KASPERSKY LAB

# Kaspersky® Mail Gateway 5.6

# ADMINISTRATOR’S GUIDE

KASPERSKY® MAIL GATEWAY 5.6

# Administrator's Guide

http://www.kaspersky.com

Revision date: July, 2008.

# Contents

# CHAPTER 1. KASPERSKY® MAIL GATEWAY 5.6

**Kaspersky® Mail Gateway 5.6**, (henceforth referred to as *Kaspersky Mail Gateway* or *the application*), filters SMTP e-mail traffic to protect e-mail system users against viruses and unwanted messages (spam). The application is a full-featured mail relay (compliant with IETF RFC internet standards) that runs under the Linux and FreeBSD operating systems.

The application allows the user to:

- Scan e-mail messages for viruses, including both attached objects and message bodies.
- Detect infected, suspicious, and password-protected attachments and message bodies.
- Perform anti-virus processing (including disