81
8.1. Viewing key details .....	82
8.2. Renewing your key.....	84
CHAPTER 9. REPORTING AND STATISTICS .....	85
9.1. Application logging .....	85
9.2. Application statistics .....	87

---

CHAPTER 10. ADVANCED SETTINGS .....	91
10.1. Monitoring of protection status via SNMP .....	91
10.2. Using the application's setup script.....	95
10.3. Managing the application from the command line .....	97
10.4. Additional informational fields in messages.....	99
10.5. Localization of displayed date and time format .....	100
CHAPTER 11. TESTING THE APPLICATION .....	101
APPENDIX A. ADDITIONAL INFORMATION.....	103
A.1. Application configuration file <i>kav4lms.conf</i> .....	103
A.1.1. Section [ <i>kav4lms:server.settings</i> ].....	103
A.1.2. Section [ <i>kav4lms:server.log</i> ].....	106
A.1.3. Section [ <i>kav4lms:server.statistics</i> ].....	107
A.1.4. Section [ <i>kav4lms:server.snmp</i> ].....	108
A.1.5. Section [ <i>kav4lms:server.notifications</i> ] .....	110
A.1.6. Section [ <i>kav4lms:filter.settings</i> ].....	111
A.1.7. Section [ <i>kav4lms:filter.log</i> ] .....	114
A.1.8. Section [ <i>kav4lms:groups</i> ].....	115
A.1.9. Section [ <i>path</i> ].....	115
A.1.10. Section [ <i>locale</i> ].....	116
A.1.11. Section [ <i>options</i> ].....	117
A.1.12. Section [ <i>updater.path</i> ].....	117
A.1.13. Section [ <i>updater.options</i> ].....	117
A.1.14. Section [ <i>updater.report</i> ].....	119
A.1.15. Section [ <i>updater.actions</i> ] .....	119
A.1.16. Section [ <i>scanner.display</i> ].....	121
A.1.17. Section [ <i>scanner.options</i> ].....	121
A.1.18. Section [ <i>scanner.report</i> ].....	124
A.1.19. Section [ <i>scanner.container</i> ] .....	125
A.1.20. Section [ <i>scanner.object</i> ].....	126
A.1.21. Section [ <i>scanner.path</i> ] .....	127
A.2. Group configuration file .....	127
A.2.1. Section [ <i>kav4lms:groups.&lt;group_name&gt;.definition</i> ].....	128
A.2.2. Section [ <i>kav4lms:groups.&lt;group_name&gt;.settings</i> ].....	129
A.2.3. Section [ <i>kav4lms:groups.&lt;group_name&gt;.actions</i> ].....	131
A.2.4. Section [ <i>kav4lms:groups.&lt;group_name&gt;.contentfiltering</i> ] .....	132

---

A.2.5. Section [ <i>kav4lms:groups.&lt;group_name&gt;.notifications</i> ].....	136
A.2.6. Section [ <i>kav4lms:groups.&lt;group_name&gt;.backup</i> ].....	138
A.3. Command line parameters for component <i>kav4lms-licensemanager</i> .....	138
A.4. Return codes of the <i>kav4lms-licensemanager</i> component.....	139
A.5. Command line parameters for component <i>kav4lms-keepup2date</i> .....	140
A.6. Return codes of the <i>kav4lms-keepup2date</i> component.....	141
APPENDIX B. KASPERSKY LAB.....	142
B.1. Other Kaspersky Lab Products .....	143
B.2. Contact Us.....	153
APPENDIX C. THIRD-PARTY SOFTWARE.....	154
C.1. <i>Pcre</i> library .....	154
C.2. <i>Expat</i> library .....	155
C.3. <i>AgentX++v1.4.16</i> library .....	155
C.4. <i>Agent++v3.5.28a</i> library .....	162
C.5. <i>Boost v 1.0</i> library .....	163
C.6. <i>Milter</i> library.....	164
C.7. <i>Libkavexim.so</i> library .....	166

# CHAPTER 1. INTRODUCTION

**Kaspersky Anti-Virus® 5.6 for Linux Mail Server** (hereinafter referred to as *Kaspersky Anti-Virus* or the *application*) provides anti-virus processing of mail traffic and file systems of servers running the Linux or FreeBSD operating systems, and using the Sendmail, Postfix, qmail, or Exim MTA.

This application allows the user to:

- Check for the presence of threats all server file systems as well as incoming and outgoing mail messages.
- Detect infected, suspicious, corrupted, and password-protected objects as well as objects that cannot be scanned.
- Neutralize threats detected in files and mail messages. Disinfect infected objects.
- Back up e-mail messages prior to their anti-virus processing and filtration.
- Process mail traffic according to rules preset for groups of senders and recipients.
- Provide content filtering of mail traffic by name, type and size of attached files, and use individual processing rules for the filtered objects.
- Notify the sender, recipients, and administrator about detection of mail messages that contain infected, suspicious, password protected objects or objects that cannot be scanned.
- Generate statistics and reports on application performance.
- Update the anti-virus databases, either using a schedule or on demand, by downloading update files from Kaspersky Lab's update servers.

The anti-virus database is used to search for and attempt to cure infected objects. During the scan each file is analyzed for the presence of threats by comparing the file's code with code typical of various threats.

- Configure and manage Kaspersky Anti-Virus both locally (using standard OS means including command line options, signals and modification of the application configuration file) and remotely via the web-based interface provided by the Webmin program.
- Obtain information about product configuration and activity statistics via SNMP and configure the application to generate SNMP traps when specified events occur.

## 1.1. What's new

Version 5.6 of **Kaspersky Anti-Virus for Linux Mail Server** merges the features of Kaspersky Anti-Virus 5.5 for Linux and FreeBSD Mail Server and Kaspersky Anti-Virus 5.6 for Sendmail with Milter API and adds the following improvements:

- Both pre-queue and post-queue integration is supported for Exim. In case of pre-queue integration, e-mail is transferred for scanning before its addition to the mail system queue while post-queue integration means that messages are scanned after addition to the queue. Automatic integration using the application configuration script is now available. See Chapter 4 on p. 28 for details on the integration procedure.
- Opportunities for configuration of mail scanning functionality have been enhanced: two scanning methods are now available. A message can be scanned as a single object or using combined approach – first as a single object and then as a collection of its parts. These methods differ in terms of the provided protection level. See 5.2 on p. 48 for details.
- The application's configuration has changed. Individual configuration of separate groups of senders and recipients is now supported. See 5.1 on p.46 for details of configuring groups.
- The list of actions performed over messages has been extended. New action type depending upon the detected malware has been added. See 5.4 on p. 52 for details.
- Content filtering capabilities have been extended by adding filtering by attachment size criterion. See 5.3.2 on p. 50 for details.
- The library of notification templates has been supplemented with added administrator templates. Templates are now stored in a separate directory.
- The opportunity to place infected objects in Backup is no longer supported.
- Backup functionality has been extended – information files can be created for each backup entry. See 5.6 on p.56 for details.
- Reporting has been improved by increasing the logging setup thoroughness. See 9.1 on p.85 for details.
- Statistics functionality has been extended by adding per-message statistics. See 9.2 on p.87 for details.
- SNMP-queries for configuration, statistics, application status are now supported. SNMP-traps are also supported. See 10.1 on p. 91 for details.



- Command line administration tool is added to the application's package. It is capable of managing various aspects of the application's functionality. See 10.3 on p. 97 for details.

## 1.2. Product requirements

The system requirements for Kaspersky Anti-Virus are:

- Hardware requirements for a mail server with about 200 MB of traffic per day:
  - Intel Pentium IV, 3 GHz processor or higher;
  - 1 GB RAM;
  - 200 MB available space on your hard drive (this amount does not include space necessary for storing backup message copies).
- Software requirements:
  - One of the following 32-bit operating systems:
    - Red Hat Enterprise Linux Server 5.2;
    - Fedora 9;
    - SUSE Linux Enterprise Server 10 SP2;
    - openSUSE 11.0;
    - Debian GNU/Linux 4.0 r4;
    - Mandriva Corporate Server 4.0;
    - Ubuntu 8.04.1 Server Edition;
    - FreeBSD 6.3, 7.0.
  - One of the following 64-bit operating systems:
    - Red Hat Enterprise Linux Server 5.2;
    - Fedora 9;
    - SUSE Linux Enterprise Server 10 SP2;
    - openSUSE Linux 11.0.
  - One of the following mail systems: Sendmail 8.12.x or higher, qmail 1.03, Postfix 2.x, Exim 4.x;
  - Optional - the Webmin program ([www.webmin.com](http://www.webmin.com)) for remote administration of Kaspersky Anti-Virus;
  - Perl version 5.0 or higher ([www.perl.org](http://www.perl.org)).

## 1.3. Service for registered users

Kaspersky Lab offers its legal users a broad range of services maximizing the efficiency of Kaspersky Anti-Virus software.

By purchasing a subscription you become a registered software user entitled to the following services throughout the license period:

- software upgrades for this software application;
- consultations regarding issues pertaining to installation, configuration and use of this software, available over the telephone or via e-mail;
- notifications about new software products from Kaspersky Lab, and about new virus outbreaks. This service is provided to users who have subscribed to the Kaspersky Lab e-mail newsletter service.

**Note:**

Kaspersky Lab does not give advice on the performance and use of your operating system, third party software or various other technologies.

# CHAPTER 2. THE STRUCTURE AND OPERATING ALGORITHM OF THE APPLICATION

Kaspersky Anti-Virus consists of the following components:

- Filter – the service for connection to the mail system, a separate program providing for interaction between Kaspersky Anti-Virus and a specific MTA. The product distribution package includes modules for each supported mail systems:
  - *kav4lms-milter* – Milter service for connection with Sendmail and Postfix via Milter API.
  - *kav4lms-filter* – SMTP service for connection with Postfix and Exim.
  - *kav4lms-qmail* – mail queue handler for qmail.
- *kavmd* - central service of the application, listening to the filter requests and implementing the anti-virus functionality of the application protecting e-mail traffic.
- *kav4lms-kavscanner* – provides for anti-virus protection of server file systems.
- *kav4lms-keepup2date* – provides for updating of the anti-virus database downloading new data from update servers of Kaspersky Lab or a local directory.
- *kav4lms-licensemanager* – component for operations with product keys: installation, removal, viewing statistical information.
- *kav4lms.wbm* – Webmin plug-in module for remote management of the application via web-based interface (optional), which allows configuration and launch of updates for the anti-virus database, viewing of statistical information, definition of actions over objects depending upon their status, and monitoring of application activity results.
- *kav4lms-cmd* – utility for Anti-Virus management via the command line.

The application uses the following algorithm to check e-mail:

1. The filter receives a message from MTA. If the filter and the central service are running on the same computer, then names of message files are passed instead of the actual messages for analysis.
2. The filter determines the groups that the message belongs to, selects the group with the highest priority (see 5.1 on p. 46) and transfers the letter for analysis to the central service of the application. If there is no such group, then the application will process that message using the rules for the **Default** group included into its distribution package.

The central service scans the message using the parameters specified in the configuration file of the group. Depending upon the method defined in the **policy**, the application can scan the message as a single solid object or use combined approach scanning it first as a whole and then checking its individual parts (see 5.2 on p. 48).

Combined analysis is more thorough and provides for higher protection level although its performance is somewhat lower assumes checking the message as a whole or as a whole and then part-by-part (combined policy).

3. If anti-virus mail scanning is enabled (see 5.3 on p. 48), the central service checks a message as a single object. In accordance with the status assigned after that check (see 5.3.1 on p. 49) the central service: blocks delivery, rejects or allows the message, replaces it with a warning, modifies its headers (see 5.4 on p. 52). If special processing is defined for individual malware types (the **VirusNameList** option), the specified actions will be performed if they are detected (**VirusNameAction** option). Message processing order is specified in the configuration file of the group.

The application creates a backup copy of the original message before its processing if that step is enabled in the group settings.

4. After anti-virus message scan the application performs its filtering if it is enabled in the group settings.

Filtration can be performed by attachment name, type and size (see 5.3.2 on p. 50). The check results in the actions defined by the filtration settings in the configuration file of the group. Processed objects matching the filtration criteria are passed over for further analysis part by part, if combined processing method is enabled in the group settings.

5. During e-mail inspection part by part the application parses its MIME structure and processes message components.

Message objects are treated in accordance with the status assigned to each individual object irrespectively of the status assigned to the message as a whole.

If a message is recognized as infected after its processing as a single object while no threat is found after examination of its parts, the application will handle the whole message using the action defined for infected mail (**InfectedAction** option). If the nesting level of an object attached to a clean message exceeds the limit specified in group settings (**MaxScanDepth** option), the application will handle the whole message using the action defined for letters causing errors during scan (**ErrorAction** option).

While processing message objects, the central service renames, deletes or replaces an object with a warning, adds informational headers or allows a message to pass (see 5.4 on p. 52). Infected messages get disinfected. The application creates a backup copy of the whole original message prior to processing of its object (unless it has been made earlier) if that step is enabled in the group settings.

6. After scanning and processing, the central service returns the message to filter. The processed message together with the notifications about results of scanning and disinfection is conveyed to the MTA, which delivers the e-mail message to local users or relays it to other mail servers.

# CHAPTER 3. INSTALLING AND UNINSTALLING THE APPLICATION

Before installing Kaspersky Anti-Virus, you are advised to make the following preparations for your system:

- Make sure your system meets the hardware and software requirements listed in section 1.2 on page 9.
- Make backup copies of configuration files of the mail system installed on your server.
- Set up an Internet connection.
- Log in to the system with **root** access rights or any other account with superuser privileges.

## **Warning!**

We advise that you install the application in off hours or when the mail traffic has the lowest intensity!

## **3.1. Installing the application on a server running Linux**

For servers running the Linux operating system, Kaspersky Anti-Virus is distributed in *two different installation packages*, depending on the type of your Linux distribution.

To install the application under Red Hat Enterprise Linux, Fedora, SUSE Linux Enterprise Server, openSUSE and Mandriva Linux, use the *rpm* package.

To initiate installation of Kaspersky Anti-Virus from the *.rpm* package, enter the following on the command line:

```
# rpm -i <package_name>
```

**Warning!**

After installing the application from the rpm package, you must run the *postinstall.pl* script to perform post-installation configuration. The default location of the *postinstall.pl* script is in the */opt/kaspersky/kav4lms/lib/bin/setup/* directory (in Linux) and in the */usr/local/libexec/kaspersky/kav4lms/setup/* directory (in FreeBSD)!

In Debian GNU/Linux and Ubuntu, the installation is performed from a deb package.

To initiate installation of Kaspersky Anti-Virus from the .deb package, enter the following on the command line:

```
# dpkg -i <package_name>
```

After you enter the command, the application will be installed automatically. Once the installation completes, information about post-install configuration will be displayed (see 3.4 on p. 19).

**Warning!**

The procedure of application setup under Mandriva distributions has some peculiarities.

To allow correct launch of Kaspersky Anti-Virus after installation, you will have to make sure that the */root/tmp/* directory is used for storage of temporary files in the operating system and the account used to run the application (by default, *kluser*) has the right to write to the directory.

You might have to change the access rights for the directory, or redefine or delete the **TMP**, **TEMP** environment variables to make the system use another directory (e.g., */tmp/*) with the rights required for application functioning.

## 3.2. Installing the application on a server running FreeBSD

The distribution file for installing Kaspersky Anti-Virus on servers running FreeBSD OS is supplied as a *pkg* package.

To initiate installation of Kaspersky Anti-Virus from a *pkg* package, enter one of the following at the command line:

```
# pkg_add <package_name>
```

After you enter the command, the application will be installed automatically. Once the installation completes, information about post-install configuration will be displayed (see 3.4 on p. 19).

## 3.3. Location of application files

During Kaspersky Anti-Virus setup the product installer copies application files to program directories on server.

### Attention!

To make the man pages for the application available upon the `man <man_page_name>` command, the following steps are necessary:

- for Debian Linux, Ubuntu Linux, SUSE Linux distributions, add the line below to the `/etc/manpath.config` file:

```
MANDATORY_MANPATH /opt/kaspersky/kav4lms/share/man
```

- for Red Hat Linux and Mandriva Linux distributions, add the line below to the `/etc/man.config` file:

```
MANPATH /opt/kaspersky/kav4lms/share/man
```

- for FreeBSD distributions, add the line below to the `/etc/manpath.config` file:

```
MANDATORY_MANPATH /usr/local/man
```

If your system uses the **MANPATH** variable, add to the list of its values the path to the directory containing man pages of the application by running the following command:

```
# export MANPATH=$MANPATH:<path to the man pages directory>
```

### 3.3.1. Location of files on a server running Linux

The default locations of Kaspersky Anti-Virus files on a server running Linux OS are as follows:

`/etc/opt/kaspersky/kav4lms.conf` – main configuration file of application;

`/etc/opt/kaspersky/kav4lms/` – directory containing the Kaspersky Anti-Virus configuration files:

`groups.d/` - directory containing the groups' configuration files;

`default.conf` – configuration file, containing the default group's settings;

`locale.d/strings.en` – file, containing strings, used by the application;

`profiles/` – directory containing predefined configuration profiles:

`default_recommended/` – directory containing the default configuration files;



- high\_overall\_security/* – directory containing the configuration files for high security profile;
- high\_scan\_speed/* – directory containing the configuration files for high scan speed profile;
- templates/* – directory containing notification templates;
- templates-admin/* – directory containing administrator's notifications templates;
- kav4lms.conf* – the application's main configuration file;
- /opt/kaspersky/kav4lms/* – main directory of Kaspersky Anti-Virus, containing:
  - bin/* – a directory that contains executable files of all Kaspersky Anti-Virus components:
    - kav4lms-cmd* – executable file of the command line tool;
    - kav4lms-setup.sh* – the application's setup script;
    - kav4lms-kavscanner* – executable file of the file system scan component;
    - kav4lms-licensemanager* – executable file of the keys management component;
    - kav4lms-keepup2date* – executable file of the updater component;
  - sbin/* – a directory that contains executable files of application's services;
  - lib/* - directory containing Kaspersky Anti-Virus library files;
    - bin/avbasestest* – utility validating downloaded updates to the anti-virus databases used by the *kav4lms-keepup2date* component;
  - share/doc/* – directory containing license agreement and deployment documentation;
  - share/man/* – directory containing manual files;
  - share/scripts/* – directory containing the application's scripts;
  - share/snmp-mibs/* – directory containing the Kaspersky Anti-Virus MIB;
  - share/webmin/* – directory containing plug-in to Webmin application;
- /etc/init.d/* – directory containing control scripts for application services:
  - kav4lms* – control script for the central service of the application;
  - kav4lms-filters* – control script for Kaspersky Anti-Virus filter;
- /var/opt/kaspersky/kav4lms/* - directory containing variable data of Kaspersky Anti-Virus:
  - backup/* – directory containing messages' backup copies and information files;
  - bases/* – directory containing anti-virus databases;
  - bases.backup/* – directory containing backup copy of the anti-virus databases;

*licenses/* – directory containing key files;  
*nqueue/* – directory containing the mail queue files;  
*patches/* – directory containing application modules' updates;  
*stats/* – directory containing statistics files;  
*updater/* – directory containing information file about the last update.

**Warning!**

Linux-related paths are used further in this document.

### 3.3.2. Location of files on a server running FreeBSD

The default locations of Kaspersky Anti-Virus files on a server running FreeBSD OS are as follows:

*/usr/local/etc/kaspersky/kav4lms.conf* – main configuration file of application;  
*/usr/local/etc/kaspersky/kav4lms/* – directory containing the Kaspersky Anti-Virus configuration files:  
*groups.d/* - directory containing the groups' configuration files;  
*default.conf* – configuration file, containing the default group's settings;  
*locale.d/strings.en* – file containing strings used by the application;  
*profiles/* – directory containing predefined configuration profiles:  
*default\_recommended/* – directory containing the default configuration files;  
*high\_overall\_security/* – directory containing the configuration files of the high security profile;  
*high\_scan\_speed/* – directory containing the configuration files of the high speed profile;  
*templates/* – directory containing notification templates;  
*templates-admin/* – directory containing administrator's notifications templates;  
*kav4lms.conf* – the application's main configuration file.  
*/usr/local/bin/* – a directory that contains executable files of all Kaspersky Anti-Virus components:  
*kav4lms-cmd* – executable file of the command line tool;  
*kav4lms-setup.sh* – the application's setup script;  
*kav4lms-kavscanner* – executable file of the file system scan component;

*kav4lms-licensemanager* – executable file of the keys management component;

*kav4lms-keepup2date* – executable file of the updater component;

*/usr/local/sbin/* – a directory that contains executable files of application's services;

*/usr/local/etc/rc.d/* – directory containing control scripts for application services:

*kav4lms.sh* – control script for the central service of the application;

*kav4lms-filters.sh* – control script for Kaspersky Anti-Virus filter;

*/usr/local/lib/kaspersky/kav4lms/* - directory containing Kaspersky Anti-Virus library files;

*/usr/local/libexec/kaspersky/kav4lms/avbasestest* – utility validating downloaded updates to the anti-virus databases used by the *kav4lms-keepup2date* component;

*/usr/local/share/doc/kav4lms/* – directory containing license agreement and deployment documentation;

*/usr/local/man/* – directory containing manual files;

*/usr/local/share/kav4lms/scripts/* – directory containing the application's scripts;

*/usr/local/share/kav4lms/snmp-mibs/* – directory containing the Kaspersky Anti-Virus MIB;

*/usr/local/share/kav4lms/webmin/* – directory containing plug-in to Webmin application;

*/var/db/kaspersky/kav4lms/* - directory containing variable data of Kaspersky Anti-Virus:

*backup/* – directory containing messages' backup copies and information files;

*bases/* – directory containing anti-virus databases;

*bases.backup/* – directory containing backup copy of the anti-virus databases;

*licenses/* – directory containing key files;

*nqueue/* – directory containing the mail queue files;

*patches/* – directory containing the application modules' updates;

*stats/* – directory containing statistics files;

*updater/* – directory containing information file about the last update.

## 3.4. Post-installation setup

Immediately after the application files have been copied to your server, the system configuration process will start. The configuration procedure will either be

started automatically or, if the package manager (such as *rpm*) does not allow the use of interactive scripts, you will have to initiate it manually.

*To start product configuration manually, enter the following in the command line:*

In Linux:

```
# /opt/kaspersky/kav4lms/lib/bin/setup/postinstall.pl
```

In FreeBSD:

```
# /usr/local/libexec/kaspersky/kav4lms/setup/postinstall.pl
```

You will see an offer to perform the following operations:

1. If the application finds on the computer configuration files of Kaspersky Anti-Virus 5.5 for Linux Mail Server or Kaspersky Anti-Virus 5.6 for Sendmail with Milter API, it will offer during this step to choose the file for conversion and saving in the format of the current product version. If you select one of the files, you will be offered to replace the default configuration file included into the distribution package with this restored and converted file.

To replace the configuration file from the distribution package with the restored file, enter **yes** as your response. To cancel the replacement, enter **no**.

By default converted configuration files are saved in the following directories:

```
kav4mailservers -  
/etc/opt/kaspersky/kav4lms/profiles/kav4mailservers5.  
5-converted  
kavmilter -  
/etc/opt/kaspersky/kav4lms/profiles/kavmilter5.6-  
converted
```

2. Specify the path to the key file.

Please note, that if the product key is not installed, the anti-virus will not update its databases and create the protected domains list during installation. In that case you will have to perform those steps manually after key installation.

3. Specify the parameters of the proxy server used for connection to the Internet in the following format:

```
http://<IP-proxy_server_address>:<port>
```

or

```
http://<user_name>:<password>@<proxy_server_IP_address>:<port>
```

if the proxy server requires authentication.

If no proxy server is used to connect to the Internet, enter **no** as your response.

The *kav4lms-keepup2date* update component will use the value to connect to the source of updates.

4. Update the anti-virus databases. To do that, enter **yes** as your response. If you wish to skip updates during this step, enter **no**. You will be able to run the update procedure later using the *kav4lms-keepup2date* component (see 7.2 on p. 78 for details).

**Note:**

The anti-virus databases can only be updated with the installed product key.

5. Configure automatic updates of the anti-virus databases. To do that, enter **yes** as your response. To skip configuration of automatic updates during this step, enter **no**. You will be able to configure updates later using the *kav4lms-setup* component (see 7.1 on p. 77) or manually (see 10.2 on p. 95 for details).

**Warning!**

In case of product integration with gmail automatic updates should be configured as follows:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-setup.sh \  
--install-cron=updater --user=root
```

6. Install the webmin module for management of Kaspersky Anti-Virus within the web-based interface of Webmin.

The remote management plug-in will only be installed provided that Webmin is installed in the default directory. After plug-in installation you will see appropriate guidelines for configuration of its interaction with the application.

Enter **yes** to install the Webmin module or **no** to cancel its installation.

7. Determine the list of domains whose e-mail traffic will be protected against viruses. The default value is **localhost, localhost.localdomain**. To use it, press **Enter**.

To specify the list of domains manually, enter them in the command line. You can define several comma-delimited values; masks and regu-

lar expressions are supported. Dots should be escaped by slash symbol.

E.g.:

```
re:.*\example\.com
```

8. Integrate Kaspersky Anti-Virus with MTA. You can agree to the default suggested method of integration with the MTA found on the computer or cancel integration and perform it manually. Please see Chapter 4 on p. 28 for a detailed description of integration with MTA.

By default, the post-queue integration is used for Exim and Postfix mail systems (see 4.1.1 on p. 29 and 4.2.1 on p. 35).

### **Warning!**

During automatic integration with Sendmail the script always tries to modify the `.mc` file because any subsequent update will preserve the entered changes. If the `.mc` file contains include directions referring to `.mc` files that do not exist, then such file cannot be used for integration of Kaspersky Anti-Virus. In such case install the **sendmail-cf** package for integration using `.cf` file.

If the `.mc` file cannot be used for integration of the application, then `.cf` file will be used for that purpose.

## **3.5. Configuration of permission rules in SELinux and AppArmor systems**

To create a SELinux module with the rules necessary for Kaspersky Anti-Virus operation, perform the following steps after application setup and its integration with the e-mail system:

1. Switch SELinux into permissive mode:
2. Send one or more test messages and make sure that they have passed anti-virus scanning and have been delivered to recipients.
3. Create a rules module based on the blocking records:

For Fedora:

```
# audit2allow -l -M kav4lms -i /var/log/messages
```

For RHEL:

```
# audit2allow -l -M kav4lms -i\  
/var/log/audit/audit.log
```

4. Load the resulting rules module:

```
# semodule -i kav4lms.pp
```

5. Switch SELinux into enforcement mode:

```
# setenforce Enforcing
```

If new audit messages pertaining to Kaspersky Anti-Virus appear, the rules module file should be updated:

For Fedora:

```
# audit2allow -l -M kav4lms -i /var/log/messages  
# semodule -u kav4lms.pp
```

For RHEL:

```
# audit2allow -l -M kav4lms -i /var/log/audit/audit.log  
# semodule -u kav4lms.pp
```

For additional information please refer to:

- **RedHat Enterprise Linux:** «Red Hat Enterprise Linux Deployment Guide», chapter «44. Security and SELinux».
- **Fedora:** Fedora SELinux Project Pages.
- **Debian GNU/Linux:** «Configuring the SELinux Policy» manual from the «Documentation for Security-Enhanced Linux» selinux-doc package.

To update AppArmor profiles necessary for operation of Kaspersky Anti-Virus, perform the following steps after application setup and its integration with the e-mail system:

1. Switch all application rules into complain mode:

```
# aa-complain /etc/apparmor.d/*  
# /etc/init.d/apparmor reload
```

2. Restart the e-mail system:

```
# /etc/init.d/postfix restart
```

3. Restart kav4lms and kav4lms-filters:

```
# /etc/init.d/kav4lms restart  
# /etc/init.d/kav4lms-filters restart
```

4. Send one or more test messages and make sure that they have passed anti-virus scanning and have been delivered to recipients.
5. Launch the profiles updating utility:

```
# aa-logprof
```
6. Reload AppArmor rules:

```
# /etc/init.d/apparmor reload
```
7. Switch all application rules into enforcement mode:

```
# aa-enforce /etc/apparmor.d/*  
# /etc/init.d/apparmor reload
```

If new audit messages pertaining to Kaspersky Anti-Virus appear, the steps 5 and 6 should be repeated.

For additional information please refer to:

- **openSUSE and SUSE Linux Enterprise Server:** «Novell AppArmor Quick Start», «Novell AppArmor Administration Guide».
- **Ubuntu:** «Ubuntu Server Guide», chapter «8. Security».

## 3.6. Installing the Webmin module to manage Kaspersky Anti-Virus

The activity of Kaspersky Anti-Virus can be controlled remotely via a web browser using Webmin.

Webmin is a program which simplifies the administration of Linux/Unix systems. The software has a modular structure, and supports connection of new or customized modules. Additional information about Webmin can be obtained, and its distribution package downloaded, from the official program web site at: [www.webmin.com](http://www.webmin.com).


The distribution package of Kaspersky Anti-Virus contains a Webmin module that can either be connected during the application's post-installation configuration (see 3.4 on p. 19) if the system already has Webmin installed, or at any time later after Webmin is installed.

The following part of this manual contains a detailed description of the procedure necessary to connect the Webmin module for administration of Kaspersky Anti-Virus.



If default settings were selected during Webmin installation, then you can access the program after setup in a web browser connecting to port 10000 via HTTP/HTTPS.

To install the Webmin module for Kaspersky Anti-Virus management:

1. Use your web browser to access Webmin with administrator privileges.
2. Select the **Webmin Configuration** tab in the program menu, and then proceed to the **Webmin Modules** section.
3. Select the **From Local File** option in the **Install Module** section and click  (see Figure 1).

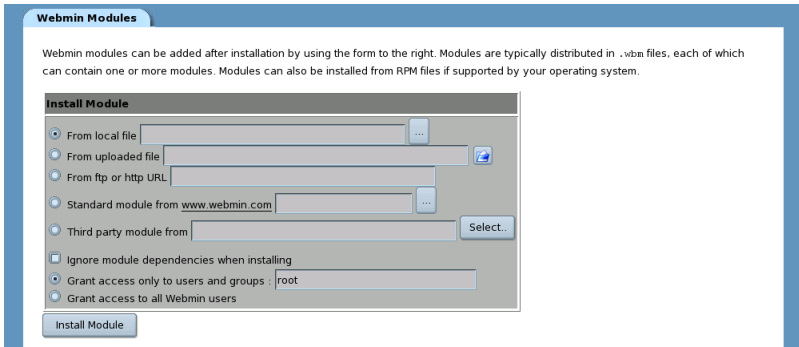


Figure 1. **Install Module** section

4. Select the path to the Webmin module of the product and click **OK**.

**Note:**

The Webmin module is the file *mailgw.wbm*, which is installed by default in the */opt/kaspersky/kav4lms/share/webmin/* directory (for Linux distributions), or the */usr/local/share/kav4lms/webmin/* directory (for FreeBSD distributions).

A message on the display will confirm the successful installation of the Webmin module.

You can access the settings of Kaspersky Anti-Virus by clicking its icon within the **Others** tab (see Figure 2).



Figure 2. The icon of Kaspersky Anti-Virus in the **Others** tab

## 3.7. Application removal

Removal of Kaspersky Anti-Virus from server requires superuser (**root**) privileges. If you have no such privileges when you start the removal procedure, you will have to log on as **root** first.

### Warning!

The removal procedure will stop the application without additional user participation!

During removal the application will be stopped, its files and directories created at product installation will be deleted. However, files and directories created or modified by the administrator (configuration file of the application, configuration files of groups, template notification files, backup directories, key file), will be preserved.

The application removal procedure can be initiated using different methods depending upon the system package manager. Let us examine those methods closely.

*In order to remove Kaspersky Anti-Virus installed from a rpm package, enter the following text in the command line:*

```
# rpm -e <package_name>
```

*In order to remove Kaspersky Anti-Virus installed from a deb package, enter the following in the command line:*

```
# dpkg -P <package_name>
```

if you wish to remove the application together with its configuration files, or:

```
# dpkg -r <package_name>
```

if you wish to uninstall the application but keep its configuration files.

*In order to remove Kaspersky Anti-Virus installed from a pkg package, enter the following in the command line:*

```
# pkg_delete <package_name>
```

A message on the display will confirm the successful removal of the application.

If a plug-in for remote management of the application (Webmin module) was installed, it must be removed manually using standard Webmin tools.

# CHAPTER 4. INTEGRATION WITH MTA

After installation the Anti-Virus must be integrated with the host e-mail system. To do that, the parameters in the configuration files of the application and MTA have to be modified. You can perform integration using the product configuration script included into the distribution package (see 3.4 on p. 19 and 10.2 on p. 95), or modify the configuration files of Kaspersky Anti-Virus and MTA manually.

For Exim and Postfix the Anti-Virus supports both pre-queue and post-queue integration. In case of pre-queue integration messages are transferred for analysis before their addition to MTA queue, post-queue integration means that they are checked after addition to the mail queue.

## Note:

MTA does not allow mail rejection if post-queue integration is used. However, if **reject** is selected as the action over objects in Kaspersky Anti-Virus settings, the sender will receive a notification about message rejection. Notification text is defined by the **RejectReply** option in the **[kav4lms: groups. <group\_name>.settings]** section of the group configuration file.

The sockets used for data exchange between MTA, filter and the central service of Kaspersky Anti-Virus are assigned using the following rules:

- `inet:<port>@<ip_address>` – for a network socket
- `local:<socket_path>` – for a local socket.

## Warning!

Two rules must be observed while using a socket:

- The port number, which is a part of network socket definition, must be greater than 1024.
- Both filter and central services must have sufficient privileges to access the local socket used.

## 4.1. Integration with Exim

The Anti-Virus can use two methods for integration with Exim:

- **post-queue integration using modification of routers:** all e-mail traffic passing the protected server will be transferred for scanning after its addition to the MTA queue (post-queue filtering).
- **pre-queue integration using dynamically loaded library:** messages will be transferred for scanning before their addition to the MTA queue (pre-queue filtering).

### 4.1.1. Post-queue integration using modification of routers

Integration using modification of routers implies that messages will be sent for scanning from all e-mail transfers. To accomplish that, **kav4lms\_filter** must be specified as the value of the **pass\_router** option for each Exim router.

In case of post-queue integration correct e-mail transfer to the Anti-Virus and its return to MTA requires observance of the following conditions:

1. The filter must be configured to intercept messages from MTA. The endpoint of the «filter - MTA» connection is the socket defined by the **FilterSocket** option in the **[kav4lms:filter.settings]** section of the main application configuration file.
2. The filter must pass messages over for scanning to the central application service. The endpoint of the «filter – central service» connection is the socket defined by the **ServiceSocket** option in the **[kav4lms:server.settings]** section of the main application configuration file.

#### Warning!

In case of post-queue integration with Exim the **FilterSocket**, **ServiceSocket** and **ForwardSocket** options must point to the network socket.

3. The filter must return messages to the MTA. The endpoint of the «application – MTA» connection is the socket defined by the **ForwardSocket** option in the **[kav4lms:filter.settings]** section of the main application configuration file.

To integrate Kaspersky Anti-Virus with Exim using the application configuration script:

run the following command:

in Linux:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-setup.sh \  
--install-filter=exim
```

in FreeBSD:

```
# /usr/local/bin/kav4lms-setup.sh \  
--install-filter=exim
```

To integrate the application with Exim manually:

1. Make a backup copy of Exim configuration files.
2. Add the following lines in the **main configuration settings** section of the Exim configuration file:

```
#kav4lms-filter-begin-1  
local_interfaces=0.0.0.0.25:<forward_socket_ip>\  
<forward_socket_port_number>  
#kav4lms-filter-end-1
```

where `<forward_socket_ip>.<forward_socket_port_number>` is the IP-address and port of the socket, to which mail is routed by application after checking.

3. Add the following lines to the **routers** section of the Exim configuration file:

```
#kav4lms-filter-begin-2  
kav4lms_dnslookup:  
    driver = dnslookup  
    domains = ! +local_domains  
    ignore_target_hosts = 0.0.0.0 : 127.0.0.0/8  
    verify_only  
    pass_router = kav4lms_filter  
    no_more
```

```
kav4lms_system_aliases:  
    driver = redirect  
    allow_fail  
    allow_defer
```

```

    data = ${lookup{$local_part}lsearch{/etc/aliases}}
    verify_only
    pass_router = kav4lms_filter

kav4lms_localuser:
    driver = accept
    check_local_user
    verify_only
    pass_router = kav4lms_filter

failed_address_router:
    driver = redirect
    verify_only
    condition = "{0}"
    allow_fail
    data = :fail: Failed to deliver to address
    no_more

kav4lms_filter:
    driver = manualroute
    condition = "${if or {{eq {$interface_port}\  
{<forward_socket_port_number>}} \  
    {eq {$received_protocol}{spam-scanned}} \  
    }}{0}{1}}"
    transport = kav4lms_filter
    route_list = "* localhost byname"
    self = send
#kav4lms-filter-end-2

```

where `<forward_socket_port_number>` is the number of port, to which mail is routed by application after checking.

4. Add the following lines to the Exim's transports definition section:

```

#kav4lms-filter-begin-3
kav4lms_filter:
    driver = smtp
    port = <filter_socket_port_number>
    delay_after_cutoff = false

```

```
allow_localhost
#kav4lms-filter-end-3
```

where `<filter_socket_port_number>` is the number of port, on which the application's filter service is listening.

5. Set the **ForwardSocket** parameter to `<forward_socket_ip>.<forward_socket_port_number>` value from step 2. The **ForwardSocket** parameter resides in the **[kav4lms:filter.settings]** section of the *kav4lms.conf* configuration file.
6. Stop the *kav4lms-filter* service.
7. Add the following line to the **[1043]** section of the */var/opt/kaspersky/applications.setup* (in Linux) */var/db/kaspersky/applications.setup* (in FreeBSD) file:

```
FILTER_SERVICE=true
FILTER_PROGRAM=kav4lms-filter
```
8. Start the *kav4lms-filter* service.
9. Restart *exim*.

## 4.1.2. Pre-queue integration using dynamically loaded library

The filter must pass messages for scanning to the central service of the application. The endpoint of the «filter – central service» connection is the socket defined by the **ServiceSocket** option in the **[kav4lms:server.settings]** section of the main product configuration file.

*To integrated Kaspersky Anti-Virus with Exim using the application configuration script:*

run the following command:

in Linux:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-setup.sh \
--install-filter=exim-dlfunc
```

in FreeBSD:

```
# /usr/local/bin/kav4lms-setup.sh \
--install-filter=exim-dlfunc
```



To integrate Kaspersky Anti-Virus with Exim manually:

1. Make sure that Exim supports the dlfunc content filtration function. To do that, run the following command:

```
exim -bV
```

Positive response will look like:

```
Expand_dlfunc
```

2. Make a backup copy of Exim configuration files.
3. Add the following lines in the **main configuration settings** section of Exim configuration file:

```
#kav4lms-filter-begin
acl_smtp_data = acl_check_data
#kav4lms-filter-end
```

4. Add the following lines in the **ACL** section of Exim configuration file:

```
acl_check_data:
#kav4lms-dlfunc-begin
warn set acl_m0 = \
${dlfunc{<libkavexim.so>}{kav}{<socket>}\
{/var/tmp/.kav4lms-exim}}
accept condition = ${if match{$acl_m0}{\N^kav4lms: \
continue\N}{yes}{no}}
logwrite = kav4lms returned continue
deny condition = ${if match{$acl_m0}{\N^kav4lms: \
reject.*\N}{yes}{no}}
logwrite = kav4lms returned reject
message = Kaspersky Anti-Virus rejected the mail
discard condition = ${if match{$acl_m0}\
{\N^kav4lms: drop.*\N}{yes}{no}}
logwrite = kav4lms returned drop
message = Kaspersky Anti-Virus dropped the mail
defer condition = ${if match{$acl_m0}\
{\N^kav4lms: temporary failure.*\N}{yes}{no}}
logwrite = kav4lms returned temporary failure
message = Kaspersky Anti-Virus returned \
temporary failure
accept
#kav4lms-dlfunc-end
```

where <socket> stands for the socket used for communication between the filter and central service of Kaspersky Anti-Virus defined by the **ServiceSocket** option in the **[kav4lms:server.settings]** section of the main Kaspersky Anti-Virus configuration file; <libkavexim.so> - path to the *libkavexim.so* library:

in 32-bit Linux distributives:

```
/opt/kaspersky/kav4lms/lib/libkavexim.so
```

in 64-bit Linux distributives:

```
/opt/kaspersky/kav4lms/lib64/libkavexim.so
```

in FreeBSD:

```
/usr/local/lib/kaspersky/kav4lms/libkavexim.so
```

5. Stop the *kav4lms-filter* service.
6. Add the following line to the **[1043]** section of the */var/opt/kaspersky/applications.setup* (in Linux) */var/db/kaspersky/applications.setup* (in FreeBSD) file:

in Linux:

```
FILTER_SERVICE=false
FILTER_PROGRAM=/opt/kaspersky/kav4lms/lib/libkavexim\
.so
```

in FreeBSD:

```
FILTER_SERVICE=false
FILTER_PROGRAM=/usr/local/lib/kaspersky/kav4lms/\
libkavexim.so
```

7. Restart *exim*.

## 4.2. Integration with Postfix

The Anti-Virus can use three methods for integration with Postfix:

- **post-queue integration:** all mail traffic going through a protected server is transferred for scanning after being added to the mail system queue;

- **pre-queue integration:** messages are transferred for scanning before being added to the mail system queue;
- **integration with Milter:** messages are transferred for scanning using the Milter program interface.

## 4.2.1. Post-queue integration

Correct e-mail transfer to the Anti-Virus and its return to MTA requires observance of the following conditions:

1. The filter must be configured to intercept messages from MTA. The endpoint of the «filter - MTA» connection is the socket defined by the **FilterSocket** option in the **[kav4lms:filter.settings]** section of the main application configuration file.
2. The filter must pass messages over for scanning to the central application service. The endpoint of the «filter – central service» connection is the socket defined by the **ServiceSocket** option in the **[kav4lms:server.settings]** section of the main application configuration file.

### Warning!

In case of integration with Postfix the **FilterSocket**, **ServiceSocket** and **ForwardSocket** options can point to a network or local socket.

3. The filter must return messages to the MTA. The endpoint of the «application – MTA» connection is the socket defined by the **ForwardSocket** option in the **[kav4lms:filter.settings]** section of the main application configuration file.

### Note:

While copying strings from the manual to the Postfix configuration file delete the «\» symbols and the line breaks that follow.

*To integrate Kaspersky Anti-Virus with Postfix using the application configuration script:*

run the command:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-setup.sh \  
--install-filter=postfix
```

in FreeBSD:

```
# /usr/local/bin/kav4lms-setup.sh \  
--install-filter=postfix
```

To integrate application with Postfix manually:

1. Add the following lines to *master.cf* file:

```
#kav4lms-filter-begin
kav4lms_filter      unix      -      -      n\
      -      10      smtp
      -o smtp_send_xforward_command=yes
<forward_socket_ip_address>:<forward_socket_port>\
      inet      n      -      n      -      10\
      smtpd
      -o content_filter=
      -o receive_override_options=\
no_unknown_recipient_checks,no_header_body_checks,\
no_address_mappings
```

**Note:**

If local sockets are used with Postfix 2.3 or higher, also add to the line above the option 'no\_milters', i.e.:

```
-o receive_override_options=\
no_unknown_recipient_checks,no_header_body_checks,\
no_address_mappings,no_milters

-o smtpd_helo_restrictions=
-o smtpd_client_restrictions=
-o smtpd_sender_restrictions=
-o smtpd_recipient_restrictions=\
permit_mynetworks,reject
-o mynetworks=127.0.0.0/8, [::1]/128
-o smtpd_authorized_xforward_hosts=\
127.0.0.0/8, [::1]/128
#kav4lms-filter-end
```

where `<forward_socket_ip_address>:<forward_socket_port>` is the address and port of the socket, to which mail is forwarded after checking by application.

2. Add the following lines to *main.cf* file:

```
#kav4lms-filter-begin
content_filter = \
kav4lms_filter:<filter_socket_ip_address>:\
<filter_socket_port>
#kav4lms-filter-end
```

where `<filter_socket_ip_address>:<filter_socket_port>` is the address and port of the socket, where the filter process is listening.

3. Stop the *kav4lms-filter* service.
4. Add the following line to the **[1043]** section of the */var/opt/kaspersky/applications.setup* (in Linux) */var/db/kaspersky/applications.setup* (in FreeBSD) file:

```
FILTER_SERVICE=true
FILTER_PROGRAM=kav4lms-filter
```

5. Start the *kav4lms-filter* service.
6. Restart Postfix.

## 4.2.2. Pre-queue integration

Correct e-mail transfer to the Anti-Virus and its return to MTA requires observance of the following conditions:

1. The filter must be configured to intercept messages from MTA. The endpoint of the «filter - MTA» connection is the socket defined by the **FilterSocket** option in the **[kav4lms:filter.settings]** section of the main application configuration file.
2. The filter must pass messages over for scanning to the central application service. The endpoint of the «filter – central service» connection is the socket defined by the **ServiceSocket** option in the **[kav4lms:server.settings]** section of the main application configuration file.

### Warning!

In case of integration with Postfix the **FilterSocket**, **ServiceSocket** and **ForwardSocket** options can point to a network or local socket.

3. The filter must return messages to the MTA. The endpoint of the «application – MTA» connection is the socket defined by the **ForwardSocket** option in the **[kav4lms:filter.settings]** section of the main application configuration file.

### Note:

While copying strings from the manual to the Postfix configuration file delete the «\» symbols and the line breaks that follow.

To integrate Kaspersky Anti-Virus with Postfix using the application configuration script:

run the command:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-setup.sh \
--install-filter=postfix-prequeue
```

in FreeBSD:

```
# /usr/local/bin/kav4lms-setup.sh \
--install-filter=postfix-prequeue
```

To integrate application with Postfix manually:

1. Add the following lines to *master.cf* file:

```
#kav4lms-prequeue-begin
kav4lms_filter      unix      -      -      n\
-      10      smtp
-o smtp_send_xforward_command=yes
<forward_socket_ip_address>:<forward_socket_port>\
inet      n      -      n      -      10\
smtpd
-o content_filter=
-o receive_override_options=\
no_unknown_recipient_checks,no_header_body_checks,\
no_address_mappings
```

**Note:**

If local sockets are used with Postfix 2.3 or higher, also add to the line above the option 'no\_milters', i.e.:

```
-o receive_override_options=\
no_unknown_recipient_checks,no_header_body_checks,\
no_address_mappings,no_milters

-o smtpd_helo_restrictions=
-o smtpd_client_restrictions=
-o smtpd_sender_restrictions=
-o smtpd_recipient_restrictions=\
permit_mynetworks,reject
-o mynetworks=127.0.0.0/8,[::1]/128
-o smtpd_authorized_xforward_hosts=\
127.0.0.0/8,[::1]/128
#kav4lms-prequeue-end
```

where `<forward_socket_ip_address>:<forward_socket_port>` is the address and port of the socket, to which mail is forwarded after checking by application.

2. Add the following lines to *master.cf* file:

```
smtp inet n - n - 20 smtpd
add the parameter
#kav4lms-prequeue-begin
-o smtpd_proxy_filter=:<filter_socket_port>
#kav4lms-prequeue-end
```

3. Stop the *kav4lms-filter* service.
4. Add the following line to the **[1043]** section of the */var/opt/kaspersky/applications.setup* (in Linux) */var/db/kaspersky/applications.setup* (in FreeBSD) file:

```
FILTER_SERVICE=true
FILTER_PROGRAM=kav4lms-filter
```

5. Start the *kav4lms-filter* service.
6. Restart Postfix.

### 4.2.3. Integration with Milter

Correct e-mail transfer to the Anti-Virus and its return to MTA requires observance of the following conditions:

1. The filter must be configured to intercept messages from MTA. The endpoint of the «filter - MTA» connection is the socket defined by the **FilterSocket** option in the **[kav4lms:filter.settings]** section of the main application configuration file.
2. The filter must pass messages over for scanning to the central application service. The endpoint of the «filter – central service» connection is the socket defined by the **ServiceSocket** option in the **[kav4lms:server.settings]** section of the main application configuration file.

#### Warning!

In case of integration with Postfix the **FilterSocket** and **ServiceSocket** options can point to a network or local socket.

**Note:**

While copying strings from the manual to the Postfix configuration file delete the «\» symbols and the line breaks that follow.

To integrate Kaspersky Anti-Virus with Postfix using the application configuration script:

run the command:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-setup.sh \
--install-filter=postfix-milter
```

in FreeBSD:

```
# /usr/local/bin/kav4lms-setup.sh \
--install-filter=postfix-milter
```

To integrate application with Postfix manually:

1. Add the following lines to *main.cf* file:

```
smtpd_milters = inet:127.0.0.1:10025,
#kav4lms-milter-begin
milter_connect_macros = j _ {daemon_name} {if_name} \
{if_addr}
milter_helo_macros = {tls_version} {cipher} \
{cipher_bits} {cert_subject} {cert_issuer}
milter_mail_macros = i {auth_type} {auth_authen} \
{auth_ssf} {auth_author} {mail_mailer} {mail_host} \
{mail_addr}
milter_rcpt_macros = {rcpt_mailer} {rcpt_host} \
{rcpt_addr}
milter_default_action = tempfail
milter_protocol = 3
milter_connect_timeout=180
milter_command_timeout=180
milter_content_timeout=600
#kav4lms-milter-end
```

2. Stop the *kav4lms-milter* service.
3. Add the following line to the **[1043]** section of the */var/opt/kaspersky/applications.setup* (in Linux) */var/db/kaspersky/applications.setup* (in FreeBSD) file:

```
FILTER_SERVICE=true
FILTER_PROGRAM=kav4lms-milter
```



4. Start the *kav4lms-milter* service.
5. Restart Postfix.

## 4.3. Integration with qmail

The qmail MTA does not provide support for filtering extensions. Filtering is implemented by the */opt/kaspersky/kav4lms/lib/bin/kav4lms-qmail* (*/usr/local/libexec/kaspersky/kav4lms/kav4lms-qmail* for FreeBSD) binary, provided with the application, which replaces the original *qmail-queue* binary. The replacing file implements filtering and passes the mail traffic to the original *qmail-queue* for delivery. Messages are transferred for analysis before their addition to MTA queue (pre-queue filtration).

### Warning!

In case of integration with qmail the **ServiceSocket** option can point to a network or local socket.

To integrate Kaspersky Anti-Virus with qmail using the application configuration script:

run the command:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-setup.sh \  
--install-filter=qmail
```

in FreeBSD:

```
# /usr/local/bin/kav4lms-setup.sh \  
--install-filter=qmail
```

To integrate the application with qmail manually:

1. Rename the *qmail-queue* file in the */var/qmail/bin* directory to *qmail-queue-real*.
2. Copy the */opt/kaspersky/kav4lms/lib/bin/kav4lms-qmail* (*/usr/local/libexec/kaspersky/kav4lms/kav4lms-qmail* for FreeBSD) file to the */var/qmail/bin* directory and rename it to *qmail-queue*.
3. Set the following permissions for *qmail-queue* and *qmail-queue-real* files:  

```
-rws-x--x 1 qmailq qmail
```
4. Stop the *kav4lms-filter* service.
5. Change the owner and group to *qmailq:qmail* for the following directories and their contents:

- for Linux:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-setup.sh \  
--switch-credentials=qmailq,qmail
```

- for FreeBSD:

```
# /usr/local/bin/kav4lms-setup.sh \  
--switch-credentials=qmailq,qmail
```

6. Add the following line to the **[1043]** section of the */var/opt/kaspersky/applications.setup* (in Linux) */var/db/kaspersky/applications.setup* (in FreeBSD) file:

in Linux:

```
FILTER_SERVICE=false  
FILTER_PROGRAM=/opt/kaspersky/kav4lms/lib/bin\  
/kav4lms-qmail
```

in FreeBSD:

```
FILTER_SERVICE=false  
FILTER_PROGRAM=/usr/local/libexec/kaspersky/kav4lms\  
/kav4lms-qmail
```

7. Restart qmail.

## 4.4. Integration with Sendmail

Sendmail provides the Milter API to implement integration with custom filters. The mail traffic should be passed from Sendmail to Kaspersky Anti-Virus and back using the Milter interface calls. Messages are transferred for analysis before their addition to MTA queue (pre-queue integration).

As a rule, in case of product integration with Sendmail changes are made to the MTA configuration file in *mc* format, the *cf* file changes automatically. If such functionality is not supported, then after modification of the appropriate *mc* file, the corresponding *cf* file should be modified, too.

### Note:

If you enter changes into *cf* file only, they will be lost the next time when generation of the *cf* file from the *mc* file is initiated.

**Warning!**

In case of integration with Sendmail the **FilterSocket** and **ServiceSocket** options can point to a network or local socket.

## 4.4.1. Integration with Sendmail using *.cf* file

To integrate Kaspersky Anti-Virus with Sendmail using the application configuration script:

run the command:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-setup.sh \
--install-filter=sendmail-milter
```

in FreeBSD:

```
# /usr/local/bin/kav4lms-setup.sh \
--install-filter=sendmail-milter
```

To integrate the application with Sendmail manually:

1. Make backup copy of *sendmail.cf* file.
2. Add the following strings to the *sendmail.cf* file:

```
#kav4lms-milter-begin-filter
O InputMailFilters=kav4lms_filter
O Milter.macros.connect=j, _, {daemon_name}, \
{if_name}, {if_addr}
O Milter.macros.helo={tls_version}, {cipher}, \
{cipher_bits}, {cert_subject}, {cert_issuer}
O Milter.macros.envfrom=i, {auth_type}, \
{auth_authen}, {auth_ssf}, {auth_author}, \
{mail_mailer}, {mail_host}, {mail_addr}
O Milter.macros.envrcpt={rcpt_mailer}, {rcpt_host}, \
{rcpt_addr}
#kav4lms-milter-end-filter
```

3. Add the following lines to the *sendmail.cf* file:
  - a) if integrating via network socket:

```
#kav4lms-milter-begin-socket
```

```
Xkav4lms_filter,
S=inet:<filter_port>@<filter_address>,F=T,\
T=S:3m;R:5m;E:10m
#kav4lms-milter-end-socket
```

where `<filter_port>` is the port number of the network socket, where the filter service is listening, `<filter_address>` is the name or IP-address of the server, where filter service is running.

- b) If the local socket is required for connection, change the socket definition section to the following:

```
#kav4lms-milter-begin-socket
Xkav4lms_filter,
S=unix:<filter_socket_file_path>,F=T,T=S:3m;\
R:5m;E:10m
#kav4lms-milter-end-socket
```

where `<socket_file_path>` is the path to the local socket.

4. Stop the *kav4lms-milter* service.
5. Add the following line to the **[1043]** section of the */var/opt/kaspersky/applications.setup* (in Linux) */var/db/kaspersky/applications.setup* (in FreeBSD) file:

```
FILTER_SERVICE=true
FILTER_PROGRAM=kav4lms-milter
```

6. Start the *kav4lms-milter* service.
7. Restart Sendmail.

## 4.4.2. Integration with Sendmail using *.mc* file

To integrate the application with Sendmail via the *.mc* file:

1. Make backup copy of *.mc* file.
2. Add the following strings to the *.mc* file:

```
dnl kav4lms-milter-begin dnl
define(`_FFR_MILTER', `true')dnl
INPUT_MAIL_FILTER(`kav4lms_filter',\
`S=inet:10025@127.0.0.1,F=T,T=S:3m;R:5m;E:10m')dnl
dnl kav4lms-milter-end dnl
```

3. Compile the *.cf* configuration file according to your operation system's rules.
4. Stop the *kav4lms-filter* service.
5. Add the following line to the **[1043]** section of the */var/opt/kaspersky/applications.setup* (in Linux)  
*/var/db/kaspersky/applications.setup* (in FreeBSD) file:

```
FILTER_SERVICE=true  
FILTER_PROGRAM=kav4lms-milter
```

6. Start the *kav4lms-filter* service.
7. Restart Sendmail.

# CHAPTER 5. ANTI-VIRUS

## PROTECTION OF E-MAIL

### 5.1. Setting up groups

A group consists of multiple addresses of senders and recipients whose messages are processed using the same settings of Kaspersky Anti-Virus.

Custom mail scanning settings can be specified for each group to define, for example:

- E-mail scanning method (see 5.2 on p. 48).
- E-mail scanning mode (see 5.3 on p. 48).
- Actions over messages and their objects (see 5.4 on p. 52).
- Backup e-mail copying before processing (see 5.6 on p. 56).
- Notifications about found objects (see 5.7 on p. 57).

Each group's settings are stored in a separate configuration file (see A.2 on p. 127). All group configuration files must be specified using the `_include` directive in the **[kav4lms:groups]** section of the main application's configuration file *kav4lms.conf*. Group configurations can be included by indicating a configuration file name or the name of a directory, containing all group configuration files.

By default, group configuration files should be located in the */etc/opt/kaspersky/kav4lms/groups.d/* directory.

The product distribution package includes the **Default** group configuration file - *default.conf*. After product installation it appears in the */etc/opt/kaspersky/kav4lms/groups.d/* directory. Values defined in that file are used as defaults if they are not specified in appropriate group configuration file. Parameters of the **Default** group configuration file are used if there are no existing groups.

The Anti-Virus scans a message using the settings of the group in which its sender or recipient are found (from the MAIL FROM and RCPT TO commands). If the sender and all recipients belong to different groups, the application selects the group with the highest *priority*. If no groups are found, such message will be processed using the settings specified in the **Default** group configuration file, which has the lowest priority of **0**. Therefore, it is recommended to specify higher protection level for groups with higher priority.

Priority is a unique group identifier. It is defined by the **Priority** option in the **[kav4lms:groups.<group\_name>.definition]** section of the group configuration file.

Senders and recipients are defined using the **Senders** and **Recipients** options in the **[kav4lms:groups.<group\_name>.definition]** section of a group configuration file.

To create a new group,

1. Create group configuration file in the directory specified in the **[kav4lms:groups]** section of the main product configuration file. The default directory is **/etc/opt/kaspersky/kav4lms/groups.d/**

**Note:**

You are advised to use the *default.conf* file as a template when creating group configuration file. Run the following commands to replace the group name quickly :

```
# cd /etc/opt/kaspersky/kav4lms/groups.d
# sed 's|groups.default|groups.<group_name>|'
default.conf > <group_name>.conf
```

2. Define group priority in its configuration file setting the **Priority** option in the **[kav4lms:groups.<group\_name>.definition]** section. Any natural value can be entered as the value. Groups with the same priority and **0** priority are not allowed.
3. Define the addresses of senders and recipients in the group configuration file settings the **Senders** and **Recipients** options in the **[kav4lms:groups.<group\_name>.definition]** section.

«\*» and «?» wildcards in masks are supported as well as regular expressions beginning with the «re:» prefix. To specify several addresses (address masks), each new record must begin in a new line:

```
Senders=reporter@*.mydomain.com
Recipients=re:office\d+@central\.mydomain\.com
```

At least one of the **Senders** or **Recipients** options must be defined. If **Recipients** or **Senders** option in group definition is missing, the application will use for that parameter the default value specified in the *default.conf* - «\*@\*» (all addresses).

**Warning!**

Regular expressions are case insensitive.

4. If necessary, specify the options for e-mail scanning in the corresponding sections of the group configuration file (see A.2 on p. 127 for details). If an option is not defined in group configuration file, the application uses the value specified for that parameter in the **Default** group configuration file - *default.conf*.

## 5.2. Definition of e-mail analysis policy

The Anti-Virus supports the following methods of e-mail analysis:

- Scanning of the whole message as a single solid object – message header and body are analyzed as a whole.
- Combined approach – a message is scanned first as a single object, then the application parses it into objects (message body, attachments, etc.) and checks each of them individually. That method provides for higher protection level and reliability.

### Note:

If an action applicable to a message part is selected as action over a whole message (see 5.4 on p. 52), then such letter will be scanned part-by-part regardless of the analysis method chosen.

The method of e-mail scanning is determined by the policy and defined by the **ScanPolicy** option in the **[kav4lms:groups.<group\_name>.settings]** section of the group configuration file.

*To scan messages as single objects,*

set the **ScanPolicy** option to **message**.

*To use combined approach while scanning messages,*

set the **ScanPolicy** option to **combined**.

## 5.3. E-mail scanning mode

The next step of group setup is to select the mode of e-mail scanning. Kaspersky Anti-Virus offers the following scanning modes:

- Scanning for the presence of malware.
- Content filtering.



The scanning mode for a group is specified via the **Check** option in the **[kav4lms:groups.<group\_name>.settings]** section of the group configuration file. It can take the following values:

- **anti-virus** – perform anti-virus mail scanning;
- **content-filter** – filter by name, type and size of attachments;
- **all** – perform both anti-virus check and content filtration;
- **none** – disable mail scanning.

If both anti-virus scanning and filtration are enabled, then analysis is performed in the following order:

1. anti-virus scan of a message as a whole object;
2. filtration of attachments;
3. message scan part-by-part (if combined scanning method is selected **ScanPolicy=combined**).

### 5.3.1. Anti-virus scanning

Anti-virus scanning is enabled by setting the **Check** option in the **[kav4lms:groups.<group\_name>.settings]** section of the group configuration file to **anti-virus** or **all**.

After a message has been scanned for viruses, the application assigns a certain status to the message or its object:

- **clean** – message contains no malware;
- **infected** – message (or its part) contains harmful objects;
- **suspicious** – message (or its part) contains a suspicious object (assigned only when heuristics analyzer is enabled);
- **protected** – message (or its part) is password-protected or encrypted;
- **error** – message is corrupted or scanning process generated an error.

The status assigned after scan is used for further processing of messages and their objects (see 5.4 on p. 52).

For infected messages (**infected**) specific handling procedure can be defined depending upon the name of the detected threat (**VirusNameAction** option in the **[kav4lms:groups.<group\_name>.actions]** section of the group configuration file). Kaspersky Anti-Virus returns the names of recognized threats in Kaspersky Lab's notation, which can be learned from [www.viruslist.com](http://www.viruslist.com). The list of virus names that are subject to action is specified via the **VirusNameList**

parameter in the **[kav4lms:groups.<group\_name>.contentfiltering]** section. This parameter excerpts virus names as-is or as regular expressions (POSIX standard).

The scanning capabilities of the application can be customized to increase the thoroughness or speed of scanning. Settings, regarding the scan engine performance are located in **[kav4lms:groups.<group\_name>.settings]** section of the group's configuration file. These settings are:

- whether to scan archives (**ScanArchives** parameter);
- whether to scan packed executables (**ScanPacked** parameter);
- whether to perform the heuristic analysis (**UseCodeAnalyzer** parameter);

**Note:**

Setting this parameter to **yes** enables the **suspicious** verdict, which is unavailable otherwise.

- how much time to spend on message or message's object scan (**MaxScanTime** parameter). If the scan time exceeds the specified limit, the scan ends with **error** verdict;
- whether the application should decode MIME objects that do not comply with RFC standards using heuristic algorithms (**MIMEEncodingHeuristics** option);
- what malware types are detected (**UseAVBasesSet** parameter in the **[kav4lms:server.settings]** section of the *kav4lms.conf* file).

## 5.3.2. Content filtering

Content filtering service is enabled by setting the **Check** parameter in the **[kav4lms:groups.<group\_name>.settings]** section of the group configuration file to **content-filter** or **all**.

The Anti-Virus can use the following criteria for content filtering:

- MIME type of attachments (applies to the "Content-Type" headers);

**Warning!**

There are situations, when actual content does not correspond to the MIME type declared. The application does not perform content identification.

- attachment name (applies to the names and extensions of attachments);
- attachment size (applies to the size of message parts, the part size is calculated after attachment decompression).

**Note:**

If both anti-virus scanning and content filtering are enabled, the content filtering is performed before scanning.

Filtration criteria are defined in the **[kav4lms:groups.<group\_name>.contentfiltering]** section of group configuration file.

Two rules can be set up for each filter criterion:

- Include rule. This rule specifies objects that are subject to filtering and described using the following parameters:
  - **IncludeMime** – specifies the list of MIME types;
  - **IncludeName** – specifies the list of attachment names;
  - **IncludeSize** – specifies the list of objects' size.
- Exclude rule. This rule specifies objects that are not subject to filtering and described using the following parameters:
  - **ExcludeMime** – specifies the list of MIME types;
  - **ExcludeName** – specifies the list of attachment names;
  - **ExcludeSize** – specifies the list of objects' size.

**Warning!**

If the include rule is empty and the exclude rule is not, then all objects, not matching the exclude rule, are included into filtering.

If both rules are empty, then content filtering is not performed, regardless of the **Check** parameter value.

Rules for MIME type and attachment name filtering criteria must be specified as the list of as:

- strings;
- wildcard (UNIX standard);
- regular (POSIX standard) expressions.

**Warning!**

Regular expressions are case insensitive; they must begin with the «re:» prefix.

Rules for object size must be specified as:

- number of bytes;
- numbers with the size mark ('KB' or 'MB');
- comparison signs.

## 5.4. Actions over objects

After the scan and content filtering is performed, Kaspersky Anti-Virus performs specific actions over messages and their parts. Some actions are applicable to whole messages only while others can apply just to message parts. The parameters that determine application actions can take the following values:

- **warn** – message is replaced completely with text warning about the presence of a dangerous object;
- **drop** – message is accepted, but silently dropped without delivery to the recipient;
- **reject** – message delivery is rejected (this action is not performed when using the application with Postfix (post-queue integration) or Exim, the bounce action is produced in that case). If this action is chosen, the sender receives a notification defined by the **RejectReply** option;
- **skip** – the message or its part is allowed to pass unchanged, the scan result is recorded into application log;
- **cure** (available only after anti-virus scan for message parts) – the application attempts to cure infected objects. If disinfection fails, the **delete** action is taken;
- **rename** (available only after content filtering for message parts) – the application adds to attachment name the **RenameTo** parameter's value. If the value defines an extension (for instance, `.vir`), then this value is added to the attachment's name. Otherwise, this value is assumed to be the whole name, so the entire attachment name is replaced;
- **delete** – the message part is removed and (if the **UsePlaceholderNotice** parameter is set to **yes**) replaced by a notification. The notification text is taken from a template file named `part_<action>`.

The actions performed after anti-virus scanning are specified by the **InfectedAction**, **SuspiciousAction**, **ProtectedAction**, **ErrorAction** and

**VirusNameAction** parameters. The actions performed after filtering are specified by the **FilteredMimeType**, **FilteredNameAction** and **FilteredSizeAction** parameters.

Action-related parameters are available in the **[kav4lms:groups.<group\_name>.actions]** section of the group configuration file.

### **Warning!**

The fact, that content filtering takes place before scan, may lead to a situation, when the whole message scan results with **infected** verdict, but part-by-part scan results with no infected part found. This can happen, if the **delete** action is chosen as the result for content filtering, and the message part is deleted after filtering.

## **5.5. Predefined security profiles**

Distribution package of Kaspersky Anti-Virus includes preset configuration profiles providing for different e-mail protection levels:

- **recommended** – stored in the *default\_recommended* directory (see 5.5.1 on p. 54 for details);
- **maximum protection** – stored in the *high\_overall\_security* directory (see 5.5.2 on p. 54 for details);
- **maximum performance** – stored in the *high\_scan\_speed* directory (see 5.5.3 on p. 55 for details).

Each profile consists of two configuration files: *kav4lms.conf* and *default.conf* (located in the *groups.d* subdirectory). Profiles are stored in the subdirectories matching their names within the */etc/opt/kaspersky/kav4lms/profiles* directory.

You can select one of the preset profiles or configure e-mail protection settings manually in the configuration files of the application.

*To use a preset profile:*

1. Create the backup copy of application's configuration files (*kav4lms.conf* and *groups.d/default.conf*).
2. Copy the contents of required profile's directory to the */etc/opt/kaspersky/kav4lms* directory.
3. Apply the new configuration by running the following command:

```
/etc/init.d/kav4lms reload
```

## 5.5.1. *Recommended profile*

This profile provides the optimal balance between anti-virus protection level and scan speed. This profile has the following characteristics:

- E-mail messages are scanned using the **message** scan policy: each message is scanned for viruses as a whole.
- Extended anti-virus databases are used while scanning.
- Maximum message nesting level allowed for MIME objects is 10.
- A backup copy and information file are created for every message that undergoes anti-virus processing.
- Infected messages are cured.
- Filtration of attachments by MIME type is enabled. The application removes from messages links to external objects (*message/external-body* type) and attachments with the *.pif*, *.com*, *.bat* and *.exe* extension.
- Warnings are issued about all messages, which are suspicious, password protected, erroneous, filtered by MIME type and attachment name. If a specific threat is detected, the message is dropped.
- The application adds to message header and body information about the results of its processing.
- The application sends notifications about message scanning to its recipients. No notifications are delivered to the sender or administrator.
- All application messages except for debug information are recorded in the report.
- Statistics are gathered for all aspects of the application functionality.

## 5.5.2. *Maximum protection profile*

This profile offers the most comprehensive protection of your mail traffic. This profile includes the following functions:

- The application scans e-mail messages using a **combined** scan policy: each message is first scanned for viruses as a whole and then each message object is scanned separately, regardless of whether infected objects are found or not.
- The application parses messages that do not comply to RFC standards using heuristic algorithms; after successful decoding it passes them for scanning.

- Extended anti-virus databases are used while scanning.
- E-mail messages are filtered by MIME type. The application filters e-mails which have references to external objects (*message/external-body*) type and deletes them. Also *.pif*, *.com*, *.bat* and *.exe*-attachments are removed.
- Maximum message nesting level is unlimited.
- An information file is created for every message that undergoes anti-virus processing or filtering.
- Infected objects are cured.
- The application deletes all suspicious, protected and filtered objects in the messages. Messages, containing threats from a specified list are dropped.
- If a message contains objects, which cause an error when scanned, its content will be replaced with a notification.
- The application sends notifications about message scanning to its recipients. No notifications are delivered to the sender or administrator.
- All application messages except for debug information are recorded in the report.
- Statistics is not preserved.

### **5.5.3. Maximum performance profile**

This profile provides maximum application performance, at some cost to the reliability of anti-virus protection. The profile has the following characteristics:

- E-mail messages are scanned using the **message** scan policy: each message is scanned for viruses as a whole.
- Message object filtering is disabled.
- The application saves a backup copy for every message to which it applies the drop and warn actions. No information file is created.
- Warnings are issued about infected, suspicious, protected and erroneous objects of mail messages. If a threat from specified list is detected, the message is dropped.
- The application adds to message header the information about the results of its processing.

- The application sends notifications about message scanning to its recipients. No notifications are delivered to the sender or administrator.
- The application logs in the activity report information about all aspects of its functionality; level of details: fatal and other errors, and important informational messages.
- Statistics is gathered about detected viruses.
- The maximum number of client requests to the central service is doubled in comparison to the recommended and maximum protection profiles. The maximum number of concurrent scanning requests is unlimited.

## 5.6. Backup

The application supports backup copying of messages prior to their processing. Backup settings are specified in the **[kav4lms:groups.<group\_name>.backup]** section of the group configuration file.

Mail backup mode is determined by Policy option, which can take the following values:

- **message** – only message copy is created;
- **info** – the information file is created together with message copy. This file contains the following information:
  - MTA client IP address (or host if available.);
  - MTA connector IP address (or host if available.);
  - the sender of the message, as provided from the MTA connector;
  - the address of processing server;
  - the name of the matching group under which the messages was analyzed;
  - the recipients list of the message, as provided from the MTA connector;
  - the cause of the backup action (cured, deleted, rejected, filtered etc.);
  - the path to the original file, relative to backup destination;
  - application instance information (process id and thread id.).
- **none** – no message backup.



The **Options** parameter specifies which application activity is the backup reason:

- **cured** – when the original message object has been cured;
- **deleted** – when the original message object has been deleted;
- **rejected** – when the original message has been rejected (the MTA client receives the error code), but the administrator may decide to backup the infected message;
- **dropped** – when the original message has been dropped;
- **warning** – when the original message has been replaced with a warning;
- **renamed** – when the message has least one object (MIME part) that had matched the filtering rules and has been renamed;
- **all** – all above mentioned.

The **Options** parameter can take one of the above values or their list, delimited by commas.

Messages' backup copies and information files are stored under a directory, specified via the **Destination** parameter.

## 5.7. Notifications

Notification is an e-mail message containing a description of processed message and sent to its recipient, sender or server administrator.

Apart from the message description, notification also contains descriptions of objects that have been removed from the message for some reason.

The application also supports appending of the original e-mail to notification. However, that is only possible for notifications to the recipient. For administrators and senders the application generates new e-mail letters containing just the notification text.

### 5.7.1. Setting up notifications

Notifications-related application parameters are stored:

- in the **[kav4lms:server.notifications]** section of the *kav4lms.conf* configuration file of the application;
- in the **[kav4lms:groups.<group\_name>.notifications]** section of each group's configuration file.

Notifications setup consists of two steps.

## Step 1. Who will be notified?

Notifications can be sent to:

- message sender (**NotifySender** parameter in the group configuration file);
- message recipients (**NotifyRecipients** parameter in the group configuration file);
- security administrators (**NotifyAdmin** parameter in group's configuration). The security administrators' e-mail addresses list is specified via the **AdminAddresses** parameter in group's configuration;
- product administrators (defined by the **ProductNotify** parameter in the *kav4lms.conf* file). Product administrators' addresses list is specified via the **ProductAdmins** parameter in the *kav4lms.conf* file.

Message sender notifications are enabled by setting these parameters to a value, other than **none**. Otherwise, notifications are disabled.

## Step 2. What will be the notifications' subject?

Message senders, recipients and security administrators can be notified about:

- the **InfectedAction** (see 5.4 on p. 52 for details) taken (at least one object was infected). This notification type is enabled by setting the required parameter to **infected** value;
- the **ProtectedAction** (see 5.4 on p. 52 for details) taken (at least one object was protected). This notification type is enabled by setting the required parameter to **protected** value;
- the **ErrorAction** (see 5.4 on p. 52 for details) taken (at least one object was erroneous). This notification type is enabled by setting the required parameter to **error** value;
- a filtering rule matched (see 5.3.2 on p. 50 for details). This notification type is enabled by setting the required parameter to **filtered** value;
- all above mentioned. This notification type is enabled by setting the required parameter to **all** value.

Product administrators can be notified about:

- a new update of anti-virus databases has been downloaded. This notification type is enabled by setting the **ProductNotify** parameter to the **update** value;

- a critical failure in the application (which was recoverable or not). This notification type is enabled by setting the **ProductNotify** parameter to the **fault** value;
- licensing related notifications. This notification type is enabled by setting the **ProductNotify** parameter to the **license** value;
- all above mentioned. This notification type is enabled by setting the **ProductNotify** parameter to the **all** value.

The license notifications is an exceptional case and cannot be excluded from the list. These kind of notifications will always be sent, and when the notifications are turned off, only log entries will be generated.

Licensing notifications are sent upon:

- key expiration – first notification is issued on the 14 day before the expiration date, then daily up to the expiration date. On the next day, the expired key notification is issued;
- license limit violation – when the number of users or amount of traffic permitted by the key was exceeded.

## 5.7.2. Notification templates

The following templates can be used to create notifications (the templates are stored in the directory defined by the **Templates** parameter in the application configuration file):

- **Template for notifications about deleted objects** – text added to the original message if one of the message parts is deleted during anti-virus processing or filtering. This text might contain a macro describing the reasons for deletion. The following templates are available:
  - *part\_infected* – text replacing the object that was deleted after an unsuccessful disinfection attempt;
  - *part\_filtered* – text replacing the MIME object that was deleted based on MIME object filtration results;
  - *part\_suspicious* – text replacing the object that was detected as suspicious and deleted;
  - *part\_filtered* – text that replaces an original e-mail object, renamed as the result of filtering;
  - *part\_protected* – text replacing an object that was deleted because it was protected and therefore could not be scanned for viruses;

- *part\_error* – text replacing the object that generated a scan error and was therefore deleted.
- **Standard notification template** – text of the notification that is sent to the sender, recipient, and administrator using the filter or a newly generated message sent by the SMTP component. This text might contain a macro describing the reasons for deletion. The following templates are available:
  - *notify\_common* – text sent by default to the recipient, sender, and administrators about the actions applied to the message;
  - *notify\_infected* – text that replaces the infected message;
  - *notify\_suspicious* – text that replaces the message containing suspicious objects;
  - *notify\_filtered* – text that replaces the filtered e-mail message;
  - *notify\_error* – text that replaces a message that generated a scan error;
  - *notify\_protected* – text that replaces a message that was protected from scanning;
  - *disclaimer* – text, added to all processed and generated messages. By default this template includes the following notification: "This message has been scanned by Kaspersky Anti-Virus. For more information about data security please visit <http://www.kaspersky.com> and <http://www.viruslist.com>".
- **Detailed notification template** – text notifying a person interested in knowing more about the anti-virus processing of an e-mail message. There are separate templates for notifications sent to the recipient, sender and administrator. Set the **UseCustomTemplates** parameter to **yes** in order to use these templates. The following templates are available:
  - sender's notifications:
    - *notify\_sender\_common* – text of the notification sent to the sender about actions applied to the original message;
    - *notify\_sender\_infected* – text that replaces the infected message;
    - *notify\_sender\_suspicious* – text that replaces the message containing suspicious objects;
    - *notify\_sender\_filtered* – text that replaces the filtered e-mail message;

- *notify\_sender\_error* – text that replaces a message that generated a scan error;
  - *notify\_sender\_protected* – text that replaces a message that was protected from scanning;
- recipients' notifications:
  - *notify\_recipients\_common* – text of the notification sent to the recipient about actions applied to the original message;
  - *notify\_recipients\_infected* – text that replaces the infected message;
  - *notify\_recipients\_suspicious* – text that replaces the message containing suspicious objects;
  - *notify\_recipients\_filtered* – text that replaces the filtered e-mail message;
  - *notify\_recipients\_error* – text that replaces a message that generated a scan error;
  - *notify\_recipients\_protected* – text that replaces a message that was protected from scanning;
- administrator's notifications:
  - *notify\_admin\_common* – text of the notification sent to the administrator about actions applied to the original message;
  - *notify\_admin\_infected* – text that replaces the infected message;
  - *notify\_admin\_suspicious* – text that replaces the message containing suspicious objects;
  - *notify\_admin\_filtered* – text that replaces the filtered e-mail message;
  - *notify\_admin\_error* – text that replaces a message that generated a scan error;
  - *notify\_admin\_protected* – text that replaces a message that was protected from scanning.
- **Special administrator notification template** – text added to special notifications sent upon critical events that require administrator's special attention. Administrator templates are stored in a directory, specified by the **Templates** parameter in the **[kav4lms:server.notifications]** section of the application configuration file. The following templates are available:

- *product\_update* – the text used to notify the administrator about receipt of updates to the anti-virus databases for the application;
- *product\_fault* – text notifying the administrator that a critical error has occurred while Kaspersky Anti-Virus was running;
- *product\_license* – text notifying the administrator about license agreement violation or end of licensing period.

**Warning!**

When the application is started, the presence of all the above templates is verified. If even one of these templates is missing, the application will return an error.

The application also verifies that the size of each template does not exceed 8 KB.

### 5.7.3. Customizing notification templates

Kaspersky Anti-Virus gives users the flexibility to customize the default notification templates that will be sent to administrators, senders, and recipients. The templates are customized using a special notification language.

The template language is a set of control statements and macros.

Below, we consider the rules of this language, its syntax and examples of use in detail.

**Warning!**

The first line of any template must not contain ':' as it will be interpreted as header. You can start with a line feed (press **Enter**) to be sure it will not be misinterpreted as notification header.

#### 5.7.3.1. Macros

A macro is a substitution element used in e-mail notification templates. In a notification text created using a template, the macro is replaced with a certain value.

The syntax for macros is `%macro_name%`.

If a macro name contains '%', it should be escaped (see section 5.7.3.5 on page 66).

Several values can be assigned to a macro. In this case, the simple input of "`%macro_name%`" will output the last assigned value.

To assign several values to one macro, use *iterative statements*.

### 5.7.3.2. Iteration constructs

An iteration construct (IC) is the main element of the template language.

The syntax for an iteration construct is

```
<FOR INAME IOP IVALUE>BODY</FOR>
```

where:

<FOR – the beginning of IC definition. The < symbol that is not the beginning of an IC definition should be screened (see section 5.7.3.5 on page 66).

INAME – IC name in the format **1\*(nchar)\*(nchar)**; the maximum length is 64 bytes.

IOP – comparison operation in the format **== | !=**; the maximum length is 2 bytes.

IVALUE – value of IC in the format **1\*(vchar)\*(vchar)**; the maximum length is 4096 bytes. IC values only work in double quotes. When comparing with a value that contains a quotation mark, use the relevant screening escape symbol (see section 5.7.3.5 on page 66). Example:

```
<FOR _macro_name_parent_ == "\"_value_1\"">
```

> – end of IC definition and the beginning of iterator body. The < symbol that is not the end of IC definition must be hidden (see section 5.7.3.5 on page 66).

BODY – iterator body in the format **\*(char)**.

</FOR> – end of the iterator body definition. The < symbol that is not the end symbol of the iterator body definition must be screened (see section 5.7.3.5 on page 66).

... – separator in the format **\*( )\*(t)**

nchar – characters from set a-z, A-Z, 0-9, -, \_

vchar – symbols from set nchar, \*, ?

char – symbols from the set of values 32 – 255

Example of an iteration construct:

```
<FOR _macro_name_ == "*" %_macro_name_%</FOR>
```

When executing this construct, the parser transforms the above command into the condition constructs:

```
<FOR _macro_name_ == "_value_1" %_macro_name_%</FOR>
```

```

<FOR _macro_name_ == " value 2">%_macro_name_%</FOR>
<FOR _macro_name_ == " value 3">%_macro_name_%</FOR>
<FOR _macro_name_ == " value N">%_macro_name_%</FOR>

```

These condition constructs are parsed sequentially.

Thus, iteration constructs are used to distinguish both the single and multiple values of a macro.

For example, if the macro %FILTERNAME% has the values of KAVFilter1, KAVFilter2, KAVFilter3, and SimpleFilter, then

the construct:

```

<FOR FILTERNAME == "KAVFilter1">%FILTERNAME%</FOR>

```

will produce the text:

```

KAVFilter1

```

the construct:

```

<FOR FILTERNAME == "KAVFilter?">%FILTERNAME%, </FOR>

```

will produce the text:

```

KAVFilter1, KAVFilter2, KAVFilter3

```

the construct:

```

<FOR FILTERNAME != "KAVFilter2">%FILTERNAME%, </FOR>

```

will produce the text:

```

KAVFilter1, KAVFilter3, SimpleFilter

```

the construct:

```

<FOR FILTERNAME != "KAV*">%FILTERNAME%, </FOR>

```

will produce the text:

```

SimpleFilter

```



### 5.7.3.3. Scope of visibility for an iterative statement

Any iteration construct can have sub-macros, which values are defined within the scope of visibility for the parent construct only. Iterative statements can be used not only to output particular values of particular macros, but also to define the scope of visibility of sub-macros.

The scope of visibility of a sub-macro is defined by the start and end tags of the condition construct:

```
<FOR _macro_name_parent_ ==
  " _value 1">%_macro_name_child_%</FOR>
```

In the above example, the scope of the macro `%_macro_name_parent_` includes all sublevels (between the **FOR** tags) if the macro value is overridden.

### 5.7.3.4. Variables

Variables provide better flexibility in customizing templates using the Template language.

A variable can be defined within the specified scope of flexibility as follows:

```
<DEF _var_name_ = " _const_value_ "/>
```

This variable can be used further as a usual macro without any limitations.

The syntax for a variable definition statement is as follows:

```
<DEF VNAME VOP VVALUE/>
```

where:

**<DEF** – beginning of variable definition statement. The **<** symbol that is not the beginning of the statement must be screened (see section 5.7.3.5 on page 66);

**VNAME** – variable name in the format **1\*(nchar)\*(nchar)**; the maximum length is 64 bytes;

**VOP** – assignment operation in the format **=**, the length is 1 byte;

**VVALUE** – variable value in the format **1\*(vchar)\*(vchar)**; the maximum length is 4096 bytes. The value only works in double quotes. If compared with a value that has a quote mark inside, use the screening escape symbol (see section 5.7.3.5 on page 66). Example:

```
<DEF _value_name_ = "\" _value 1\""/>
```

> – end of the variable definition statement. The > symbol that is not the end of the variable definition must be screened (see section 5.7.3.5 on page 66). Unlike the FOR statement, the DEF statement has no body. Therefore, the tag end bracket should notify the parser that the end tag is missing.

... – separator in the format **\*(\*)\*(\t)**

nchar – symbols from set a-z, A-Z, 0-9, -, \_

vchar – symbols from set nchar, \*, ?

If a variable is redefined in its scope, a new value will be substituted after each redefinition. Thus, the statement:

```
<DEF __NAME__ = "NAME_1"/>Now you will see the first
value: %__NAME__%.
```

```
<DEF __NAME__ = "NAME_2"/>Now you will see the sec-
ond value: %__NAME__%.
```

will be output as:

```
Now you will see the first value: NAME_1.
```

```
Now you will see the second value: NAME_2.
```

A variable can have a macro as its value.

```
<DEF _var_name_ = "% macro_name %"/>
```

In this case, the parser will first substitute the macro for a value and then it will replace the variable with this value in the current scope.

## 5.7.3.5. Language syntax

### Special symbols

- % marks a macro. The macro should be between two symbols "%".  
Example: %VIRUSNAME%
- < opening bracket of a tag.  
Example: <FOR FILTERNAME == "KAVFilter1">
- > closing bracket of a tag.  
Example: <FOR FILTERNAME == "KAVFilter1">
- </ opening bracket of an end tag.  
Example: </FOR>

- />** closing bracket of the end tag for a construct without a body.  
Example: `<DEF __NAME__ = "NAME_1"/>`
- \** escape symbol. Instructs the parser to treat the following special character as a plain one. Example: `\%VIRUSNAME\%`
- ==** equal sign: a coincidence in mask or value.  
Example: `<FOR FILTERNAME == "KAVFilter1">`  
Example: `<FOR FILTERNAME == "KAVFilter*">`
- !=** unequal sign: a non-coincidence in mask or value.  
Example: `<FOR FILTERNAME != "KAVFilter1">`  
Example: `<FOR FILTERNAME != "KAVFilter*">`
- \*** Unlimited length of all possible values. It is used only inside tags in comparison with templates.  
Example: `<FOR FILTERNAME == "KAV*">`
- ?** All possible one-character values. It is used only inside tags in comparison with templates.  
Example: `<FOR FILTERNAME == "KAVFilter?">`
- #** Comment; the parser ignores all characters after '#' till the end of line.

### Reserved keywords

- FOR** Iteration construct definition.  
Example: `<FOR FILTERNAME = "KAVFilter1">`
- DEF** Variable definition (statement without an end tag). Example: `<DEF __NAME__ = "NAME_1"/>`

### Predefined macros

- %CRLF%** Line feed macro (CR+LF)
- %TAB%** Tab macro

The processing is performed within a global section (no statement is needed) or within a condition construct:

```
<FOR KAV_LANGUAGE == "5.0"> ... </FOR>
```

## Escape sequences

The following sequences can be used to present special characters in the template language:

- To output the ‘\’ symbol in the template text, enter ‘\\’.
- If a line is ended with ‘\’, it will be interpreted as a string continued on the following line. Additionally, an escape symbol at the end of the line screens the following EOL which otherwise would exist in the generated message. Such a line is concatenated with the following one during processing before any other actions performed by the parser. This situation is handled independently by either the escape sequence being met inside a tag or outside a tag. See item 1 above if you want to place a ‘\’ at the end of line.
- To output the ‘%’ symbol into the template text, use ‘\%’.
- To output the ‘/’ symbol into the template text, use ‘\/’.
- To output the ‘<’ symbol into the template text, use ‘\<’.
- To output the ‘>’ symbol into the template text, use ‘\>’.
- To output the ‘#’ symbol into the template text, use: ‘\#’.

### Note:

The template language is case sensitive. The number of spaces or tab symbols (either their presence or absence) between the language constructs is not regulated. Reserved keywords must be separated either by white space characters or by the special symbols.

## 5.7.3.6. Notification macros for the application

Macros can be used in notification templates for either entire messages or their parts. Using macros, you can customize notifications to include additional information on the properties of an original message or object or about actions applied to them.

The administrator can use the following macro in notifications concerning entire messages:

**%VERSION%** – version number of the installed Kaspersky Anti-Virus instance used to scan the message.

**%PRODUCT%** – complete product name of Kaspersky Anti-Virus.

**%CLIENT%** – remote IP address of the mail client.

**%SERVER%** – server's name of the server running the central service of the application.

**%SENDER%** – sender address.

**%RECIPIENTS%** – recipient address.

**%HEADERS%** – message header.

**%MSGID%** – message identification number.

**%SUBJECT%** – subject (the **Subject** field) of the original message.

**%DATE%** – date when message was processed.

**%TIME%** – time when message was processed.

**%BK\_ACTION%** – actions applied to the message that caused a backup copy to be created (if the application is configured to back up messages).

**%BK\_LOCATION%** – full path to the backup storage (if the storage exists).

**%ACTION\_LIST%** – list containing information about the message and its object and a list of actions applied to them. The information is output in the following format:

```
<status> <action> <information>
```

for each processed part of the message.

In notifications related to objects deleted from a message, the following macros can be used:

**%INFO%** – information related to the following actions performed:

- list of detected viruses (malicious software) – for infected objects;
- error code description – for objects that generated a scan error;
- MIME type or attachment name – for filtered objects.

The macros must be specified in the text of notification templates.

# CHAPTER 6. ANTI-VIRUS PROTECTION OF FILE SYSTEMS

The *kav4lms-kavscanner* component provides anti-virus protection of the computer's file systems, by scanning files and processing infected and suspicious objects according to its settings.

**Note:**

All settings of the *kav4lms-kavscanner* component are grouped in the **[scanner.\*]** options of the application's configuration file.

**Warning!**

By default, only the **root** and **kluser** users can launch an on-demand scan.

You can scan the entire file system, an individual directory or a single file. All protection settings may be divided into groups that define:

- Scan scope (see 6.1 on p. 71).
- How objects are to be scanned and disinfected (see 6.2 on p. 72).
- Actions to be performed on objects (see 6.3 on p. 72).

The scan of your computer's file systems may be started:

- As a one-time task - from the command line (see 6.4 on p. 74).
- According to the schedule using the **cron** application (see 6.5 on p. 74).

**Warning!**

An anti-virus scan of the entire computer is a process that requires considerable resources. It should be noted that when you start this task, your computer's efficiency will be reduced: therefore we recommend that no other heavy application should run at the same time. To avoid such problems, we recommend that you scan individual selected catalogs.

## 6.1. Scan scope

The scan scope can be roughly divided into two parts:

- *scan path* – the list of directories and objects to be searched for viruses;
- *scan objects* – types of objects to be scanned for the presence of viruses (archives, etc.).

By default all objects of all available file systems are scanned, starting with the current directory.

### Note:

To scan all file systems of the computer, you have to switch to the root directory, or specify the scan scope at the command line as “/”.

You can redefine the scan path by the following methods:

- Listing at the command line (using a space as a separator) all directories and files to be scanned, using absolute or relative (relative to the current directory) paths.
- List the scan paths in a text file, and specify this file to be used by using the parameter **-@<filename>** in the command line. Each object in this file should be entered on a new line, using its absolute path only.

### Warning!

If you specified at the command line both scan paths and a text file containing a list of the scan objects, only the paths indicated in the file will be scanned. The paths entered at the command line will be ignored.

- Turn off the *recursive scan of the catalogs* (**[scanner.options]** section, the **Recursion** setting or command line parameter **-r**).
- Create an alternative configuration file and specify this file to be used using the command line parameter **-c <filename>** at component startup.

The path to the object for scanning may not exceed 4096 bytes. Objects located on longer paths will not be scanned.

The default scan objects are specified in the *kav4lms.conf* configuration file (**[scanner.options]** section) and they can be redefined:

- directly in this file;
- using command line parameters at component startup;

- by using an alternative configuration file.

## 6.2. Object scan and disinfection mode

The settings of this mode are very important, because they determine whether the application will cure infected files when they are detected.

By default, disinfection is turned off: the default behavior is to scan objects and to notify about detected viruses and other suspicious or corrupted files by printing messages to the screen and in the report.

As a result of an anti-virus scan, each object will be assigned a status from those listed below:

- **Clean** – no viruses detected (the object is not infected).
- **Infected** – the object is infected.
- **Warning** – object code resembles the code of a known virus.
- **Suspicious** – the object is suspected of being infected with an unknown virus (not assigned if the **UseCodeAnalyzer=no**).
- **Corrupted** – the object is corrupted.
- **Protected** – the object cannot be scanned because it is encrypted (password-protected).
- **Error** – an error has occurred while scanning the object.

With the disinfection mode turned on (section [**scanner.options**], setting **Cure=yes**) only objects with the **Infected** status will be sent for disinfection. As a result of the disinfection, the object will be assigned a status from those listed below:

- **Cured** – the object has been successfully disinfected.
- **CureFailed** – the object could not be disinfected. Files with this status will be processed according to rules specified for infected objects.

## 6.3. Actions to be performed on objects

The actions to be performed on an object depend on the object's status. The default action is only to provide notification about the detection of infected or



suspicious objects. However, for objects with **Infected**, **Suspicious**, **Warning**, **Error**, **Protected** and **Corrupted** status you can configure further responses, including:

- *moving to a directory* – moving objects with the given status to a directory (*simple* and *recursive* moving is supported);
- *deleting object* from the file system;
- *performing a command* – processing of files using standard Unix script files, or similar.

Please note that Kaspersky Anti-Virus discriminates between simple objects (files) and container objects (consisting of several objects, for example, an archive). Actions performed with such objects are also discriminated; in the configuration files these actions are located in different sections, with section **[scanner.object]** for simple objects, and section **[scanner.container]** for container objects.

### Warning!

Actions performed with self-extracting archives can be differentiated: if the archive itself is infected, it will be viewed as a simple object, while if objects within the archive are infected, the archive will be viewed as a container. Therefore actions to be performed on archives, depending on the case, will be determined by the settings specified in different sections of the configuration file.

You can select actions to be performed on an object using several methods as follows:

- You can specify them in the *kav4lms.conf* configuration file if you plan to use these actions as default actions (sections **[scanner.object]** and **[scanner.container]**).
- Specify actions in the alternative configuration file and use this file at component startup.

### Note:

If no configuration file is specified in the command line at the component startup, the operating settings will be taken from the *kav4lms.conf* file. The use of this file at startup does not have to be specified.

- You can specify them for the current work session using command line parameters when starting the *kav4lms-kavscanner* component.

Actions for both simple and container objects use the same syntax (sections **[scanner.object]** and **[scanner.container]**).

## 6.4. On-demand scan of an individual directory

One of the commonest tasks implemented by Kaspersky Anti-Virus is the anti-virus scan and disinfection of an individual directory.

*Perform the anti-virus scan with the following conditions:*

1. Start an anti-virus scan of the /tmp directory with automatic disinfection of all infected objects detected. Delete all objects that cannot be disinfectd.
2. Create the files **infected.lst**, **suspicion.lst**, **corrupted.lst** and **warning.lst** to record the filenames of all infected, suspicious and corrupted objects detected during the scan.
3. The results of the component operation (starting date, information about all files, except clean files) will be printed in the report file `kavscanner-current_date-pid.log` that will be created in the current directory.

*To implement this task, enter at the command line:*

```
# /opt/kaspersky/kav4lms/bin/kav4lms-kavscanner -\
rlq -pi/tmp/infected.lst -ps/tmp/suspicion.lst -\
pc/tmp/corrupted.lst -pw/tmp/warning.lst -o /tmp/ \
kav4lms-kavscanner-`date +%Y-%m-%d-%H` .log -i3 \
```

## 6.5. Scheduled scan

Kaspersky Anti-Virus tasks can be scheduled to run using the **cron** application.

*Run an anti-virus scan of the /home directory every day at 0:00, using the scan settings specified in the configuration file /etc/kav/scanhome.conf. To perform this task, do the following:*

1. Create the configuration file `/etc/kav/scanhome.conf` and specify the required scan settings in this file.
2. Edit file that defines the rules for the operation of the cron (**crontab -e**) process by entering the following line:

```
0 0 * * * /opt/kaspersky/kav4lms/bin/kav4lms-\
kavscanner -c /etc/kav/scanhome.conf /home
```

## 6.6. Sending notifications to the administrator

Using standard Unix tools, you can specify that notifications are sent to the administrator upon detection of infected, suspicious or corrupted objects in the computer's file systems.

*Configure administrator notification when infected files and archives are detected during file systems scans performed using the settings specified in the `kav4lms.conf` configuration file.*

### Warning!

The example is for Linux!

To perform this task, do the following:

Enter these rules for processing simple objects and container objects in the configuration file `kav4lms.conf`:

```
[scanner.object]
OnInfected=exec echo %FULLPATH%/%FILENAME% is \
infected by %VIRUSNAME% |
mail -s kav4lms-kavscanner admin@localhost
[scanner.container]
OnInfected=exec echo archive %FULLPATH%/%FILENAME% \
is infected, viruses list is in the attached file \
%LIST% | mail -s kav4lms-kavscanner -a %LIST% \
admin@localhost
```

### Warning!

Before launching the example, make sure that the **mail** utility is located at this utility's standard installation path in the operating system.

# CHAPTER 7. UPDATING THE ANTI-VIRUS DATABASES

Updating the anti-virus database is performed by the *kav4lms-keepup2date* component, and is an integral factor in full-fledged anti-virus protection. The default source used for updating the anti-virus database is Kaspersky Lab's updates servers. The list of these servers includes:

<http://downloads1.kaspersky-labs.com/>

<http://downloads2.kaspersky-labs.com/>

<ftp://downloads1.kaspersky-labs.com/>, etc.

The list of URL's from which you can download the updates is contained in the *updcfg.xml* file, included in the application's distribution kit. To view the list of update servers, enter the following in the command line:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-keepup2date -s
```

During the update process, the *kav4lms-keepup2date* component selects the first address from this list and attempts to download the anti-virus database from the server. The current computer location (as the two-lettered code of the country according to the ISO 3166-1 standard) can be specified via the **RegionSettings** parameter in the **[updater.options]** section of the application configuration file. In this case the *kav4lms-keepup2date* component starts choosing the update servers, marked as belonging to the specified region. If the update cannot be performed from the address selected, the component switches to the next URL and makes another attempt.

## Note:

Updates to the anti-virus database are uploaded to Kaspersky Lab's updates servers on an hourly basis.

After a successful update, a command, specified by the **PostUpdateCmd** parameter of the configuration file's **[updater.options]** section, is executed. By default, this command automatically reloads the anti-virus database. If an invalid change is made to this setting, the application may fail to use the updated database or will function improperly.

## Note:

All settings of the *kav4lms-keepup2date* component are grouped in the **[updater.\*]** section of the configuration file.

If the structure of your local area network is complex, you are advised to download updates to the anti-virus database from the updates servers every hour, place them in a network directory, and configure local computers throughout the network to use this directory as their update source. For details on the creation of a network directory, see 7.3 on p. 79.

The update may be scheduled using the **cron** utility (see 7.1 on p. 77) or it may be performed on-demand by the administrator who can run this task manually from the command line (see 7.2 on p. 78).

## 7.1. Automatically updating the anti-virus database

You can schedule regular automatic updates of the anti-virus database by modifying the configuration file.

*Configure automatic anti-virus database updates to be performed every hour. Only record application errors in the system log. Maintain the general log for all tasks started, and do not print any information to the screen. To perform this task, do the following:*

1. Specify these values in the application's configuration file, for example:

```
[updater.options]
KeepSilent=yes
[updater.report]
Append=yes
ReportLevel=1
```

2. Edit the configuration file for the cron (crontab -e) process by entering the following line:

```
0 0-23/1 * * * /opt/kaspersky/kav4lms/bin/kav4lms-\
keepup2date -e
```

*Configure the kav4lms-keepup2date component to select the URL of the updates server automatically from the list, provided with the application. To perform this task, do the following:*

Assign the value **No** to the **UseUpdateServerUrl** setting in the **[updater.options]** section of the application's configuration file.

Configure the `keepup2date` component to download updates from the URL specified by the administrator. If the download cannot be performed from this URL, abort the downloading process. To perform this task, do the following:

Assign the value **Yes** to both the **UseUpdateServerUrl** and **UseUpdateServerUrlOnly** settings of the `[updater.options]` value. Additionally, the **UpdateServerUrl** setting must contain the URL of the updates server.

Configure the `keepup2date` component to download updates from a specified URL. If the download cannot be performed from this URL, update the anti-virus database from the URLs specified in the list included in the `keepup2date` component. To perform this task, do the following:

Assign the value **Yes** to the **UseUpdateServerUrl** setting of the `[updater.options]` section, and the value **No** to the **UseUpdateServerUrlOnly** setting. Additionally, the **UpdateServerUrl** setting must contain the URL of the updates server.

## 7.2. On-demand updating of the anti-virus database

You can start the update of the anti-virus database from the command line at any time. To do that, type the following command:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-keepup2date
```

Start the update of the anti-virus database and record the results in the file `/tmp/updatesreport.log`. To implement this task enter at the command line:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-keepup2date -l \
/tmp/updatesreport.log
```

The most convenient way to update the anti-virus database on several computers is to download the updates once from the updates servers, place the updates in a network directory and then direct the computers to treat this directory as their update source.

Arrange updating of the anti-virus database from the network directory `ftp://10.10.10.1/home/bases` and only if this directory is not accessible or empty, update the database from Kaspersky Lab's updates servers. Print the results in the **report.txt** report file.

To perform this task, do the following:

1. Specify the corresponding values for the settings in the application's configuration file:

```
[updater.options]
```

```
UpdateServerUrl=ftp://10.10.10.1/home/bases
UseUpdateServerUrl=yes
UseUpdateServerUrlOnly=no
```

2. Enter at the command line:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-keepup2date -l \  
/tmp/report.txt
```

## 7.3. Creating a network directory to store the updates

To ensure that the anti-virus database is correctly updated from the network directory, the directory must contain the same file structure as Kaspersky Lab's updates servers. Provided below is a detailed discussion of this task.

*Create a network directory from which anti-virus database updates can be copied to local computers within the network. To perform this task, do the following:*

1. Create a local directory.
2. Start the *kav4lms-keepup2date* component as follows:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-keepup2date -u  
<dir>
```

where *<dir>* is the full path to the local directory.

3. Grant local computers read-only network access to this catalog.

*Configure the anti-virus database update to be performed via a proxy server. To perform this task, do the following:*

1. Assign the value **Yes** to the **UseProxy** setting of the **[updater.options]** section.
2. Make sure that the **ProxyAddress** setting in the **[updater.options]** section of the configuration file contains the URL of the proxy server. The address must be specified in the format **http://username:password@ip\_address:port**. The values **ip address** and **port** are mandatory, while **username** and **password** are necessary only if the proxy server requires authorization.

*or:*

1. Assign value **Yes** to the **UseProxy** setting of the **[updater.options]** section.

2. Specify the environment variable **http\_proxy** using format **http://username:password@ip\_address:port**. Note that the environment variable will be considered only if the **UseProxy** setting of the **[updater.options]** section is missing or is assigned value **Yes**.



# CHAPTER 8. KEY MANAGEMENT

The key file gives you the right to use the application, and contains all required information pertaining to the license that you have purchased, including the licensing scheme, the key expiration date, and details of the dealer.

In addition to the right to use the application, during the key's active period you obtain:

- 24/7 technical support;
- new updates of the anti-virus database on an hourly basis;
- application updates (patches);
- receiving new versions of the application (upgrades);
- up-to-date information about new viruses.

Upon the expiration of the key you automatically lose the right to receive the above services. Kaspersky Anti-Virus will continue performing anti-virus processing, but it will use the anti-virus database that was up-to-date on the key expiration date. The anti-virus database updating function will not be available. If the anti-virus database is updated manually, its release date may be newer than the key expiration date. In this case, the application will lose its anti-virus functionality and the corresponding note is logged.

Therefore, it is extremely important to regularly review report files that contain the key details, and to keep track of the key expiration date.

The application supports several licensing schemes:

- **by traffic.**

This licensing scheme offers protection for the amount of daily traffic, specified in the key. Only the processed traffic, found **clean** or **notchecked**, is taken into account. If the daily traffic exceeds the license limit, the administrator's notification is issued for the first and subsequent messages, exceeding the license limit.

- **by addresses.**

This licensing scheme offers protection for a certain number of mail addresses. This applies to list of domains, specified via the **LicensedUsersDomains** parameter in the **[kav4lms:server.settings]**

section of the *kav4lms.conf* file, and to addresses on the server, where the application is running.

The licensed domain names can be specified:

- as-is string
- by wildcard expression (UNIX syntax)
- by regular expression (POSIX syntax).

**Warning!**

Regular expressions are case insensitive.

If number of mail addresses within a domain exceeds license limit, the administrator will be prompted to purchase a key for the amount of extra traffic.

## 8.1. Viewing key details

Apart from this, Kaspersky Anti-Virus provides a special *kav4lms-licensemanager* component that allows you to view not only the full information about the keys, but also receive some analytical data.

All information will be printed to the screen.

To view information about all keys, enter at the command line:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-licensemanager -s
```

Information similar to the following will be printed to the screen:

```
Kaspersky license manager for Linux. Version  
5.6/RELEASE #68
```

```
Copyright (C) Kaspersky Lab, 1997-2007.
```

```
Portions Copyright (C) Ian Crypto
```

```
License info:
```

```
Product name: Kaspersky Anti-Virus BO for SendMail /  
Qmail / Postfix Milter API International Edition. 10-  
14 MailAddress 1 month Beta Licence
```

```
Expiration date: 01-09-2007, expires in 28 days
```

```
Active key info:
```

```
Key file:          00BEA0DB.key
```

```
Install date:     02-08-2007
```

```
Product name:    Kaspersky Anti-Virus BO for SendMail
/ Qmail / Postfix Milter API International Edition.
10-14 MailAddress 1 month Beta Licence
Creation date:   02-02-2007
Expiration date: 03-03-2008
Serial:          0038-000413-00BEA0DB
Type:            Beta
Count:           10
Lifespan:        30
Objs:            7:10
```

The `Objs` parameter represents the licensing object. Its value consists of `<type_of_objects>:<number_of_objects>` parts. The `<type_of_objects>` part can have the following values:

- o 3 – represents the daily traffic;
- o 7 – represents the mail addresses.

The `<number_of_objects>` part has the same value, as `Count` parameter.

*To view information about a specific key, enter at the command line:*

```
# /opt/kaspersky/kav4lms/bin/kav4lms-licensemanager\
-k <key filename>
```

where `<key filename>` is the name of the key file, for instance, `0003D3EA.key`.

The following information will be printed to the screen:

```
Kaspersky license manager for Linux. Version
5.6/RELEASE #68
Copyright (C) Kaspersky Lab, 1997-2007.
Portions Copyright (C) Lan Crypto
Product name:    Kaspersky Anti-Virus BO for SendMail
/ Qmail / Postfix Milter API International Edition.
10-14 MailAddress 1 month Beta Licence
Creation date:   02-02-2007
Expiration date: 03-03-2008
Serial:          0038-000413-00BEA0DB
Type:            Beta
Count:           10
Lifespan:        30
Objs:            7:10
```

## 8.2. Renewing your key

Renewal of your key grants you the right for to restore the application's full functionality: that is, to update the anti-virus database, and resume the additional services listed in 1.3 on p. 10.

The key period depends on the type of licensing that you selected when you purchased the application.

*To renew your key:*

Contact the dealer you purchased the application from, and renew your license for the use of Kaspersky Anti-Virus.

or:

Renew your key directly at Kaspersky Labs, by sending a request directly to our Sales Department ([sales@kaspersky.com](mailto:sales@kaspersky.com)), or filling out a form at our website (<http://www.kaspersky.com>), section **eStore -> Renewal**. Upon receipt of your payment, we will send a new key to the e-mail address specified in your order.

### Note:

Kaspersky Lab Ltd. periodically announces campaigns that give you considerable discounts when you renew your license for our products. To keep informed about our offers, visit Kaspersky Lab's corporate website and go to **Products → Sales and special offers**.

You must install the key that you purchased.

*To install your new key, enter at the command line:*

```
# /opt/kaspersky/kav4lms/bin/kav4lms-licensemanager\  
-a <key filename>
```

After this we recommend that you update your anti-virus database (see Chapter 7 on p. 76).

*To remove a key, enter at the command line:*

```
# /opt/kaspersky/kav4lms/bin/kav4lms-licensemanager\  
-da
```

to remove active key,

or

```
# /opt/kaspersky/kav4lms/bin/kav4lms-licensemanager\  
-dr
```

to remove the additional key.

# CHAPTER 9. REPORTING AND STATISTICS

## 9.1. Application logging

### Note:

The application supports logging for both its components: the server and the filter. Logging options are stored correspondingly in the `[kav4lms:server.log]` and `[kav4lms:filter.log]` sections of the `kav4lms.conf` file.

Application components' work results are stored in either system log or a log file. The destination is specified via the **Destination** parameter. The syntax for destination is:

- `syslog:<name>@<facility>` - log as application `<name>`, to category `<facility>`
- `file:<log_file_path>` - messages are logged to specified file.

### Warning!

Do not use the same file as the log destination for server and filter – only one process can use the log file at once.

The type and thoroughness of logged information is specified via the **Options** parameter. The **Options** parameter value is the list of logging options. Logging option consists of two parts, delimited by dot:

1. Logging module. This part represents the functionality module of the application, whose work is being logged. Possible values are:
  - **all** - includes all groups;
  - **config** - messages related to configuration;
  - **app** - events from the business logic of the product;
  - **scan** - scan status, actions;
  - **cfilter** - content filtering status, actions;
  - **backup** - backup related message;
  - **notif** - messages from notifications system;

- **admin** - events related to administrative features (for instance, SNMP, commands);
  - **smtp** - SMTP dialog information between MTA and application.
2. Report level. This part represents the importance of the logged information. It can be specified by name, or by a letter or number. See the table for available options and descriptions.

Level symbol	Level name	Description
0, F	<b>fatal</b>	Information about critical errors only. For example, the component is infected, or an error occurred during verification, or loading of the database or the keys. Critical errors information is marked with 'F' symbol in the log file.
1, E	<b>error</b>	Information about other errors including those that cause the component to close: for example, object scan error information. Non-critical errors are marked with 'E' symbol in the log file.
2, W	<b>warning</b>	Information about errors that may cause the application to close: for example, information about insufficient free disk space or key expiration. Such messages are marked with 'W' symbol in the log file.
3, I	<b>info</b>	Important informational messages: for example, information stating whether the component is running, the path to the configuration file, scan scope, information about the anti-virus database, about keys, and statistical info about the results. Informational messages are marked with 'I' symbol in the log file.
4, A	<b>activity</b>	Messages about current application activity (for example, the name of the object being scanned). Such messages are marked with 'A' symbol in the log file.

Level symbol	Level name	Description
9, D	<b>debug</b>	Debug messages. Such messages are marked with 'D' symbol in the log file.

Logging options can be specified in the following manners:

- a combination of group and level (for instance, **scan.info**);
- level-group combination, prefixed with '-' will determine the exclusion of the specified option.

Example:

```
[kav4lms:server.log]
Options = backup.all, config.error, scan.all, -scan.debug
Options = backup.all, config.E, scan.all, -scan.9
```

This will enable all backup messages, all error messages from config and all scan messages, except debug. The second example is identical to the first one and shows usage of level selection options.

### **Warning!**

Report levels do not contain the previous (lower) levels. In order to select several levels, all these levels must be listed or non-desired levels must be excluded.

Log files can grow very quickly, but their size can be limited by enabling the logs rotation. This feature is enabled by setting the **RotateSize** and **RotateRounds** parameters to non-zero values.

If logs rotation is enabled, then the log file grows until its size reaches **RotateSize**. Then it is renamed to have '.1' suffix. If file with this suffix already exists, then files with suffixes '.2', '.3', etc. are created, until their number (suffix) reaches **RotateRounds** value. When this value is reached, file with '.1' suffix will be used again.

## **9.2. Application statistics**

### **Note:**

The options of application statistics collection are located in the **[kav4lms:server.statistics]** section of the main configuration file.

While the application is running, statistics of two types is collected:

- **General** statistics gathered from time to time and reflecting overall application activity.
- **Detailed statistics** gathered on each processed message.

The type of statistics stored is specified via the **Options** parameter. The list of available values is given in the table below.

Statistics category	Options value	Information stored
Messages	<b>messages</b>	Number of incoming messages, number of scanned messages, number of protected messages, number of infected messages, number of erroneous (corrupted) messages, average of all message sizes (in bytes), average time spent on checking one message (in milliseconds)
System resources	<b>resources</b>	The time in seconds since last request for statistics was made, total traffic size (in kilobytes), total CPU usage by user, total CPU usage by system
Detected threats	<b>viruses</b>	Latest 10 viruses detected, first 10 IP addresses that sent most viruses
Content filtering	<b>filters</b>	Number of MIME-filtered messages, number of messages filtered by attachment, number of messages filtered by size, number of messages filtered by virus name
All	<b>all</b>	All above mentioned
Per message statistics	<b>raw</b>	Comprehensive per message statistics
No statistics	<b>none</b>	No statistics are gathered

The **Options** parameter's value is the list of the mentioned values, delimited by commas.



**Examples:**

```
Options = all
```

Will collect only the aggregate ones (messages, resources, viruses, filters)

```
Options = all, raw
```

Will also collect per message statistics.

```
Options = none, raw
```

Will collect only per-message data, no aggregates.

In order to enable statistics collecting, set the **Options** parameter to value, other than **none**.

**Warning!**

Setting the **Options** parameter to **all** does not enable raw statistics! This statistics type must be specified explicitly.

Sample record from the raw statistics file:

```
1210247100      1208      from@exmaple.com  
rcpt@example.com  infected      EICAR-Test-File 127.0.0.1  
1Ju4YW-000Du9-0U Default
```

where:

- 1210247100 – time when a message was processed (in UNIX format);
- 1208 – message size;
- from@example.com – message sender's address;
- rcpt@example.com – message recipient's address;
- infected – status assigned to message after scanning;
- EICAR-Test-File – name of the threat detected in message;
- 127.0.0.1 – IP address used to send the message;
- 1Ju4YW-000Du9-0U – message ID in mail system queue;
- Default – name of the group associated with the settings used to process the message.

To write statistics to a file run the following command:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-cmd -m statistics -x \  
write
```

This command also rewrites the existing statistics file with new information.

To reset internal statistics counters run the following command:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-cmd -m statistics -x \  
reset
```

**Note:**

The statistics file must be rewritten to reflect changes after the counters have been reset.

The statistics functionality parameters are grouped in the **[kav4lms:server.statistics]** section of the *kav4lms.conf* file.

There are two types of statistics:

- **aggregate** - accumulated over time, reflecting overall product activity;
- **per message** - written for each processed message, showing detailed information about its processing; these are also called **raw** statistics.

The aggregate statistics are stored in the file, specified via the **Destination** parameter. The raw statistics are stored in the file, specified via the **RawDestination** parameter.

**Warning!**

In case a message contains several types of objects that have different scanning verdicts, then the same message is accounted by each of the corresponding counters. Therefore, counters are not cumulative - i.e. their sum may not give the number of total scanned messages.

For example, a single message with three attachments: one infected, one password protected, and one of type `application/msword` might be counted (depending on configuration) for:

- **total\_messages** - because was one of the transferred messages;
- **scanned\_messages** – because it was analyzed;
- **protected\_messages** – because it had a protected part;
- **infected\_messages** –because it had an infected part;
- **filtered\_mime** – because it had a matching MIME type.

The statistics can be gathered in 2 formats:

- **txt**-file
- **xml**-file.

The statistics file format is specified via the **Format** parameter.

# CHAPTER 10. ADVANCED SETTINGS

## 10.1. Monitoring of protection status via SNMP

Beginning with version 5.6, the application offers read-only access to the following information via the SNMP protocol:

- *product configuration* – parameters from all sections of the application's configuration files, including group configuration files;
- *operational statistics* – comprehensive statistics about the application's operation.

### Note:

The application works with agents, supporting the SNMP protocol, v1, v2, and v3. Please note that the product sends v2 traps, so the trap sink must be configured accordingly.

The information that can be accessed over SNMP is determined by the **SNMPServices** parameter, located in the **[kav4lms:server.snmp]** section of the *kav4lms.conf* configuration file. This parameter can take the following values:

- **config** – application configuration information;
- **statistics** – operational statistics (see 9.2 on p. 87 for details on published statistics);
- **admin** – administrative information that contains:
  - **Status.StartedOn** – the date when the application was started, in ISO 8601 format;
  - **Status.UpTime** – the time (in seconds) that has elapsed since the application started;
- **update** – application update information that includes:
  - **Last.Checked** – the date of the last check for an update, in ISO 8601 format;

- **Last.Result** – the status of the last update which can be:
  - **updated** – successful update, new anti-virus databases were installed;
  - **not-needed** – update completed correctly, but no new files were needed;
  - **error** – update process has failed;
  - **rolled-back** – update was successful, but anti-virus database was corrupted so a rollback was performed;
  - **unknown** – the last update status could not be determined.
- **Current.Loaded** – the date of the last successful update, in ISO 8601 format;
- **Current.Records** – the number of signatures in the anti-virus database currently in use;
- **Current.Released** – the date in ISO 8601 format when the last update was released.
- **all** – all information described above;
- **none** – do not publish any information over SNMP.

Kaspersky Anti-Virus employs an SNMP subagent that interacts with the SNMP master agent via *AgentX* protocol. The AgentX protocol parameters are as follows:

- **Socket** – interaction socket; you can use a local file or network socket as shown in the example:

```
Socket=local:/var/agentx/master
```

or

```
Socket=inet:705@127.0.0.1
```

### Warning!

If the local Unix socket is used, make sure that the subagent and the master agent can access it. This may imply changing the **RunAsUser** and **RunAsGroup** settings, as well as the access rights of a socket and data files used by the service (and the central service, if they are both on same machine).

- **Timeout** – time-out (in seconds) for an AgentX request. The default value is **5**.
- **Retries** – number of retries for an AgentX request. The default value is **10**. If this parameter is not set, the application will use value **5**.

**Warning!**

Actual number of retries may differ with the **Retries** value specified. This occurs because of the *watchdog* activity and is not an issue.

- **PingInterval** – time interval (in seconds) between subagent attempts to connect to master agent if it becomes disconnected.

You can use any SNMP agent that supports the AgentX protocol as a master agent. The following section gives a configuration example for *NET-SNMP* agent, in which the application subagent uses local socket to connect to NET-SNMP.

**Warning!**

You are advised to use NET-SNMP version 5.1.2 or higher which correctly implements the AgentX protocol.

To configure the master agent, please perform these steps:

1. Add the following lines to the *snmpd.conf* configuration file:

```
master agentx
AgentXSocket /var/agentx/master
AgentXPerms 770 770 root klusers
rocommunity public localhost
trapsink localhost
```

or, if a network socket is used, change the second line to:

```
AgentXSocket tcp:127.0.0.1:705
```

2. Add the following lines to the *snmp.conf* configuration file:

For Linux:

```
mibdirs +/opt/kaspersky/kav4lms/share/snmp-mibs
mibs all
```

For FreeBSD:

```
mibdirs +/usr/local/share/kav4lms/snmp-mibs/
mibs all
```

where the path */opt/kaspersky/kav4lms/share/snmp-mibs* specifies the default directory where the MIB files for Kaspersky Anti-Virus are stored. If the application was installed into another directory, change this path accordingly.

3. Restart *NET-SNMP*.

**Note:**

You will find more information about *NET-SNMP* at <http://www.net-snmp.org/>. For more information about *snmpd.conf* and *snmp.conf* configuration files, please see the corresponding manual pages.

The product OIDs are accessible under the following branch:

**.1.3.6.1.4.1.23668.1043**

or, in symbolic form:

**iso.org.dod.internet.private.enterprises.kaspersky.kav4lms**

This node contains the following groups:

- **config** – application configuration parameters, including groups configuration, divided into sections as in configuration files;
- **statistics** – statistical information about processed messages, resources in use and detected viruses;
- **update** – application update information;
- **admin** – administrative information (application start time, errors etc.).

**Warning!**

To get parameter values for objects in the **config.Groups** section, use the *Walk* method instead of *Get*.

The Administrator can also set the application to send SNMP-traps in case of specific events. The **SNMPTraps** parameter, in the **[kav4lms:server.snmp]** section of the *kav4lms.conf* configuration file, determines the events which should trigger the sending of SNMP traps by the application. The possible values are:

- **config** – a SNMP-trap is sent when the configuration or the databases are reloaded (*ConfigReloaded trap* and *BasesReloaded trap*);
- **admin** – a SNMP-trap is sent when the application starts or stops (*ProductStart trap*, *ProductStop trap*) or has a fatal error (*ProductError trap*). Additionally, if the *AlertThreshold* parameter value is not set to zero, an SNMP-trap will be sent if the percentage of infected messages found during the last hour exceeds the specified value (*AlertThreshold*). An *AlertThreshold trap* will be sent every hour since the threshold was exceeded until the percentage of infected messages falls below the defined limit.

**Note:**

There is the **ConfigReloaded** trap corresponding to the application reload. However, the **ProductStart**, **ProductStop** and **BasesReloaded** traps are also sent in this case. This occurs because the watchdog warm-restarts the application.

- **update** – a SNMP-trap is sent when the application update is performed (*UpdateStatus trap*) or the anti-virus database is older than five days (*ObsoleteBases trap*);
- **all** – SNMP-trap is sent when any of the above described events occurs;
- **none** – no SNMP-traps are sent.

**Warning!**

If you use NET-SNMP master agent, you should start *snmptrapd* daemon to receive traps.

## 10.2. Using the application's setup script

Kaspersky Anti-Virus provides a special script, allowing managing the application after it has been installed.

The setup script is used as follows:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-setup.sh <option>
```

The available options are:

- `--install-services` - register central and filter services with the existing system;
- `--remove-services` - unregister all services (will not be started/stopped with the system);
- `--check-services` - verify if the application's services are registered;
- `--install-filter=<MTA>` - register the specified filter into MTA configuration. Note this also registers it as a service (where applicable);
- `--remove-filter=<MTA>` - unregister the filter service from the specified MTA;

- `--remove-filters` - remove all filters that are found active in MTA config(s);
- `--check-filter=<MTA>` - verify if the changes for registering with the MTA were done;
- `--filter-options=<options>` - sets filter specific options. This option is used only with the `--install-filter` option in order to specify the filter specific parameters. For Sendmail the following options are available: **tempfail**, **reject**, **pass**;
- `--install-cron=<component_name>` - install cron job for specified component;
- `--remove-cron=<component_name>` - remove cron job for specified component;
- `--check-cron=<component_name>` - check if the cron job for component is registered;
- `--user=<user_name>` - specify the user name whose credentials will be used to run the central service and application filter. When used together with the `--install-cron` and `--remove-cron` parameters, the option defines the user account, for which a schedule is created.

For example:

```
# /opt/kaspersky/kav4lms/bin/kav4lms-setup.sh \  
--install-cron=updater --user=root
```

or

```
# /opt/kaspersky/kav4lms/bin/kav4lms-setup.sh \  
--install-cron=updater --user=qmailq
```

- `--add-components-info` - add product specific options to *applications.setup* file;
- `--del-components-info` - remove components specific options from application registry;
- `--check-components-info` - verify if the product components options are present;
- `--install-webmin-module` - add web based administration module to Webmin;
- `--remove-webmin-module` - remove the module from Webmin;
- `--check-webmin-module` - verify if the Webmin module is installed;



- `--register-key=key-id` - register a key by full path, or id relative to *licenses* directory;
- `--group=<group_name>` - specify the group that will be used to run Kaspersky Anti-Virus; the option modifies the **Group** value in the **[options]** section of the application configuration file;
- `--switch-credentials=<user_name>[,<group_name>]` - specify the user and (if entered) group whose credentials will be used to start the central service and filter of the application. The option modifies the **RunAsUser** and (if entered) **RunAsGroup** values in the **[kav4lms:server.settings]** and **[kav4lms:filter.settings]** sections of the application configuration file. If this option is used, the central service and the filter of the application will be restarted.

The `<MTA>` parameter specifies the MTA to integrate with. Available values are:

- **exim** – post-queue integration with Exim;
- **exim-dlfunc** – pre-queue integration with Exim using dynamically loaded library;
- **postfix** – post-queue integration with Postfix;
- **qmail** – integration with qmail;
- **sendmail-milter** – integration with Sendmail.

The `<component_name>` parameter specifies the application's component name. The available option is **updater**.

**Note:**

All `--check` options are silent and return 0 for presence of the checked item or non-0 value for its absence.

## 10.3. Managing the application from the command line

Kaspersky Anti-Virus provides a command line management tool *kav4lms-cmd*, stored in the `/opt/kaspersky/kav4lms/bin` directory.

**Warning!**

The *ka4lms-cmd* tool requires running central service of the application.

This tool's command line options are divided into two categories:

1. General application options. This includes:
  - `-v` or `--version` - display program version
  - `-h` or `--help` - display inline help message
  - `-m` or `--module <argument>` - selects a specific module for further commands; available options for module are: `config`, `filter`, `kavmd`, `statistics`, `update`
  - `-c` or `--config <argument>` - specify a configuration file other than the default one
  - `-l` or `--list` - list available modules
2. Module-specific options.
  - a) **Config** module. This module modifies the application's configuration files by querying and setting the configuration keys:
    - `-q <key>` - query the value of a configuration key. , e.g. `-q Path.TempPath`;
  - b) **Filter** module. This module manages the filter component. The available option is:
    - `-x <command>` - invoke a filter component command; available options are: `start`, `stop`, `restart`, `status`, `test-service`.
  - c) **Central service (kavmd)** module. This module manages the central application service. The available option is:
    - `-x <service-command>` - invoke a central service command; available options are: `start`, `stop`, `restart`, `reload`, `status`, `test-service`.
  - d) **Statistics** module. This module manages application's statistics. The available options are:
    - `-x <stats-command>` - invoke a statistics command; available options are: `write`, `reset`.
  - e) **Update** module. This module manages the `kav4lms-keepup2date` component:
    - `-e <event-name>` - specify the delivery of a certain event, options are: `OnUpdated`, `OnNotNeeded`, `OnError`, `OnRolledback`, `OnUnknown`.

## 10.4. Additional informational fields in messages

The application enables some supplementary information to be added to mail messages as header fields using one of two separate methods:

- Addition of an extension header field to mail message.

The information may indicate the application version, the date when the anti-virus database was last updated, the time and result of message scanning (determined by the **AddXHeaders** parameter in the **[kav4lms:groups.<group\_name>.settings]** section of the group configuration file).

Header format:

```
X-Anti-Virus: <product name and version>, bases:
<date of the last update to anti-virus databases in
YYYYMMDDTHHMMSS format> #<the number of records in AV
databases>, check: <scan date in YYYYMMDD format>
<scanning status or notchecked>
```

where:

YYYY stands for the year in four-digit format;

MM – month;

DD – date;

HH – hour;

MM – minute;

SS - second.

For example:

```
X-Anti-Virus: Kaspersky Anti-Virus for Linux Mail
Server 5.6.17/RELEASE build 4,
bases: 20080415 #705877, check: 20080415 clean
```

- Addition of disclaimer text to mail message body.

The information will be added as plain text; it may contain any statement generated in accordance with the security policy (or other rules) of a specific organization, and is specified by the **AddDisclaimer** parameter in the **[kav4lms:groups.<group\_name>.settings]** section. The default message text notifies that the message has been scanned by Kaspersky Anti-Virus. Upon the administrator's demand the

application can modify the information format (e.g., generate disclaimer message as a HTML text).

- Deleted message part replacement.

When message is being processed, its parts can be deleted according to the action chosen. The deleted parts can be replaced with a notice about the reason. For this purpose set the **UsePlaceholderNotice** (in **[kav4lms:groups.<group\_name>.settings]** section of the group configuration file) parameter to **yes**. If the **UsePlaceholderNotice** value is **no**, then the respective parts will be completely removed from the message, leaving it like that they never existed.

The notice text is taken from a template file named *part\_<action\_taken>*, which also supports notification macros (see 5.7 on p. 57 for details).

## 10.5. Localization of displayed date and time format

While operating, Kaspersky Anti-Virus compiles reports for each of its components as well as various notifications for users and administrators. Such information is always supplemented with the date and time of its output.

By default, Kaspersky Anti-Virus uses the date and time formats corresponding to the strftime standard:

**%H:%M:%S** – format of time output (**hh.mm.ss**).

**%d-%m-%y** – format of date output (**dd.mm.yy**).

The administrator may change the date and time format. Localization of formats is performed in the **[locale]** section of the *kav4lms.conf* configuration file. You can define the following formats:

**%I:%M:%S %P** – for time output in twelve-hour format (**TimeFormat** parameter).

**%y/%m/%d** and **%m/%d/%y** – for date output (**DateFormat** parameter) (**yy.mm.dd** and **mm.dd.yy** respectively).

# CHAPTER 11. TESTING THE APPLICATION

After Kaspersky Anti-Virus is installed and configured, you are advised to verify the correctness of its operation using a test "virus" and its modifications.

This test "virus" was specially designed by **eicar** (The European Institute for Computer Antivirus Research) for testing anti-virus products.

The test "virus" IS NOT A VIRUS because it does not contain code that can harm your computer. However, most anti-virus products manufacturers identify this file as a virus.

## **Warning!**

**Never use real viruses for testing the operation of an anti-virus product!**

You can download this test "virus" from the official website of the **EICAR** organization at [http://www.eicar.org/anti\\_virus\\_test\\_file.htm](http://www.eicar.org/anti_virus_test_file.htm).

## **Note:**

You will have to disable anti-virus protection before the download because the *anti\_virus\_test\_file.htm* file will be identified and processed by the anti-virus solution installed on the computer as an infected object transferred via HTTP.

Please note that anti-virus protection should be enabled immediately after you download the test "virus".

The file downloaded from the **EICAR** website or created as described above contains the body of a standard test "virus". Kaspersky Anti-Virus will detect it, assign it the **Infected** non-disinfectable status and apply the action defined by the administrator for processing objects of this type.

To test the response of Kaspersky Anti-Virus when other types of objects are detected, modify the content of this standard test "virus" by adding one of the prefixes (see Table below). You can edit the text in any text editor.

## **Note:**

You can verify proper functioning of the application using modifications of the EICAR test "virus" only if your anti-virus database was last updated on or after October 24, 2003 (contains the cumulative updates for October 2003).

Table. Modifying the test "virus"

Prefix	Object type
No prefix, standard test "virus"	<b>Infected.</b> Non-disinfectable object.
CORR–	<b>Corrupted.</b>
SUSP–	<b>Suspicious</b> (unknown virus code).
WARN–	<b>Suspicious</b> (modified code of a known virus).
ERRO–	<b>Not analyzed</b> due to an error.
CURE–	<b>Disinfected.</b> The object will be disinfected; the text of the "virus" body will be replaced with the word "CURE".

The first table column lists the prefixes to be added at the beginning of the string of the standard test "virus".

After adding a prefix to the test "virus", save it to a file with another name, for example *eicar\_corr.com*.

The second column of this table contains the types of objects identified by the anti-virus application after you have added the prefix. The actions for each type of object are defined by the application settings configured by the administrator.

# APPENDIX A. ADDITIONAL INFORMATION

## A.1. Application configuration file *kav4lms.conf*

The package of Kaspersky Anti-Virus includes the *kav4lms.conf* configuration file containing application settings. This section contains a detailed explanation of the configuration file settings including their default values after standard installation of the product.

Configuration file consists of sections describing individual aspects of application functionality. Each section uses the following syntax: the first line contains section header in **[section\_name]** format followed by descriptions of section parameters.

### Note:

For Boolean parameter values that can be set to **true|false** the configuration file also supports equivalent values: **yes|no, y|n, or 1|0**.

Numeric parameters have the upper limit **UINT\_MAX=4294967295**.

### Warning!

Parameters marked in the description as “mandatory” are required for normal operation of the application. They must be specified; otherwise, the Anti-Virus will not function!

### A.1.1. Section *[kav4lms:server.settings]*

The **[kav4lms:server.settings]** section contains parameters for the central application's service:

**RunAsUser** – name of the account whose privileges are used to run the central service.

Mandatory parameter.

The default value is **kluser**.

**Note:**

If the filter and central service are installed on the same computer, make sure that the **RunAsUser** parameter has the same value for both those components as it will allow correct access to shared files.

**RunAsGroup** – name of the group whose privileges are used to run the central service.

Mandatory parameter.

The default value is **klusers**.

**ServiceSocket=inet:<port>@<ip-address>|local:<path\_to\_socket>** – the local or network socket which is used by Kaspersky Anti-Virus filter service to interact with the central service of the application (endpoint of the central service – filter connection).

**Warning!**

The central application's service must be stopped before modification of this parameter. After its modification, start the service to apply the new value.

Syntax:

`ServiceSocket=inet:<port>@<ip-address>` - for a network socket

`ServiceSocket=local:<path_to_socket>` - for a local socket.

where:

- **<port>**: interaction port;
- **<ip-address>**: IP address;
- **<path\_to\_socket>**: path to local socket.

Mandatory parameter.

The default value is **local:/var/run/kav4lms/kavmd.sock**.

**Note:**

If a local socket is used, make sure that the directory where socket file is located and the file itself are accessible for reading and writing both to the filter service and to the central application service.

**ServiceSocketPerms** – permissions for the **ServiceSocket** if a local socket is used. The socket's owner is defined by the **RunAsUser:RunAsGroup** pair of parameters.

The default value is **0600** (it is used if no parameter value is specified).



**AdminSocket** – the local socket which is used for administering the central service (for example, via SNMP). The central service can be controlled by administrative commands, and can also service request for information from the SNMP component. The dialog is done on this specified socket.

**Warning!**

The central application's service must be stopped before modification of this parameter. After its modification, start the service to apply the new value.

Mandatory parameter.

The default parameter value is

**local:/var/run/kav4lms/kavmdctl.sock.**

**Warning!**

When selecting this setting, make sure that socket file and socket folder are write accessible only for the user under which account the application runs.

**AdminSocketPerms** – permissions for the **AdminSocket**. The socket's owner is **RunAsUser:RunAsGroup**.

The default value is **0600**.

**MaxWatchdogRetries=0...UINT\_MAX** – maximum number of retries to restart the Kaspersky Anti-Virus using *watchdog*. The value of **-1** (minus one) corresponds to the unlimited number of retries. The value of **0** disables watchdog.

The default value is **10**.

**MaxClientRequests=0...UINT\_MAX** – the maximum number of client requests accepted and treated by the central service. If the parameter is **0**, the number of requests is unlimited.

The default value is **20**.

**MaxScanRequests=0...UINT\_MAX** – the maximum number of requests for scanning messages. If the parameter is **0**, the number of requests is unlimited.

The default value is **0**.

**LicensedUsersDomains** – list of domains containing accounts, which should be protected, according to the licensing scheme of Kaspersky Anti-Virus for Linux Mail Server. This option is available only if your key is issued for a certain number of mail addresses. You can specify several comma-delimited values.

The default value is **localhost, localhost.localdomain**.

## A.1.2. Section *[kav4lms:server.log]*

The **[kav4lms:server.log]** section contains parameters for the central service's log:

**Options=<functionality\_category>.<details\_level>** – category of events registered in log, where:

- **<functionality\_category>** can take one of the following values: **all**, **config**, **app**, **scan**, **cfilter**, **backup**, **notif**, **admin**, **smtp** (see section 9.1 on page 85).
- **<details\_level>** can take one of the following values: **debug**, **activity**, **info**, **warning**, **error**, **fatal** (see section 9.1 on page 85).

You can specify several comma-delimited levels.

E.g.:

```
Options = backup.all, config.error, \
scan.all, -scan.debug
Options = backup.all, config.E, \
scan.all, -scan.9
```

Mandatory parameter.

The default value is **all,-all.debug**.

**Destination=syslog:<name>@<category>|file:<path\_to\_file>** – path to the file where information about activity of the central application service will be logged:

- **syslog:<name>@<facility>**: write report to system log; **<name>** defines the application name; **<facility>** defines the logged category.
- **file:<path\_to\_file>**: write report to file located at the specified path.

Mandatory parameter.

The default value is **syslog:kavmd@mail**.

**Append=yes|no** – specifies how the information will be added to the log file:

- **yes** – add new information to the existing file.
- **no** – create a new log file every time the application starts.

The default value is **yes**.

**RotateRounds=0...UINT\_MAX** – number of report files created during rotation. When this number is reached, the application starts to overwrite the oldest one. If this number is non-zero, the rotation is enabled.

The default value is **10**.

**RotateSize=1M** – report file size in bytes. When it is reached, a new report file is created.

The default value is 1M.

**Warning!**

The **Append**, **RotateRounds** and **RotateSize** parameters are in effect only when log destination is a file.

### A.1.3. Section [*kav4lms:server.statistics*]

The [*kav4lms:server.statistics*] section contains parameters for the central service statistics:

**Options=none|all|messages|resources|viruses|filters|raw** – data category for logging purposes (see section 9.2 on page 87). You can specify several comma-delimited categories.

E.g.:

```
Options=none, raw
```

Mandatory parameter.

The default value is **none**.

**Format=xml|txt** – specifies the statistics file format.

The default value is **xml**.

**Destination=file:<path\_to\_file>** – the destination for the central service logging. The current version of Kaspersky Anti-Virus supports file destination only.

The default value is:

**file:/var/opt/kaspersky/kav4lms/stats/statistics.xml** (in Linux)

**file:/var/db/kaspersky/kav4lms/stats/statistics.xml** (in FreeBSD).

**RawDestination= file:<path\_to\_file>** – the destination for the raw (or per-message) statistics. The current version of Kaspersky Anti-Virus supports file destination only.

Mandatory parameter.

The default value is:

**file:/var/opt/kaspersky/kav4lms/stats/statistics.raw** (in Linux)

**file:/var/db/kaspersky/kav4lms/stats/statistics.raw** (in FreeBSD).

## A.1.4. Section *[kav4lms:server.snmp]*

The **[kav4lms:server.snmp]** section contains parameters defining the interaction with application via the SNMP protocol:

**SNMPServices=config|statistics|admin|update|all|none** – application information that can be read over SNMP:

- **config**: information about all settings in all sections of the application configuration file;
- **statistics**: summarized statistical information about application activity;
- **admin**: information related to application runtime (startup time, up-time, etc.);
- **update**: information about update of the anti-virus databases (date of the last update, number of records in databases, etc.);
- **all**: all statistical information and data on configuration of the application;
- **none**: access to information over SNMP is disabled.

You can define several values as a list, each parameter must be entered in a separate line.

E.g.:

```
SNMPServices=config
```

```
SNMPServices=admin
```

Mandatory parameter.

The default value is **none**.

**SNMPTraps=config|admin|update|all|none** – list of events which trigger a notification to administrator via SNMP-traps.

- **config**: upon modification of application configuration or successful update of the anti-virus databases.

- **admin**: when application starts or stops or critical errors occur in its operation and also when infected objects are detected triggering the condition defined by the **AlertThreshold** parameter.
- **update**: upon update of the anti-virus databases irrespectively of its result;
- **all**: when any of the events listed above occurs;
- **none**: SNMP traps are disabled.

You can define several values as a list, each parameter must be entered in a separate line.

E.g.:

```
SNMPTraps=config
SNMPTraps=admin
```

Mandatory parameter.

The default value is **none**.

**AlertThreshold=0...100** – threshold percentage of infected messages in all messages scanned during the last hour which when exceeded will trigger an SNMP-trap sent by the application (in case the **SNMPTraps** parameter is set to **admin**).

The default value is **10**.

**Socket** – the socket used to interact with master agent; a local or network socket can be used.

Syntax:

```
inet:<port>@<ip-address> - for a network socket.
local:<path_to_socket> - for a local socket.
```

Where:

- **<port>**: interaction port.
- **<ip-address>**: IP address.
- **<path\_to\_socket>**: path to local socket.

**Note:**

For a local socket you should specify a file named «*master*», that is a naming constraint of SNMP. Therefore an absolute path should be specified as **<path\_to\_socket>** including the name of the «*master*» file.

The default value is- **inet:705@127.0.0.1**.

**Timeout=0...UINT\_MAX** – timeout (in seconds) for requests sent to master agent.

The default value is 5.

**Retries=0...UINT\_MAX** – number of attempts for requests sent to master agent.

The default value is **10**.

**Warning!**

Actual number of retries may differ with the **Retries** value specified. This occurs because of the *watchdog* activity and is not an issue.

**PingInterval=0...UINT\_MAX** – time interval (in seconds) between attempts by the subagent to connect to the master agent, if the connection fails.

The default value is **30**.

## A.1.5. Section

### ***[kav4lms:server.notifications]***

The **[kav4lms:server.notifications]** section contains notification options:

**ProductAdmins** – e-mail address of the Kaspersky Anti-Virus administrator. You can specify several addresses, separated with a comma.

The default value is **postmaster**.

**ProductNotify=fault|update|license|all|none** – notify the administrator of Kaspersky Anti-Virus upon the events specified:

- **fault** – critical errors;
- **update** – results of anti-virus database updates;
- **license** – product key expiry and situations when the license restrictions in the product key are exceeded;
- **all** – all events;
- **none** – notifications are disabled.

Several comma-delimited values can be specified.

Mandatory parameter.

The default value is **all**.

**Subject** – header of the standard notification added to the **Subject** field.

The default value is **Anti-virus notification message**.

**Charset** – character set to be used in sent notifications.

The default value is **us-ascii**.

**TransferEncoding** – value of the notification encoding algorithm. The default value is **7bit**.

**NotifierRelay** – specifies the notifications' MTA address.

Syntax:

```
NotifierRelay=<protocol>:<host>:<port>
```

The default value is **smtp:127.0.0.1:25**.

**NotifierQueue** – the directory under which the notifications' MTA stores its queue and management files.

The default value is:

**/var/opt/kaspersky/kav4lms/nqueue/** (in Linux)

**/var/db/kaspersky/kav4lms/nqueue/** (in FreeBSD).

**NotifierTimeout=0...UINT\_MAX** – timeout (in seconds) for sending notifications. The default value is **5**.

**NotifierPersistence=yes|no** – specifies whether the connection to the notifications' MTA is persistent.

**Templates** – the directory containing templates for product admin's notifications.

The default value is:

**/etc/opt/kaspersky/kav4lms/templates-admin/en** (in Linux),

**/usr/local/etc/kaspersky/kav4lms/templates-admin/en** (in FreeBSD).

## A.1.6. Section *[kav4lms:filter.settings]*

The **[kav4lms:filter.settings]** section contains parameters for the filter service of Kaspersky Anti-Virus:

**RunAsUser** name of the account whose privileges are used to run the filter service.

Mandatory parameter.

The default value is **kluser**.

**Note:**

If the filter and central service are installed on the same computer, make sure that the **RunAsUser** parameter has the same value for both those components as it will allow correct access to shared files.

**RunAsGroup** – name of the group whose privileges are used to run the filter service.

Mandatory parameter.

The default value is **klusers**.

**FilterSocket=inet:<port>@<ip-address>|local:<path\_to\_socket>** – the local or network socket which is used by the Kaspersky Anti-Virus filter service to interact with the central service of the application (endpoint of the central service – filter connection).

**Warning!**

The filter service must be stopped before modification of this parameter. After its modification, start the service to apply the new value.

Syntax:

`FilterSocket=inet:<port>@<ip-address>` - for a network socket

`FilterSocket=local:<path_to_socket>` - for a local socket.

where:

- **<port>**: interaction port;
- **<ip-address>**: IP address;
- **<path\_to\_socket>**: path to local socket.

Mandatory parameter.

The default value is **inet:10025@127.0.0.1**.

**Note:**

If a local socket is used, make sure that the directory where socket file is located and the file itself are accessible for reading and writing both to the filter service and to the central application service.

**FilterSocketPerms** – permissions for the **FilterSocket**, if a local Unix socket is used. The socket's owner is **RunAsUser:RunAsGroup**.

The default value is **0660**.

**ServiceSocket=inet:<port>@<ip-address>|local:<path\_to\_socket>** – the local or network socket which is used by the Kaspersky Anti-Virus filter service to interact with the central service of the application (endpoint of the central service – filter connection). Record format is identical to **FilterSocket**.



**Warning!**

The filter service must be stopped before modification of this parameter. After its modification, start the service to apply the new value.

Mandatory parameter.

The default value is **local:/var/run/kav4lms/kavmd.sock**.

**AdminSocket=local:<path\_to\_socket>** – the local socket which is used to manage filter service (for example, via SNMP). The filter service can be controlled by administrative commands. The dialog is done on this specified socket.

**Warning!**

The filter service must be stopped before modification of this parameter. After its modification, start the service to apply the new value.

Mandatory parameter.

The default value is **local:/var/run/kav4lms/kavmdctl.sock**.

**Warning!**

While setting the parameter, make sure that only the account used to run the application has write permissions for the socket file and the socket directory.

**AdminSocketPerms=0600** – permissions for the **AdminSocket**. The socket's owner is **RunAsUser:RunAsGroup**.

**ForwardSocket=inet:<port>@<ip-address>|local:<path\_to\_socket>** – the local or network socket which is used by Kaspersky Anti-Virus filter to interact with MTA (endpoint of the application – MTA connection).

**Warning!**

The filter service must be stopped before modification of this parameter. After its modification, start the service to apply the new value.

Record format is identical to **FilterSocket**.

Mandatory parameter.

The default value is **inet:10026@127.0.0.1**.

**Note:**

The **ForwardSocket** parameter is used for integration with Postfix and Exim.

**FilterTimeout=0...UINT\_MAX** – timeout (in seconds) for communication between the filter service and MTA. If no data / commands is send during

the time specified here, Kaspersky Anti-Virus will close connection to MTA.

The default value is **600**.

**FilterThreads=0...UINT\_MAX** – the number of threads, spawned by the filter service to listen to the MTA requests.

The default value is **10**.

**MaxMilterThreads=0...UINT\_MAX** – the maximum number of threads, run by the Milter library simultaneously. The value of 0 specifies the unlimited number of threads.

The default value is **0**.

**Warning!**

Applies for Sendmail only!

## A.1.7. Section *[kav4lms:filter.log]*

The **[kav4lms:filter.log]** section contains parameters for the filter service's log:

**Options=<functionality\_category>.<details\_level>** – category of filter events registered in log, where:

- **<functionality\_category>** can take one of the following values: **all**, **config**, **app**, **scan**, **cfilter**, **backup**, **notif**, **admin**, **smtp** (see section 9.1 on page 85).
- **<details\_level>** can take one of the following values: **debug**, **activity**, **info**, **warning**, **error**, **fatal** (see section 9.1 on page 85).

You can specify several comma-delimited levels.

Mandatory parameter.

The default value is **all,-all.debug**.

**Destination=syslog:<name>@<category>|file:<path\_to\_file>** – path to the file where information about filter service activity will be logged:

- **syslog:<name>@<facility>**: write report to system log; **<name>** defines the application name; **<facility>** defines the logged category;
- **file:<path\_to\_file>**: write report to file located at the specified path.

Mandatory parameter.

The default value is **syslog:kav4lms-filters@mail**.

**Append=yes|no** – specifies how the information about filter activity will be added to the log file:

- **yes** – add new information to the existing file;
- **no** – create a new log file every time the application starts.

The default value is **yes**.

**RotateRounds=0...UINT\_MAX**– number of report files created during rotation. When this number is reached, the application starts to overwrite the oldest one. If this number is non-zero, the rotation is enabled.

The default value is **10**.

**RotateSize=1M** – report file size in bytes. When it is reached, a new report file is created.

The default value is **1M**.

### **Warning!**

The **Append**, **RotateRounds** and **RotateSize** parameters are in effect only when log destination is a file.

## **A.1.8. Section [*kav4lms:groups*]**

The [**kav4lms:groups**] section contains references to the groups' configuration files:

**\_includes=<path\_to\_directory>** – path to the directory where configuration files for groups are stored. The directory path should be relative to the location of the main configuration file of the application.

Mandatory parameter.

The default value is **groups.d/**.

## **A.1.9. Section [*path*]**

The [**path**] section contains parameters that define the paths to critical directories.

**BasesPath** – full path to the directory containing anti-virus databases.

Mandatory parameter.

The default value is **`/var/opt/kaspersky/kav4lms/bases`** (in Linux) or **`/var/db/kaspersky/kav4lms/bases`** (in FreeBSD).

**LicensePath** – full path to the directory where keys are stored.

The default value is **`/var/opt/kaspersky/kav4lms/bases`** (in Linux) or **`/var/db/kaspersky/kav4lms/bases`** (in FreeBSD).

**PidPath** – path to the central application's service PID file.

Mandatory parameter.

The default value is **`/var/run/kav4lms/`**.

**TempPath** – path to the directory containing temporary files. The application creates `.kav4lms-<id>` subdirectories at the specified path.

Mandatory parameter.

The default value is **`/var/tmp/`**.

**iCheckerDBFile** – path to the iChecker™ database.

Mandatory parameter.

The default value is **`/var/opt/kaspersky/kav4lms/iChecker.db`** (in Linux) or **`/var/db/kaspersky/kav4lms/iChecker.db`** (in FreeBSD).

## A.1.10. Section `[locale]`

The `[locale]` section contains options for displaying the date and time in the reports and statistics.

**DateFormat** – date format displayed in the report.

Mandatory parameter.

The default value is **`%d-%m-%Y`**.

**TimeFormat** – time format displayed in the report.

Mandatory parameter.

The default value is **`%H:%M:%S`**.

### Note:

You can change the time format to twelve hours (am, pm): **`!l:%M:%S %P`**.

**Strings** – the path to file containing the string constants used by the application. The directory path should be relative to the location of the main configuration file of the application.

Mandatory parameter.

The default value is **locale.d/strings.en**.

## A.1.11. Section *[options]*

The **[options]** section contains various application parameters not included into other groups:

- **User** – system account used to run the application components.  
Mandatory parameter.  
The default value is **kluser**.
- **Group** – system group used to run the application components.  
Mandatory parameter.  
The default value is **klusers**.

## A.1.12. Section *[updater.path]*

The **[updater.path]** section defines the paths to directories used for updating.

**BackUpPath=/var/opt/kaspersky/kav4lms/bases.backup/** – full path to the directory for backup storage of the anti-virus databases.

## A.1.13. Section *[updater.options]*

The **[updater.options]** section contains parameters defining update options.

**UpdateComponentsList** – list of components, which will be updated.

The default value is **AVS, AVS\_OLD, CORE, Updater, BLST**.

**RetranslateComponentsList** – list of components for which updates will be saved to a network directory.

If the parameter value is empty (default), the **UpdateComponentsList** parameter value will be used.

**KeepSilent=yes|no** – defines whether the application should display a report about an update to the console. If set to **yes**, reports are not sent to the console.

The default value is **no**.

**UseUpdateServerUrl=yes|no** – defines whether the application should use the Kaspersky Lab server URL defined by **UpdateServerUrl** parameter as the update source.

The default value is **no**.

**UpdateServerUrl=http://url|ftp://url|/local\_path/** – the address of the server used as a source for updating.

The default parameter value is empty.

**UseUpdateServerUrlOnly=yes|no** – defines whether the application should use only the URL specified by **UpdateServerUrl** to update the database. If this option is set to **no**, then whenever updating from the **UpdateServerUrl** address fails the application will use an alternative address from the list of update servers.

The default value is **no**.

**RegionSettings** – defines the customer region used to update the anti-virus databases from the nearest Kaspersky Lab's update server.

The default value is **ru**.

**ConnectTimeout** – interval (in seconds) within which the application will attempt to connect to the update source.

The default value is **30**.

**ProxyAddress** – IP-address of a proxy server if one is required for Internet connection.

By default, the value is not set.

**UseProxy=yes|no** – use a proxy-server to connect to one of update servers. If the parameter is **no**, the proxy server will not be used. If the parameter is **yes**, the proxy server address defined by the **ProxyAddress** parameter will be used.

The default value is **no**.

**PassiveFtp=yes|no** – whether to use passive FTP mode when downloading updates via FTP.

The default value is **yes**.

**Index=u0607g.xml** – file containing the main index of the update system used to choose the set of updates on the servers of Kaspersky Lab. Modification of that value is not recommended.

**IndexRelativeServerPath=index/6** – path to the file containing the main update system index. The path should be relative to the location of the main configuration file of the application. Modification of that value is not recommended.

## A.1.14. Section *[updater.report]*

The **[updater.report]** section contains update report parameters.

**Append=yes|no** – determines how the activity of the *kav4lms-keepup2date* component should be logged:

- **yes** – append new information to the existing file;
- **no** – create a new log file each time the component starts. Then the log file will only contain information about the results of the last update.

The default value is **yes**.

**ReportFileName** – name of the *kav4lms-keepup2date* report file.

The default value is **/var/log/kaspersky/kav4lms/keepup2date.log**.

**ReportLevel=0|1|2|3|4|9** – the level of details in update report (**0** – Fatal, **1** – Error, **2** – Warning, **3** – Info, **4** – Activity, **9** – Debug). The default value is: **3**.

## A.1.15. Section *[updater.actions]*

The **[updater.actions]** section contains parameters defining the actions, taken upon specific *keepup2date* events.

**OnAny** – specifies the command, executed whenever an event occurs. By default other application components are notified about the event.

The default value is **/opt/kaspersky/kav4lms/bin/kav4lms-cmd -m \update -e %EVENT\_NAME%** (in Linux),  
**/usr/local/bin/kav4lms-cmd -m update -e %EVENT\_NAME%** (in FreeBSD).

**OnStarted** – specifies the command, executed when the *kav4lms-keepup2date* component starts.

The value is empty by default.

**OnUpdated** – specifies the command, executed when the update completes successfully.

The default value restarts the application –

**/opt/kaspersky/kav4lms/bin/kav4lms-cmd -x bases** (in Linux),  
**/var/db/kaspersky/kav4lms/bin/kav4lms-cmd -x bases** (in FreeBSD).

**OnRetranslated** – command performed after database updates are successfully downloaded from a network directory to the directory where the anti-virus databases are located.

The value is empty by default.

**OnNotUpdated** – specifies the command, executed if the update was not performed.

The value is empty by default.

**OnFailed** – specifies the command, executed when the update has failed.

The value is empty by default.

**OnRolledBack** – specifies the command, executed when a rollback occurs.

The value is empty by default.

**OnBasesCheck** – specifies the command, executed after the update to validate the anti-virus bases. The *avbasestest* utility is used to check the integrity of the anti-virus databases by default. It checks the updates downloaded from source and stored in a temporary directory. If the updates are not corrupted, they are copied from the temporary location to the directory storing the anti-virus databases.

**Note:**

The *avbasestest* utility starts automatically, it requires no user participation.

The default value is **/opt/kaspersky/kav4lms/lib/bin/avbasestest %TEMP\_BASES\_PATH% %BASES\_PATH%** (in Linux),  
**/usr/local/libexec/kaspersky/kav4lms/avbasestest %TEMP\_BASES\_PATH% %BASES\_PATH%** (in FreeBSD).

**Note:**

The *avbasestest* actions support the following macros:

- **%EVENT\_NAME%** - the name of the event that triggered this command;
- **%BASES\_PATH%** - when applicable, the path to existing bases;
- **%TEMP\_BASES\_PATH%** - when applicable, the path to temporary dir where bases are being updated;
- **%AVS\_UPDATE\_DATE%** - date of the event in **mm:dd:yyyy hh:mm:ss** format.



## A.1.16. Section *[scanner.display]*

The **[scanner.display]** section contains settings for printing the *kav4lms-kavscanner* report to the screen:

**ShowContainerResultOnly=true|false** – the mode of displaying the results of the archive scan on the screen. To display short format results, assign the value **true** to this setting. By default use expanded format of messages.

Mandatory parameter.

The default value is **false**.

**ShowObjectResultOnly=true|false** – the mode of displaying the results of the scan of a simple object on the screen. To display short format, assign value **true** to this setting. By default use expanded format of messages.

Mandatory parameter.

The default value is **false**.

**ShowOK=true|false** – mode for printing messages about clean files to the screen. In order to disable this mode assign value **false** to this setting.

Mandatory parameter.

The default value is **true**.

**ShowProgress=true|false** – mode of displaying on the screen information about the current component operation, including the process of downloading of the anti-virus database, or information about the scan of the current file. To disable this mode assign value **false** to this setting.

Mandatory parameter.

The default value is **true**.

## A.1.17. Section *[scanner.options]*

Section **[scanner.options]** contains settings of the *kav4lms-kavscanner* component:

**ExcludeDirs=mask1:mask2:...:maskN** – masks of the directories excluded from the scanning scope. They are defined as standard shell masks.

The default value is **/dev:/udev:/proc:/sys**.

**ExcludeMask=mask1:mask2:....:maskN** – masks of the files excluded from the scanning scope. By default all files are scanned. Masks are defined as standard shell masks.

The default value is **not defined**.

**Packed=true|false** – scanning mode for packed objects. To disable scanning, set the parameter to **false**.

Mandatory parameter.

The default value is **true**.

**Archives=true|false** – archives scan mode. To disable this mode assign value **false** to this setting.

Mandatory parameter.

The default value is **true**.

**Cure=true|false** – mode for disinfecting infected objects. In order to enable this mode assign value **true** to this setting.

Mandatory parameter.

The default value is **false**.

**Heuristic=true|false** – mode for using heuristic code analyzer during the scan. To disable this mode assign value **false** to this setting.

Mandatory parameter.

The default value is **true**.

**LocalFS=true|false** – mode for scanning only the local file system. In order to enable this mode assign value **true** to this setting.

Mandatory parameter.

The default value is **false**.

**MailBases= true|false** – e-mail database scan mode. In order to disable this mode assign value **false** to this setting.

Mandatory parameter.

The default value is **true**.

**MailPlain=true|false** – scan e-mail messages in plain text format. To disable this mode assign value **false** to this setting.

Mandatory parameter.

The default value is **true**.

**Packed=true|false** – packed files scan mode. To disable this mode assign value **false** to this setting.

Mandatory parameter.

The default value is **true**.

**Recursion=true|false** – mode for recursive scanning of directories during the anti-virus scan. In order to disable this mode assign value **false** to this setting.

Mandatory parameter.

The default value is **true**.

**SelfExtArchives=true|false** – self-extracting archives scan mode. In order to disable this mode assign value **no** to this setting. If the archive scan mode is enabled (**Archives=yes**), self-extracting archives will be scanned even if the **SelfExtArchives** setting is assigned the value **false**.

Mandatory parameter.

The default value is **true**.

**Ichecker=true|false** – the mode for the use of the iChecker technology during the anti-virus scan. In order to disable this mode assign value **false** to this setting.

The default value is **true**.

**MaxLoadAvg** – maximum CPU load. If the value is exceeded, the *kav4lms-kavscanner* component stops operation.

The value is empty by default.

**UseAVbasesSet=standard|extended** – the set of anti-virus databases used by the application while scanning. The **extended** set contains, in addition to records contained in the **standard** set, descriptions of riskware, such as adware and remote administration programs.

The default value is **standard**.

**FollowSymlinks=true|false** – the option controls handling of symbolic links. If the parameter is set to **true**, the application will follow the links that point to directories while scanning and check objects located at the corresponding addresses. To disable the mode, set the parameter to **false**.

The default value is **true**.

## A.1.18. Section **[*scanner.report*]**

Section **[*scanner.report*]** contains settings for generating reports about the results of the *kav4lms-kavscanner* component operation.

**Append=true|false** – mode for addition of new messages to the file containing report on the results of file system anti-virus scan:

- **true** – append new information to the existing file.
- **false** – create a new log file each time the application starts.

Mandatory parameter.

The default value is **true**.

**ReportFileName** – filename of the report file into which results of the component operation will be logged.

The value is empty by default.

**ReportLevel=0|1|2|3|4|9** – report detail level (**0** – Fatal, **1** – Error, **2** – Warning, **3** – Info, **4** – Activity, **9** – Debug).

Mandatory parameter.

The default value is **4**.

**ShowOK=true|false** – mode for logging messages about clean files into the report. In order to disable this mode assign value **false** to this setting.

Mandatory parameter.

The default value is **true**.

**ShowContainerResultOnly=true|false** – the mode of displaying the results of the archive scan. In order to display short report assign value **true** to this setting. By default use expanded format of messages.

Mandatory parameter.

The default value is **false**.

**ShowObjectResultOnly=true|false** – the mode of displaying the results of the scan of a simple object. In order to display short format assign value **yes** to this setting. By default use expanded format of messages.

Mandatory parameter.

The default value is **false**.

## A.1.19. Section [*scanner.container*]

The [**scanner.container**] section includes settings that determine actions to be performed on archives during the anti-virus protection of the server's file systems.

**OnInfected=action** – actions to be performed if an infected object is detected. If the disinfection mode for infected files is enabled, then the specified action will be performed with objects that could not be disinfected.

The value is empty by default.

**OnSuspicion=action** – actions to be performed if the application detects a suspicious object resembling a threat yet unknown to Kaspersky Lab.

The value is empty by default.

**OnWarning=action** – actions to be performed if the application detects a file resembling a known threat.

The value is empty by default.

**OnCured=action** – actions to be performed if the application detects an infected file and disinfects it successfully.

The value is empty by default.

**OnProtected=action** – actions to be performed if the application finds a password-protected object. Such objects cannot be scanned.

The value is empty by default.

**OnCorrupted=action** – actions to be performed if the application detects a corrupted file.

The value is empty by default.

**OnError=action** – actions to be performed if a system error occurs during object scan.

The value is empty by default.

Syntax of the **action** parameter consists of two parts: the action and its additional parameter delimited by a space. Additional parameter value must be specified in quotes.

E.g.:

```
OnInfected=move "/tmp/infected"
```

The action can take one of the following values:

- *move <directory>* – move file to the <directory>.

- *movePath* <directory> – move file to the <directory> recursively (using absolute path).
- *remove* – delete file.
- *exec* <parameter> – run an external command defined by the <parameter> variable.

The following macros can be used as additional parameter of the **exec** action over containers:

- %VIRUSNAME% – name of detected threat or error.
- %LIST% – name of file or a list of infected, suspicious or corrupted files found in a container. The record has the following format:<virus name>lt<file name>.
- %FULLPATH% – full path to container.
- %FILENAME% – file name without path.
- %CONTAINERTYPE% – container type as a string.

## A.1.20. Section [*scanner.object*]

The [**scanner.object**] section contains settings that define actions to be performed on simple objects of certain types during the anti-virus protection of computer file system.

**OnInfected=action** – actions to be performed if an infected object is detected. If the disinfection mode for infected files is enabled, then the specified action will be performed with objects that could not be disinfected.

The value is empty by default.

**OnSuspicion=action** – actions to be performed if the application detects a suspicious object resembling a threat yet unknown to Kaspersky Lab.

The value is empty by default.

**OnWarning=action** – actions to be performed if the application detects a file resembling a known threat.

The value is empty by default.

**OnCured=action** – actions to be performed if the application detects an infected file and disinfects it successfully.

The value is empty by default.

**OnProtected=action** – actions to be performed if the application finds a password-protected object. Such objects cannot be scanned.

The value is empty by default.

**OnCorrupted=action** – actions to be performed if the application detects a corrupted file.

The value is empty by default.

**OnError=action** – actions to be performed if a system error occurs during object scan.

The value is empty by default.

Syntax of the **action** parameter is identical to that for the **[scanner.container]** section (see section A.1.19 on page 125).

The following macros can be used as additional parameter of the **exec** action over containers:

- %VIRUSNAME% – name of detected threat or error.
- %LIST% – name of an infected, suspicious or corrupted file. The record has the following format:<virus name>\t<file name>.
- %FULLPATH% – full path to file.
- %FILENAME% – file name without path.

## A.1.21. Section **[scanner.path]**

Section **[scanner.path]** contains parameters that determine paths to files without which the *kav4lms-kavscanner* component will not function.

**BackupPath= path** – full path to the backup storage directory for backup copies of objects being scanned by the component.

The value is empty by default.

## A.2. Group configuration file

This appendix provides detailed explanation of every section of the *default.conf* configuration file, which defines the **Default** group of settings used to process messages.

Parameters specified for the **Default** group are used in cases, when:

- no groups have been created;
- neither message sender nor recipient are found in any of the existing groups;

- parameter value in a group is not defined.

### Warning!

If a group configuration file is created on the basis of the default.conf configuration file for the **Default** group, remember to change the group name entered in the titles of the configuration file sections.

## A.2.1. Section

### ***[kav4lms:groups.<group\_name>.definition]***

The **[kav4lms:groups.<group\_name>.definition]** section contains group identification parameters:

**Priority** – group priority; if the message belongs to several groups according to its sender (recipient), it will be processed using the group rules with the highest priority. Any natural number can be specified as parameter value. Groups with the same priority and **0** priority are not allowed.

Mandatory parameter.

Parameter value for the **Default** group is **0**.

**Senders** – list of e-mail sender addresses. Each address must be specified in a separate line. Masks and regular expressions are supported. If this option is not defined, the value is assumed to be **\*@\*** (all addresses).

E.g.:

```
Senders=user1@mycompany.com
Senders=reporter*@mycompany.com
Senders=re:office@.*\example\.com
```

Parameter value for the **Default** group is not defined.

**Recipients** – list of e-mail recipient addresses. Each address must be specified in a separate line. Masks and regular expressions are supported. If this option is not defined, the value is assumed to be **\*@\*** (all addresses).

E.g.:

```
Recipients=user2@mycompany.com
Recipients=reporter*@mycompany.com
Recipients=re:office\d+@central\mydomain\.com
```



Parameter value for the **Default** group is not defined.

**Warning!**

At least one of the **Senders** or **Recipients** parameters has to be specified.

## A.2.2. Section

### ***[kav4lms:groups.<group\_name>.settings]***

The **[kav4lms:groups.<group\_name>.settings]** section contains parameters defining message scan policy and addition of special information fields to processed messages.

**Check=anti-virus|content-filter|all|none** – the security service for a group.

Mandatory parameter.

Parameter for the **Default** group is **all**.

**ScanPolicy=message|combined** – e-mail scan policy which defines the way of message analysis.

Mandatory parameter.

Parameter value for the **Default** group is **message**.

**ScanArchives=yes|no** – scan archives. To disable this mode, set the parameter to **no**.

Parameter value for the **Default** group is **yes**.

**ScanPacked=yes|no** – scan packed executables. To disable this mode, set the parameter to **no**.

Parameter value for the **Default** group is **yes**.

**UseAVBasesSet=standard|extended** – the set of anti-virus databases used by the application while scanning. The **extended** set contains, in addition to records contained in the **standard** set, descriptions of riskware, such as adware, remote administration programs, network scanners and virus simulators.

Parameter value for the **Default** group is **standard**.

**UseCodeAnalyzer=yes|no** – scan using a heuristic code analyzer to detect malicious programs, virus modifications, and unknown viruses. To disable this mode, set the parameter to **no**.

Parameter value for the **Default** group is **yes**.

**MaxScanTime** – the maximum time, in seconds, which the application can spend scanning a single object (a message or a message object). If the value is exceeded, the application returns an error.

Parameter value for the **Default** group is **30**.

**Note:**

There can be a situation, when the total scan time of a specific message exceeds the **MaxScanTime** parameter's value, but no error is issued. This occurs when the **combined** type of the scanning policy is selected. Then the total message scanning duration is a sum of the message-as-object scan and part-by-part scan.

**MaxScanDepth=0...UINT\_MAX** – the maximum nesting level of MIME objects allowed in a single message. If the value is exceeded, the application returns an error. The value **0** means unlimited nesting is allowed.

Parameter value for the **Default** group is **10**.

**MIMEEncodingHeuristics=yes|no** – the mode for parsing of MIME objects, which do not comply with RFC standards.

By default, application filter transfers for scanning only RFC-compliant messages. If **MIMEEncodingHeuristics** is set to **yes**, a non-compliant message will be parsed using heuristic algorithms and transferred for scanning in case of its successful decoding. If message decoding fails or if **MIMEEncodingHeuristics** is set to **no**, such messages will not be passed over for scanning.

Parameter value for the **Default** group is **no**.

**Note:**

When enabled, this parameter can slow down scanning.

**AddXHeaders=none|message|parts|all** – instruction to add informational headers containing message scan results (for details please see section 10.4 on page 99).

Mandatory parameter.

Parameter value for the **Default** group is **message**.

**AddDisclaimer=yes|no** – add a disclaimer text to each processed or generated message. You can customize this text by editing *disclaimer* tem-

plate. The disclaimer text is added as a text part at the message end and does not affect or change the content of the original message.

Parameter value for the **Default** group is **no**.

**UsePlaceholderNotice=yes|no** – attach a notification about the deleted object.

Parameter value for the **Default** group is **yes**.

**RejectReply** – header of the notification about the rejected message. The option is not used in case of product integration with gmail.

Parameter value for the **Default** group is:

**Message rejected because it contains malware.**

### A.2.3. Section

#### ***[kav4lms:groups.<group\_name>.actions]***

The **[kav4lms:groups.<group\_name>.actions]** contains options, which determine how e-mail objects are processed after anti-virus scanning:

**InfectedAction=warn|drop|reject|cure|delete|skip** – the default action applied to infected objects.

Mandatory parameter.

Parameter value for the **Default** group is **skip**.

**SuspiciousAction=warn|drop|reject|delete|skip** – the default action applied to objects suspected of being infected with unknown malware.

Mandatory parameter.

Parameter value for the **Default** group is **skip**.

**ProtectedAction=warn|drop|reject|skip|delete** – the action applied to password protected objects that could not be scanned for presence of threats.

Mandatory parameter.

Parameter value for the **Default** group is **skip**.

**ErrorAction=warn|skip|delete** – the action applied to corrupted objects that could not be scanned because of an error.

Mandatory parameter.

Parameter value for the **Default** group is **skip**.

**VirusNameAction=warn|drop|reject** – actions to be applied to a message or its object if it is infected with a virus listed in the **VirusNameList** parameter.

Mandatory parameter.

Parameter value for the **Default** group is **drop**.

**FilteredMimeType=skip|delete|drop|reject|warn** – action to be applied to the attachment of the MIME type defined by the **IncludeMimeType** parameter.

Parameter value for the **Default** group is **skip**.

**FilteredNameAction=skip|delete|drop|reject|rename|warn** – action to be applied to the attachment with the name defined by the **IncludeName** parameter mask.

Parameter value for the **Default** group is **skip**.

**FilteredSizeAction=skip|delete|drop|reject|warn** – action to be applied to the attachment if its size corresponds with the value set by the **IncludeSize** parameter.

Parameter value for the **Default** group is **skip**.

## A.2.4. Section

### ***[kav4lms:groups.<group\_name>.contentfiltering]***

The **[kav4lms:groups.<group\_name>.contentfiltering]** section defines rules for message filtering:

**IncludeMimeType** – defines masks for filtering by MIME type. Objects will be filtered if their MIME types match the specified masks and do not match the masks used to define exclusions from scanning (**ExcludeMimeType** parameter).

You can define several values as a list. Each parameter must be entered in a separate line. Wildcards («\*» and «?») and regular expressions are supported.

E.g.:

```
IncludeMimeType=application/octet-stream
```

```

IncludeMime=application/vnd.*
IncludeMime=re:image/.*
IncludeMime=re:multipart/(encrypted|signed)

```

If parameter value is not specified or if it is empty, filtration by MIME type will not be performed.

For the **Default** group parameter value is empty.

**ExcludeMime** – defines MIME type masks of objects that will be excluded from filtering. Objects will be skipped if their types do match these masks.

If the **ExcludeMime** list is specified and **IncludeMime** is not, the masks not belonging to the **ExcludeMime** list will be filtered.

You can define several values as a list. Each parameter must be entered in a separate line. Wildcards («\*» and «?») and regular expressions are supported.

E.g.:

```

ExcludeMime=application/octet-stream
ExcludeMime=application/vnd.*
ExcludeMime=re:image/.*
ExcludeMime=re:multipart/(encrypted|signed)

```

For the **Default** group parameter value is empty.

**IncludeName** – defines masks for filtering by name. The application will filter the objects if their names match the specified masks and do not match the masks used to define exclusions from scanning (**ExcludeName** parameter).

If parameter value is not specified or if it is empty, filtration by attachment name will not be performed.

You can define several values as a list. Each parameter must be entered in a separate line. Wildcards («\*» and «?») and regular expressions are supported.

E.g.:

```

IncludeName=*accounting*
IncludeName=re:.*\.(doc|xls|ppt)
IncludeName=re:.*\.(pif|com|exe)

```

For the **Default** group parameter value is empty.

**ExcludeName** – defines masks of objects that will be excluded from filtering. The application will skip objects matching these masks.

If the **ExcludeName** is specified and **IncludeName** is not, the masks, not belonging to the **ExcludeName** list are considered to be included into filtering.

You can define several values as a list. Each parameter must be entered in a separate line. Wildcards («\*» and «?») and regular expressions are supported.

E.g.:

```
ExcludeName=re:.*\.(txt|rtf)
ExcludeName=re:.*\.(doc|xls|ppt)
ExcludeName=re:.*\.(pif|com|exe)
```

For the **Default** group parameter value is empty.

**IncludeSize** – size of e-mail attachments to be filtered. You can specify the value in bytes, for example, **3456261**, or use short record format indicating the size magnitude: **10KB**, **100MB**. To filter out empty values, set the parameter to **0**.

Record format:

**IncludeSize=attachment\_size** – the application will filter attachments with size matching the specified value.

**IncludeSize<attachment\_size** - the application will precisely filter attachments with size smaller than the specified value.

**IncludeSize<=attachment\_size** - the application will filter attachments with size smaller than the specified value or equal to it.

**IncludeSize>attachment\_size** - the application will precisely filter attachments with size larger than the specified value.

**IncludeSize>=attachment\_size** - the application will filter attachments with size larger than the specified value or equal to it.

**IncludeSize=0** – the application will filter all empty attachments.

If parameter value is not specified, filtration by attachment type will not be performed.

For the **Default** group parameter value is empty.

**ExcludeSize** – size of e-mail attachments to be excluded from filtering. Record format is identical to that of the **IncludeSize** parameter. To skip empty attachments, set the parameter to **0**.

For the **Default** group parameter value is empty.

**VirusNameList** – list of threats that require special actions defined by **VirusNameAction** to be applied to objects or messages which they infect. Threat name should be specified as it appears in Virus Encyclopedia at [www.viruslist.com](http://www.viruslist.com). Masks and regular expressions are allowed. To specify several values, separate them with commas.

E.g.:

```
VirusNameList=re:trojan.*, backdoor*
```

If parameter value is not defined, objects will be processed in accordance with the status assigned to them during scanning.

For the **Default** group parameter value is empty.

**RenameTo=<file\_name>|.<extension>** – the mode of object renaming when the **rename** action is applied:

- **RenameTo=<file\_name>** – file name will be replaced completely with the specified value.
- **RenameTo=.<extension>** – the specified extension will be appended to file name.

E.g.:

```
RenameTo=.vir
```

The *file.doc* file will be renamed to *file.doc.vir*.

```
RenameTo=VIRUS-DO-NOT-OPEN
```

The *file.doc* file will be renamed to *VIRUS-DO-NOT-OPEN*.

If parameter value is not defined, the application will not rename objects.

Parameter value for the **Default** group is **.vir**.

## A.2.5. Section

### ***[kav4lms:groups.<group\_name>.notifications]***

The **[kav4lms:groups.<group\_name>.notifications]** section contains notification options:

**NotifySender=all|filtered|infected|protected|suspicious|error|none** – notify the original mail senders upon detection of e-mail messages (or message objects) with this status.

You can define several values as a list. Each parameter must be entered in a separate line. If an empty value is specified, no notifications will be delivered to message senders.

Mandatory parameter.

Parameter value for the **Default** group is **none**.

#### **Note:**

To make the application send notifications upon detection of objects with various statuses, you can set several values for **NotifySender** parameter, e.g.:

```
NotifySender=filtered
NotifySender=infected
```

You can assign the **NotifyRecipients** and **NotifyAdmin** parameters in the same manner.

**NotifyRecipients=all|filtered|infected|protected|suspicious|error|none** – notify the original mail recipients upon detection of e-mail messages (or message objects) with this status.

You can define several values as a list. Each parameter must be entered in a separate line. If an empty value is specified, no notifications will be delivered to original message recipients.

Mandatory parameter.

Parameter value for the **Default** group is **all**.

**NotifyAdmin=all|filtered|infected|protected|suspicious|error|none** – notify the administrator upon detection of e-mail messages or message objects with this status.



You can define several values as a list. Each parameter must be entered in a separate line. If an empty value is specified, no notifications will be delivered to the administrator.

Mandatory parameter.

Parameter value for the **Default** group is **none**.

**AdminAddresses** – e-mail address of the mail server administrator. You can specify several addresses, separated with a comma.

Parameter value for the **Default** group is **postmaster**.

**Note:**

The **AdminAddresses** parameter refers to the security administrator, not the administrator of Kaspersky Anti-Virus (referred to by the **ProductAdmins** parameter in the **[kav4lms:server.notifications]** section of the *kav4lms.conf* file.

**PostmasterAddresses** – e-mail address substituted as the sender's address ('FROM' field) for issued notifications.

Parameter value for the **Default** group is **POSTMASTER@localhost**.

**Templates** – directory for storing notification templates.

Parameter value for the **Default** group is  
**/etc/opt/kaspersky/kav4lms/templates/en** (in Linux)  
**/usr/local/etc/kaspersky/kav4lms/templates/en** (in FreeBSD).

**Subject** – header of the standard notification added to the **Subject** field.

Parameter value for the **Default** group is **Anti-virus notification message**.

**Charset** – character set to be used in notifications.

Parameter value for the **Default** group is **us-ascii**.

**TransferEncoding** – value of the notification encoding algorithm.

Parameter value for the **Default** group is **7bit**.

**UseCustomTemplates=yes|no** – enable using custom templates for generating notifications. To enable this mode, set the parameter to **yes**.

Parameter value for the **Default** group is **no**.

**SenderSubject** – mail subject header of the sender notification.

Parameter value for the **Default** group is **Anti-virus notification message**.

**AdminSubject** – mail subject header of the security administrator notification.

Parameter value for the **Default** group is **Anti-virus notification message**.

## A.2.6. Section

### ***[kav4lms:groups.<group\_name>.backup]***

The **[kav4lms:groups.<group\_name>.backup]** section contains options for creating backup copies before applying any actions to e-mail messages:

**Policy=message|info|none** – defines backup policy.

Parameter value for the **Default** group is **info**.

**Options=cured|deleted|dropped|rejected|warning|renamed|all** – type of messages for which backups must be created.

You can specify several comma-delimited values.

Mandatory parameter.

Parameter value for the **Default** group is **all**.

**Destination=/var/opt/kaspersky/kav4lms/backup/** – directory for storing backup copies of messages.

Parameter value for the **Default** group is  
**/var/opt/kaspersky/kav4lms/backup/** (in Linux)  
**/var/db/kaspersky/kav4lms/backup/** (in FreeBSD).

## A.3. Command line parameters for component *kav4lms-licensemanager*

Help options:

<b>-h</b>	Display help information about the <i>kav4lms-licensemanager</i> component to the screen;
-----------	---

<b>-v</b>	Display the application version.
Key management options:	
<b>-s</b>	Display information about all installed keys to the screen.
<b>-c (-C) &lt;path_to_file&gt;</b>	Use alternative configuration file <path_to_file>.
<b>-k &lt;path_to_file&gt;</b>	Display information about key <path_to_key_file> on the screen.
<b>-a &lt;path_to_file&gt;</b>	Install the key <path_to_key_file>.
<b>-d(a r)</b>	Remove active (-da option) key or additional (-dr option) key.
<b>-i</b>	Output detailed information about licensed objects to console.

## A.4. Return codes of the *kav4lms-licensemanager* component

During its operation the *kav4lms-licensemanager* component may return the following codes:

<b>0</b>	The component successfully loaded information about the key and successfully completed its operation.
<b>30</b>	System error occurred during the component's operation.
<b>64</b>	Key information is missing, or no keys were found at the path specified in the configuration file.
<b>65</b>	Unable to load configuration file.
<b>66</b>	Invalid configuration file option.
<b>70</b>	Component <i>kav4lms-licensemanager</i> is corrupted.

## A.5. Command line parameters for component *kav4lms-keepup2date*

Help options:	
<b>-v</b>	Print to the screen the version of the application and close the component.
<b>-h</b>	Print to the screen help information about the command line parameters supported by the component, and close the component.
Operation options:	
<b>-r</b>	Rollback the last update to the previous version.
<b>-s</b>	Print the list of the updates servers to the screen.
<b>-k</b>	Do not execute <b>PostUpdateCmd</b> command after the anti-virus database update has been successfully completed.
<b>-q</b>	The mode of the component operation during which no system messages will be printed to the screen.
<b>-e</b>	The mode of the component operation during which only messages about critical errors will be printed to the screen.
<b>-x &lt;path_to_file&gt;</b>	Copy all updates of the anti-virus database into local directory <b>&lt;path_to_file&gt;</b> .
<b>-g &lt;URL&gt;</b>	Address for updating the anti-virus database. When this modifier is specified, the update will be performed from this address.
<b>-d &lt;path_to_file&gt;</b>	Use pid-file of the component, located in local directory <b>&lt;path_to_file&gt;</b> .

Report generation options:	
<b>-l &lt;path_to_file&gt;</b>	Log the results of the component's operation in file <path_to_file>.

## A.6. Return codes of the *kav4lms-keepup2date* component

During its operation the *kav4lms-keepup2date* component may return the following codes:

<b>0</b>	The anti-virus database does not need to be updated.
<b>1</b>	The anti-virus database has been updated successfully.
<b>10</b>	Critical error occurred, the updating process will be terminated.
<b>11</b>	Error occurred – another application instance is running.
<b>12</b>	Error occurred during the rollback to the last update of the anti-virus database.
<b>30</b>	Could not run command <b>PostUpdateCmd</b> after the anti-virus database update.
<b>60</b>	Key information is missing or no keys were found at the path specified in the configuration file.
<b>75</b>	Unable to load the configuration file or settings error.

## **APPENDIX B. KASPERSKY LAB**

Founded in 1997, Kaspersky Lab has become a recognized leader in information security technologies. It produces a wide range of data security software and delivers high-performance, comprehensive solutions to protect computers and networks against all types of malicious programs, unsolicited and unwanted e-mail messages, and hacker attacks.

Kaspersky Lab is an international company. Headquartered in the Russian Federation, the company has representative offices in the United Kingdom, France, Germany, Japan, USA (CA), the Benelux countries, China, Poland, and Romania. A new company department, the European Anti-Virus Research Centre, has recently been established in France. Kaspersky Lab's partner network incorporates more than 500 companies worldwide.

Today, Kaspersky Lab employs more than 450 specialists, each of whom is proficient in anti-virus technologies, with 10 of them holding M.B.A. degrees, 16 holding Ph.Ds, and senior experts holding membership in the Computer Anti-Virus Researchers Organization (CARO).

Kaspersky Lab offers best-of-breed security solutions, based on its unique experience and knowledge, gained in over 14 years of fighting computer viruses. A thorough analysis of computer virus activities enables the company to deliver comprehensive protection from current and future threats. Resistance to future attacks is the basic policy implemented in all Kaspersky Lab's products. The company's products consistently remain at least one step ahead of many other vendors in delivering extensive anti-virus coverage for home users and corporate customers alike.

Years of hard work have made the company one of the top security software manufacturers. Kaspersky Lab was one of the first businesses of its kind to develop the highest standards for anti-virus defense. The company's flagship product, Kaspersky Anti-Virus, provides full-scale protection for all tiers of a network, including workstations, file servers, mail systems, firewalls, Internet gateways, and hand-held computers. Its convenient and easy-to-use management tools ensure advanced automation for rapid virus protection across an enterprise. Many well-known manufacturers use the Kaspersky Anti-Virus kernel, including Nokia ICG (USA), F-Secure (Finland), Aladdin (Israel), Sybari (USA), G Data (Germany), Deerfield (USA), Alt-N (USA), Microworld (India) and BorderWare (Canada).

Kaspersky Lab's customers benefit from a wide range of additional services that ensure both stable operation of the company's products, and compliance with specific business requirements. Our databases are updated every hour. The company provides its customers with a 24-hour technical support service, which is available in several languages to accommodate its international clientele.

## B.1. Other Kaspersky Lab Products

### Kaspersky Lab News Agent

The News Agent is intended for timely delivery of news published by Kaspersky Lab, notifications about the current status of virus activity, and fresh news. The program reads the list of available news feeds and their content from the Kaspersky Lab news server at specified intervals.

News Agent enables users to;

- See the current virus forecast .in the system tray.
- Subscribe to and unsubscribe from news feeds.
- Retrieve news from each selected feed at the specified interval and receive notifications about fresh news.
- Review news on the selected feeds.
- Review the list of feeds and their status.
- Open full article text in your browser.

News Agent is a stand-alone Microsoft Windows application that can be used independently or may be bundled with various integrated solutions offered by Kaspersky Lab Ltd.

### Kaspersky® OnLine Scanner

This program is a free service provided to the visitors of Kaspersky Lab's corporate website. The service delivers an efficient online anti-virus scan of your computer. Kaspersky OnLine Scanner runs directly from your browser. This way, users receive quick responses to questions regarding potential infections on their computers. Using the service, visitors can:

- Exclude archives and e-mail databases from scanning.
- Select standard/extended databases for scanning.
- Save a report on the scanning results in .txt or .html formats.

### Kaspersky® OnLine Scanner Pro

The program is a subscription service available to the visitors of Kaspersky Lab's corporate website. The service delivers an efficient online anti-virus scan of your computer and disinfects dangerous files. Kaspersky OnLine Scanner Pro runs directly from your browser. Using the service, visitors can:

- Exclude archives and e-mail databases from scanning.
- Select standard/extended databases for scanning.
- Save a report on the scanning results in .txt or .html formats.

## Kaspersky Anti-Virus® 7.0

Kaspersky Anti-Virus 7.0 is designed to safeguard personal computers against malicious software as an optimal combination of conventional methods of anti-virus protection and new proactive technologies.

The program provides for complex anti-virus checks, including:

- Anti-virus scanning of e-mail traffic on the level of data transmission protocol (POP3, IMAP and NNTP for incoming mail and SMTP for outgoing messages), regardless of the mail client being used, as well as disinfection of e-mail databases.
- Real-time anti-virus scanning of Internet traffic transferred via HTTP.
- Anti-virus scanning of individual files, folders, or drives. In addition, a preset scan task can be used to initiate anti-virus analysis exclusively for critical areas of the operating system and start-up objects of Microsoft Windows.

Proactive protection offers the following features:

- **Controls modifications within the file system.** The program allows users to create a list of applications, which it will control on a per component basis. It helps protect application integrity against the influence of malicious software.
- **Monitors processes in random-access memory.** Kaspersky Anti-Virus 7.0 in a timely manner notifies users whenever it detects dangerous, suspicious or hidden processes or in case when unauthorized changes in active processes occur.
- **Monitors changes in OS registry** due to internal system registry control.
- **Hidden Processes Monitor** helps protect from malicious code concealed in the operating system using rootkit technologies.
- **Heuristic Analyzer.** When scanning a program, the analyzer emulates its execution and logs all suspicious activity, such as, opening or writing to a file, interrupt vector intercepts, etc. A decision is made based on this procedure regarding possible infection of the program with a virus. Emulation occurs in an isolated virtual environment which reliably protects the computer of infection.
- **Performs system restore** after malware attacks by logging all changes to the registry and computer file system and rolls them back at user's discretion.

## Kaspersky® Internet Security 7.0

Kaspersky Internet Security 7.0 is an integrated solution for protection of personal computers against the major information- threats (viruses, hackers, spam



and spyware). A single interface enables users to configure and manage all the program's components.

The anti-virus protection features include:

- **Anti-virus scanning of e-mail traffic** on the level of data transmission protocol (POP3, IMAP and NNTP for incoming mail and SMTP for outgoing messages), regardless of the mail client being used. The program includes plug-ins for popular e-mail clients (such as Microsoft Office Outlook, Microsoft Outlook Express/Windows Mail, and The Bat!) and supports disinfection of their e-mail databases.
- **Real-time anti-virus scanning of Internet traffic** transferred via HTTP.
- **File system protection:** anti-virus scanning of individual files, folders or drives. In addition, the application can perform anti-virus analysis exclusively for critical areas of the operating system and Microsoft Windows start-up objects.
- **Proactive protection:** the program constantly monitors application activity and processes running in random-access memory, preventing dangerous changes to the file system and registry, and restores the system after malicious influence.

**Protection against Internet-fraud** is ensured by recognition of phishing attacks, thereby preventing confidential data leaks (above all passwords, bank account and credit card numbers) and blocking execution of dangerous scripts on web pages, pop-up windows and advertisement banners. The **autodialer blocking** feature helps identify software that attempts to use your modem for hidden unauthorized connections to paid phone services and blocks such activity. *Privacy Control* module keeps your confidential information secure from unauthorized access and transmission. *Parental Control* is a Kaspersky Internet Security component that monitors user access to the Internet.

Kaspersky Internet Security 7.0 **registers attempts to scan the ports of your computer**, which frequently precede network attacks, and successfully defends against typical network attacks. The program uses **defined rules as a basis** for control over all network transactions tracking all **incoming and outgoing data packets**. **Stealth Mode** (owing to the SmartStealth™ technology) **prevents computer detection from outside**. When you switch to Stealth Mode, the system blocks all network activity except for a few transactions allowed in user-defined rules.

The program employs an all-inclusive approach to anti-spam filtering of incoming e-mail messages:

- Verification against black and white lists of recipients (including addresses of phishing sites).
- Inspection of phrases in message body.

- Analysis of message text using a learning algorithm.
- Recognition of spam sent in image files.

### Kaspersky Anti-Virus Mobile

Kaspersky® Anti-Virus Mobile provides antivirus protection for mobile devices running Symbian OS and Microsoft Windows Mobile. The program provides comprehensive virus scanning, including:

- **On-demand scans** of the mobile device's onboard memory, memory cards, an individual folder, or a specific file; if an infected file is detected, it is moved to Quarantine or deleted.
- **Real-time scanning** – all incoming and outgoing files are automatically scanned, as well as files when attempts are made to access them.
- **Protection from text message spam.**

### Kaspersky Anti-Virus for File Servers

This software package provides reliable protection for file systems on servers running Microsoft Windows, Novell NetWare, Linux and Samba from all types of malware. The suite includes the following Kaspersky Lab applications:

- [Kaspersky Administration Kit.](#)
- [Kaspersky Anti-Virus for Windows Server.](#)
- [Kaspersky Anti-Virus for Linux File Server.](#)
- [Kaspersky Anti-Virus for Novell Netware.](#)
- [Kaspersky Anti-Virus for Samba Server.](#)

Features and functionality:

- *Protects server file systems in real time:* All server files are scanned when opened or saved on the server;
- *Prevents virus outbreaks;*
- *On-demand scans* of the entire file system or individual files and folders;
- *Use of optimization technologies* when scanning objects in the server file system;
- *System rollback after virus attacks;*
- *Scalability of the software package* within the scope of system resources available;
- *Monitoring of the system load balance;*

- *Creating a list of trusted processes* whose activity on the server is not subject to control by the software package;
- *Remote administration* of the software package, including centralized installation, configuration, and administration;
- *Saving backup copies of infected and deleted objects* in case you need to restore them;
- *Quarantining suspicious objects*;
- *Send notifications on events* in program operation to the system administrator;
- *Log detailed reports*;
- *Automatically update* program databases.

### **Kaspersky Open Space Security**

Kaspersky Open Space Security is a software package with a new approach to security for today's corporate networks of any size, providing centralized protection information systems and support for remote offices and mobile users.

The suite includes four programs:

- Kaspersky Work Space Security
- Kaspersky Business Space Security
- Kaspersky Enterprise Space Security
- Kaspersky Total Space Security

Specifics on each program are given below.

**Kaspersky WorkSpace Security** is a program for centralized protection of workstations inside and outside of corporate networks from all of today's Internet threats (viruses, spyware, hacker attacks, and spam).

Features and functionality:

- *Comprehensive protection from viruses, spyware, hacker attacks, and spam*;
- *Proactive Defense* from new malicious programs whose signatures are not yet added to the database;
- *Personal Firewall* with intrusion detection system and network attack warnings;
- *Rollback for malicious system modifications*;
- *Protection from phishing attacks and junk mail*;

- *Dynamic resource redistribution* during complete system scans;
- *Remote administration* of the software package, including centralized installation, configuration, and administration;
- *Support for Cisco® NAC* (Network Admission Control);
- *Scanning of e-mail and Internet traffic* in real time;
- *Blocking of popup windows and banner ads* when on the Internet;
- *Secure operation in any type of network*, including Wi-Fi;
- *Rescue disk creation tools* that enable you to restore your system after a virus outbreak;
- *An extensive reporting system* on protection status;
- *Automatic database updates*;
- *Full support for 64-bit operating systems*;
- *Optimization of program performance on laptops* (Intel® Centrino® Duo technology);
- *Remote disinfection capability* (Intel® Active Management, Intel® vPro™).

**Kaspersky Business Space Security** provides optimal protection of your company's information resources from today's Internet threats. Kaspersky Business Space Security protects workstations and file servers from all types of viruses, Trojans, and worms, prevents virus outbreaks, and secures information while providing instant access to network resources for users.

Features and functionality:

- *Remote administration* of the software package, including centralized installation, configuration, and administration;
- *Support for Cisco® NAC* (Network Admission Control);
- *Protection of workstations and file servers from all types of Internet threats*;
- *iSwift technology to avoid rescanning files within the network*;
- *Distribution of load among server processors*;
- *Quarantining suspicious objects* from workstations;
- *Rollback for malicious system modifications*;
- *scalability of the software package within the scope of system resources* available;

- *Proactive Defense* for workstations from new malicious programs whose signatures are not yet added to the database;
- *Scanning of e-mail and Internet traffic* in real time;
- *Personal Firewall* with intrusion detection system and network attack warnings;
- *Protection while using Wi-Fi networks*;
- *Self-Defense from malicious programs*;
- *Quarantining* suspicious objects;
- *Automatic database updates*.

### **Kaspersky Enterprise Space Security**

This program includes components for protecting linked workstations and servers from all today's Internet threats. It deletes viruses from e-mail, keeping information safe while providing secure access to network resources for users.

Features and functionality:

- *Protection of workstations and file servers from viruses, Trojans, and worms*;
- *Protection of Sendmail, Qmail, Postfix and Exim mail servers*;
- *Scanning of all e-mails on Microsoft Exchange Server, including shared folders*;
- *Processing of e-mails, databases, and other objects for Lotus Domino servers*;
- *Protection from phishing attacks and junk mail*;
- *Preventing mass mailings and virus outbreaks*;
- Scalability of the software package within the scope of system resources available;
- Remote administration of the software package, including centralized installation, configuration, and administration;
- Support for Cisco® NAC (Network Admission Control);
- Proactive Defense for workstations from new malicious programs whose signatures are not yet added to the database;
- Personal Firewall with intrusion detection system and network attack warnings;

- *Secure operation while using Wi-Fi networks;*
- *Scans Internet traffic* in real time;
- *Rollback for malicious system modifications;*
- *Dynamic resource redistribution* during complete system scans;
- Quarantining suspicious objects;
- *An extensive reporting system* on protection system status;
- *Automatic database updates.*

### **Kaspersky Total Space Security**

This solution monitors all inbound and outbound data streams (e-mail, Internet, and all network interactions). It includes components for protecting workstations and mobile devices, keeps information safe while providing secure access for users to the company's information resources and the Internet, and ensures secure e-mail communications.

Features and functionality:

- *Comprehensive protection from viruses, spyware, hacker attacks, and spam* on all levels of the corporate network, from workstations to Internet gateways;
- Proactive Defense for workstations from new malicious programs whose signatures are not yet added to the database;
- *Protection of mail servers and linked servers;*
- *Scans Internet traffic* (HTTP/FTP) entering the local area network in real time;
- Scalability of the software package within the scope of system resources available;
- *Blocking access from infected workstations;*
- *Prevents virus outbreaks;*
- *Centralized reporting on protection status;*
- Remote administration of the software package, including centralized installation, configuration, and administration;
- *Support for Cisco® NAC* (Network Admission Control);
- *Support for hardware proxy servers;*
- *Filters Internet traffic* using a trusted server list, object types, and user groups;

- *iSwift technology to avoid rescanning files within the network;*
- *Dynamic resource redistribution during complete system scans;*
- *Personal Firewall with intrusion detection system and network attack warnings;*
- *Secure operation for users on any type of network, including Wi-Fi;*
- *Protection from phishing attacks and junk mail;*
- *Remote disinfection capability (Intel® Active Management, Intel® vPro™);*
- *Rollback for malicious system modifications;*
- *Self-Defense from malicious programs;*
- *Full support for 64-bit operating systems;*
- *Automatic database updates.*

### **Kaspersky Security for Mail Servers**

This program is for protecting mail servers and linked servers from malicious programs and spam. The program includes application for protecting all standard mail servers (Microsoft Exchange, Lotus Notes/Domino, Sendmail, Qmail, Postfix and Exim) and also enables you to configure a dedicated e-mail gateway. The solution includes:

- [Kaspersky Administration Kit.](#)
- [Kaspersky Mail Gateway.](#)
- [Kaspersky Anti-Virus for Lotus Notes/Domino.](#)
- [Kaspersky Anti-Virus for Microsoft Exchange.](#)
- [Kaspersky Anti-Virus for Linux Mail Server.](#)

Its features include:

- *Reliable protection from malicious or potentially dangerous programs;*
- *Junk mail filtering;*
- *Scans incoming and outgoing e-mails and attachments;*
- *Scans all e-mails on Microsoft Exchange Server for viruses, including shared folders;*
- *Processes e-mails, databases, and other objects for Lotus Notes/Domino servers;*
- *Filters e-mails by attachment type;*

- *Quarantines suspicious objects;*
- *Easy-to-use administration system for the program;*
- *Prevents virus outbreaks;*
- *Monitors protection system status using notifications;*
- *Reporting system for program operation;*
- Scalability of the software package within the scope of system resources available;
- *Automatic database updates.*

### **Kaspersky Security for Internet Gateways**

This program provides secure access to the Internet for all an organization's employees, automatically deleting malware and riskware from the data incoming on HTTP/FTP. The solution includes:

- [Kaspersky Administration Kit.](#)
- [Kaspersky Anti-Virus for Proxy Server.](#)
- [Kaspersky Anti-Virus for Microsoft ISA Server.](#)
- [Kaspersky Anti-Virus for Check Point FireWall-1.](#)

Its features include:

- *Reliable protection from malicious or potentially dangerous programs;*
- *Scans Internet traffic (HTTP/FTP) in real time;*
- *Filters Internet traffic using a trusted server list, object types, and user groups;*
- *Quarantines suspicious objects;*
- *Easy-to-use administration system;*
- *Reporting system for program operation;*
- *Support for hardware proxy servers;*
- Scalability of the software package within the scope of system resources available;
- *Automatic database updates.*

### **Kaspersky® Anti-Spam**

Kaspersky® Anti-Spam is a cutting-edge software suite designed to help organizations with small- and medium-sized networks wage war against the onslaught



of unsolicited e-mail messages (spam). The product combines the revolutionary technology of linguistic analysis with modern methods of e-mail filtration, including DNS Black Lists and formal letter features. Its unique combination of services allows users to identify and wipe out up to 95% of unwanted traffic.

Installed at the entrance to a network, where it monitors incoming e-mail traffic streams for spam, Kaspersky® Anti-Spam acts as a barrier to unsolicited e-mail. The product is compatible with any mail system and can be installed on either an existing mail server or a dedicated one.

Kaspersky® Anti-Spam's high performance is ensured by daily updates to the content filtration database, adding samples provided by the Company's linguistic laboratory specialists. Databases are updated every 20 minutes.

### **Kaspersky Anti-Virus® for MIMESweeper**

Kaspersky Anti-Virus® for MIMESweeper provides high-speed scanning of traffic on servers running Clearswift MIMESweeper for SMTP / Clearswift MIMESweeper for Exchange / Clearswift MIMESweeper for Web.

The program is a plug-in and scans for viruses and processes inbound and outbound e-mail traffic in real time.

## **B.2. Contact Us**

If you have any questions, comments, or suggestions, please refer them to one of our distributors or directly to Kaspersky Lab. We will be glad to assist you in any matters related to our product by phone or via e-mail. Rest assured that all of your recommendations and suggestions will be thoroughly reviewed and considered.

Technical support	Please find the technical support information at <a href="http://www.kaspersky.com/supportinter.html">http://www.kaspersky.com/supportinter.html</a> Helpdesk: <a href="http://www.kaspersky.com/helpdesk.html">www.kaspersky.com/helpdesk.html</a>
General information	WWW: <a href="http://www.kaspersky.com">http://www.kaspersky.com</a> <a href="http://www.viruslist.com">http://www.viruslist.com</a> E-mail: <a href="mailto:info@kaspersky.com">info@kaspersky.com</a>

# APPENDIX C. THIRD-PARTY SOFTWARE

This section contains a list of third-party software used in development of Kaspersky Anti-Virus 5.6 for Linux Mail Server and the terms of its use.

## C.1. *Pcre* library

The following terms regulate *Pcre* library use:

PCRE is a library of functions to support regular expressions whose syntax and semantics are as close as possible to those of the Perl 5 language.

Release 5 of PCRE is distributed under the terms of the "BSD" licence, as specified below. The documentation for PCRE, supplied in the "doc" directory, is distributed under the same terms as the software itself.

Written by: Philip Hazel <ph10@cam.ac.uk>

University of Cambridge Computing Service,

Cambridge, England. Phone: +44 1223 334714.

Copyright (c) 1997-2004 University of Cambridge

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University of Cambridge nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## **C.2. *Expat* library**

**The following terms regulate Expat library use:**

Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd and Clark Cooper.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## **C.3. *AgentX++v1.4.16* library**

**The following terms regulate AgentX++v1.4.16 library use:**

AGENTX++ LICENSE AGREEMENT

=====

THIS LICENSE AGREEMENT (this "Agreement") is made effective as of the date the product is installed by and between (i) Frank Fock, the author of

AgentX++ ("LICENSOR") and the party executing this Agreement as Licensee ("LICENSEE").

## 1. DEFINITIONS.

1.1 The term "Software Product" means Frank Fock's AgentX++ computer software (including Source Code, derived Object Code, and derived Executable Code as defined in Section 1.3, 1.4, and 1.5) and documentation thereof, as specified in Exhibit A, that is provided by LICENSOR to LICENSEE hereunder, including bug fixes and updates thereto provided by LICENSOR to LICENSEE in connection with this Agreement. The term "derived" in the above context refers to the process of creating machine executable code from the original Source Code only. It does not refer to amendment or alteration of the original Source Code by LICENSOR or any third party.

1.2 The term "Intellectual Property Rights" means patent rights, copyright rights, trade secret rights, and any other intellectual property rights.

1.3 The term "Executable Code" is a fully compiled and linked program that contains any code derived from the Software Product. It can no longer be altered or combined with any other code. Executable code is ready to be executed by a computer and is essentially a complete software image for use in a specific product.

1.4 The term "Object Code" is any compiled version of the Software Product that can be linked and therefore combined with other code to create Executable Code. Examples of Object Code are libraries and software development kits, in particular SNMP agent development kits.

1.5 The term "Source Code" is the human readable form of the Software Product, as specified in Exhibit A.

1.6 Documentation means the documentation regarding the Licensed Software provided by LICENSOR to LICENSEE hereunder.

1.7 The term "Site" is a specific address belonging to a single business unit operating at that address.

## 2. GRANT OF LICENSE.

2.1 Source Code Site License. Subject to the terms and conditions of this Agreement, and upon payment by LICENSEE to LICENSOR of the one-time license fee set forth in Addendum A, LICENSOR grants LICENSEE a perpetual (subject to termination rights in Section 6), non-exclusive, non-transferable license to reproduce, use, modify, or have modified by a third party contractor (modifications in accordance to Section 2.6) subject to a confidentiality agreement no less restrictive than this Agreement, the Source Code for internal use only, for the sole purpose of developing AgentX-enabled SNMP agents at the Site (hereafter "Licensed Site") specified by LICENSEE during license purchase. Additionally, Customer's contractors and employees reporting directly and only to

a manager at the Licensed Site, such as telecommuters, may use the Software Product at remote locations. Off-site employees re-reporting in any way to a manager at their location are not covered under this Site License.

2.2 Except as specified in 2.1, neither the Software Product Source Code nor Object Code derived from the Software Product may be redistributed or resold. Executable Code programs derived from the Software Product may be redistributed and resold without limitation and without royalty, provided that LICENSEE added significant functionality to those derived Executable Code programs. Functionality in this context refers to the program's behavior, not appearance.

2.3 No Sublicense Right. LICENSEE has no right to transfer, or sublicense the Licensed Software to any third party, except as specified in 2.2 and except if the third party takes over the business of LICENSEE.

2.4 Other Restrictions in License Grants. LICENSEE may not: (i) copy the Licensed Software, except as necessary to use the Licensed Software in accordance with the license granted under Section 2.1 and 2.2, and except for a reasonable number of backup copies.

2.5 No Trademark License. LICENSEE has no right or license to use any trademark of LICENSOR during or after the term of this Agreement.

2.6 Proprietary Notices. The Licensed Software is copyrighted. All proprietary notices incorporated in, marked on, or affixed to the Licensed Software by LICENSOR shall be duplicated by LICENSEE on all copies, in whole or in part, in any form of the Licensed Software and not be altered, removed, or obliterated on such copies.

2.7 Reservation. LICENSOR reserve all rights and licenses to the Licensed Software not expressly granted to LICENSEE under this Agreement.

2.8 Delivery. Upon execution of this Agreement, and payment of the amounts due and owing under this Agreement, LICENSOR will provide LICENSEE with one (1) copy of the Software Product by downloading from LICENSOR's Web site.

### 3. PRODUCT WARRANTY.

3.1. LICENSOR warrants to LICENSEE that, at the date of delivery of the Software Product to LICENSEE and for a period ending 90 days following the date of

delivery of the Software Product to LICENSEE the Software Product shall perform substantially in accordance with the published specifications and Documentation. If notified in writing by LICENSEE, LICENSOR may, at its option, correct significant program errors in the Software Product within a reasonable time period. THE FOREGOING PRODUCT WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A

PARTICULAR PURPOSE, WHETHER IMPOSED BY CONTRACT, STATUTE, COURSE OF DEALING, CUSTOM OR USAGE OR OTHERWISE.

3.2. In no event shall LICENSOR be liable to LICENSEE, in excess of the price paid to LICENSOR by LICENSEE for the Software Product hereunder, for any breach of warranty or any claim, loss or damage arising from or relating to the installation, use or performance of the Software Product (including, without limitation, any indirect, special, incidental or consequential damages).

3.3. LICENSOR reserves the right at any time to make changes to the Software Product.

3.4. IN NO EVENT SHALL LICENSOR BE LIABLE (WHETHER IN TORT, NEGLIGENCE, CONTRACT, WARRANTY, PRODUCT LIABILITY OR OTHERWISE) FOR ANY INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES OR LOSS OF PROFITS OR SAVINGS ARISING OUT OF ITS PERFORMANCE OR NONPERFORMANCE OF TERMS OF THIS AGREEMENT OR THE USE, INABILITY TO USE OR RESULTS OF USE OF THE SOFTWARE PRODUCT EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

3.5 In no event will LICENSOR be liable for any third-party products used with, or installed in, the Software Product. LICENSOR does not warrant the compatibility of the Software Product with any third-party products, whether hardware or software.

3.6 The above sections do not apply for liability for damages caused by gross negligence or wilful default.

3.7 General Provision. This warranty shall not apply in any case of amendment or alterations of the Software Product made by LICENSEE.

#### 4. INTELLECTUAL AND PROPERTY INDEMNIFICATION.

4.1. LICENSOR agrees to indemnify and hold LICENSEE harmless from any final award of costs and damages against LICENSEE for any action based on infringement of any German intellectual property rights as a result of the use of the Licensed Software: (i) under the terms and conditions specified herein; (ii) under normal use; and (iii) not in combination with other items; provided that LICENSOR is promptly notified in writing of any such suit or claim against LICENSEE and further provided that LICENSEE permits LICENSOR to defend, compromise or settle the same and gives LICENSOR all available information, reasonable assistance and authority to enable LICENSOR to do so. LICENSOR'S LIABILITY TO LICENSEE PURSUANT TO THIS ARTICLE IS LIMITED TO THE TOTAL FEES PAID BY LICENSEE TO LICENSOR IN THE CALENDAR YEAR IN WHICH ANY FINAL AWARD OF COSTS AND DAMAGES IS DUE AND OWING.

#### 5. TRADE SECRETS AND PROPRIETARY INFORMATION.

5.1. LICENSEE acknowledges that LICENSOR is the owner of the Software Product, that the Software Product is confidential in nature and not in the public domain, that LICENSOR claims all intellectual and industrial property rights granted by law therein and that, except as set forth herein, LICENSOR does not hereby grant any rights or ownership of the Software Product to LICENSEE or any third party. Except as set forth herein, LICENSEE agrees not to copy or otherwise reproduce the Software Product, in whole or in part, without LICENSOR's prior written consent. LICENSEE further agrees to take all reasonable steps to ensure that no unauthorized persons shall have access to the Software Product and that all authorized persons having access to the Software Product shall refrain from any such disclosure, duplication or reproduction except to the extent reasonably required in the performance of LICENSEE'S rights under this Agreement.

5.2. LICENSEE agrees to accord the Software Product and the Documentation and all other confidential information relating to this Agreement the same degree and methods of protection as LICENSEE undertakes with respect to its confidential information, trade secrets and other proprietary data.

5.3. LICENSEE agrees not to challenge, directly or indirectly, the right, title and interest of LICENSOR in and to the Software Product, nor the validity or enforceability of LICENSOR's rights under applicable law. LICENSEE agrees not to directly or indirectly, register, apply for registration or attempt to acquire any legal protection for the Software Product or any proprietary rights therein or to take any other action which may adversely affect LICENSOR's right, title or interest in or to the Software Product in any jurisdiction.

5.4. LICENSEE acknowledges that, in the event of a material breach by LICENSEE of its obligations under this Article 5, LICENSOR may immediately terminate this Agreement, without liability to LICENSEE and may bring an appropriate legal action to enjoin any such breach hereof, and shall be entitled to recover from LICENSEE reasonable legal fees and costs in addition to other appropriate relief.

5.5. LICENSEE agrees to notify LICENSOR immediately and in writing of all circumstances surrounding the unauthorized possession or use of the Software Product and Documentation by any person or entity. LICENSEE agrees to cooperate fully with LICENSOR in any litigation relating to or arising from such unauthorized possession or use.

## 6. TERMINATION.

6.1. LICENSOR may terminate this Agreement at any time after the occurrence of any of the following events if LICENSOR provides 30 days notice of its intention to terminate as a result of the occurrence and LICENSEE fails to cure such occurrence within such 30 days:

(a) LICENSEE is declared or acknowledges that it is insolvent or otherwise unable to pay its debts as they become due or upon the filing of any proceeding

(whether voluntary or involuntary) for bankruptcy, insolvency or relief from creditors of LICENSEE;

(b) LICENSEE assigns or transfers this Agreement or any of its rights to obligations hereunder, without LICENSOR's prior written consent; or (c) LICENSEE violates any material provision of this Agreement, including without limitation, the payment obligations set forth in Addendum A.

6.2. LICENSEE may terminate this Agreement at any time after the occurrence of any of the following events if LICENSEE provides 30 days notice of its intention to terminate as a result of the occurrence and LICENSOR fails to cure such occurrence within such 30 days:

(a) LICENSOR is declared or acknowledges that it is insolvent or otherwise unable to pay its debts as they become due or upon the filing of any proceeding (whether voluntary or involuntary) for bankruptcy, insolvency or relief from creditors or LICENSOR; or

(b) LICENSOR violates any material provision of this Agreement.

6.3. Upon the termination of this Agreement for any reason, LICENSEE will discontinue all use of the Software Product and, within ten (10) days after termination, will destroy or delete all copies of the Software Product then in its possession, including but not limited to, any back-up or archival copies of the Software Product and Documentation. At LICENSOR's request, LICENSEE will verify in writing to LICENSOR that such actions have been taken.

6.4. No termination of this Agreement for any reason whatsoever shall in any way affect the continuing obligations of the parties under Articles 5 hereof.

## 7. APPLICABLE LAW

This LICENSE shall be deemed to have been made in, and shall be construed pursuant to, the laws of Germany, without reference to conflicts of laws principles. All controversies and disputes arising out of or relating to this Agreement shall be submitted to the exclusive jurisdiction of Esslingen am Neckar, Germany, as long as LICENSEE is deemed to be a merchant (as defined by Handelsgesetzbuch, §1-7). The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed.

## 8. GENERAL PROVISIONS.

8.1. This Agreement does not create any relationship of association, partnership, joint venture or agency between the parties.

8.2. This Agreement (including the Exhibit and Addendum attached to the Agreement) sets forth the entire agreement and understandings between the parties hereto with respect to the subject matter hereof. This Agreement merges all previous discussions and negotiations between the parties and supersedes and replaces any and every other agreement, which may have existed between LICENSOR and LICENSEE with respect to the contents hereof.



8.3. Except to the extent and in the manner specified in this Agreement, any modification or amendment of any provision of this Agreement must be in writing and bear the signature of the duly authorized representative of each party.

8.4. The failure of either party to exercise any right granted herein, or to require the performance by the other party hereto of any provision of this Agreement, or the waiver by either party of any breach of this Agreement, shall not prevent a subsequent exercise or enforcement of such provisions or be deemed a waiver of any subsequent breach of the same or any other provision of this Agreement.

8.5. Except in the case of merger, acquisition or the sale of substantial assets or equity of Licensee or assignment to any direct or indirect subsidiary or affiliate of LICENSEE, LICENSEE shall not sell, assign or transfer any of its rights, duties or obligations hereunder without the prior written consent of LICENSOR. LICENSOR reserves the right to assign or transfer this Agreement or any of its rights, duties and obligations hereunder, to any direct or indirect subsidiary or affiliate of LICENSOR.

8.6. All notices required by this Agreement must be sent by certified mail in order to be deemed effective when sent to the following:

FOR LICENSOR:

Frank Fock

Schlossstrasse 8

73765 Neuhausen, Germany

EXHIBIT A

Licensed Software

AgentX++

a. Source Code - (ANSI C++ for Linux, Solaris, Win32) Includes AgentX++ and Agent++Win32 Source Code.

b. Executable Code - AgentX++Win32 Master Agent (Win XP/2000/NT4)

ADDENDUM A

For evaluation purposes and non commercial use only, a free license is granted, provided that the LICENSEE accepts this license agreement.

In order to obtain a license to use AgentX++ in a commercial environment,

LICENSEE has to purchase a commercial license from LICENSOR. The actual pricing list and other related information can be found at <http://www.agentpp.com>

## **C.4. Agent++v3.5.28a library**

**The following terms regulate Agent++v3.5.28a library use:**

AGENT++ API Version 3.x

-----

Copyright (C) 2001 Frank Fock, Jochen Katz

### **LICENSE AGREEMENT**

WHEREAS, Frank Fock and Jochen Katz are the owners of valuable intellectual property rights relating to the AGENT++ API and wish to license AGENT++ subject to the terms and conditions set forth below; and WHEREAS, you ("Licensee") acknowledge that Frank Fock and Jochen Katz have the right to grant licenses to the intellectual property rights relating to AGENT++, and that you desire to obtain a license to use AGENT++ subject to the terms and conditions set forth below; Frank Fock and Jochen Katz grants Licensee a non-exclusive, non-transferable, royalty-free license to use AGENT++ and related materials without charge provided the Licensee adheres to all of the terms and conditions of this Agreement.

By downloading, using, or copying AGENT++ or any portion thereof, Licensee agrees to abide by the intellectual property laws and all other applicable laws of Germany, and to all of the terms and conditions of this Agreement, and agrees to take all necessary steps to ensure that the terms and conditions of this Agreement are not violated by any person or entity under the Licensee's control or in the Licensee's service.

Licensee shall maintain the copyright and trademark notices on the materials within or otherwise related to AGENT++, and not alter, erase, deface or overprint any such notice.

Except as specifically provided in this Agreement, Licensee is expressly prohibited from copying, merging, selling, leasing, assigning, or transferring in any manner, AGENT++ or any portion thereof.

Licensee may copy materials within or otherwise related to AGENT++ that bear the author's copyright only as required for backup purposes or for use solely by the Licensee.

Licensee may not distribute in any form of electronic or printed communication the materials within or otherwise related to AGENT++ that bear the author's copyright, including but not limited to the source code, documentation, help files, examples, and benchmarks, without prior written consent from the authors. Send any requests for limited distribution rights to [sales@agentpp.com](mailto:sales@agentpp.com).

Licensee hereby grants a royalty-free license to any and all derivatives based upon this software code base, that may be used as a SNMP agent development environment or a SNMP agent development tool.

Licensee may modify the sources of AGENT++ for the Licensee's own purposes. Thus, Licensee may not distribute modified sources of AGENT++ without prior written consent from the authors.

The Licensee may distribute binaries derived from or contained within AGENT++ provided that:

- 1) The Binaries are not integrated, bundled, combined, or otherwise associated with a SNMP agent development environment or SNMP agent development tool; and
- 2) The Binaries are not a documented part of any distribution material.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## **C.5. *Boost v 1.0* library**

**The following terms regulate Boost v 1.0 library use:**

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including

the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT

SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE

FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,

ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER

DEALINGS IN THE SOFTWARE.

## **C.6. *Milter* library**

**The following terms regulate Milter library use:**

The following license terms and conditions apply, unless a different license is obtained from Sendmail, Inc., 6425 Christie Ave, Fourth Floor, Emeryville, CA 94608, USA, or by electronic mail at [license@sendmail.com](mailto:license@sendmail.com).

License Terms:

Use, Modification and Redistribution (including distribution of any modified or derived work) in source and binary forms is permitted only if each of the following conditions is met:

1. Redistributions qualify as "freeware" or "Open Source Software" under one of the following terms:

a) Redistributions are made at no charge beyond the reasonable cost of materials and delivery.

b) Redistributions are accompanied by a copy of the Source Code or by an irrevocable offer to provide a copy of the Source Code for up to three years at the cost of materials and delivery. Such redistributions must allow further use, modification, and redistribution of the Source Code under substantially the same terms as this license. For the purposes of redistribution "Source Code" means the complete compilable and linkable source code of sendmail including all modifications.

2. Redistributions of source code must retain the copyright notices as they appear in each source code file, these license terms, and the disclaimer/limitation of liability set forth as paragraph 6 below.

3. Redistributions in binary form must reproduce the Copyright Notice, these license terms, and the disclaimer/limitation of liability set forth as paragraph 6 below, in the documentation and/or other materials provided with the distribution. For the purposes of binary distribution the "Copyright Notice" refers to the following language:

"Copyright (c) 1998-2004 Sendmail, Inc. All rights reserved."

4. Neither the name of Sendmail, Inc. nor the University of California nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission. The name "sendmail" is a trademark of Sendmail, Inc.

5. All redistributions must comply with the conditions imposed by the University of California on certain embedded code, whose copyright notice and conditions for redistribution are as follows:

a) Copyright (c) 1988, 1993 The Regents of the University of California. All rights reserved.

b) Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

i. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

ii. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

iii. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

6. Disclaimer/Limitation of Liability: THIS SOFTWARE IS PROVIDED BY SENDMAIL, INC. AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL SENDMAIL, INC., THE

REGENTS OF THE UNIVERSITY OF CALIFORNIA OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## **C.7. *libkavexim.so* library**

**The libkavexim.so library is distributed in accordance with GPLv2, and its use is regulated by the following terms:**

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program).

Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.



3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.

However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of

any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program

`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General.

Public License instead of this License.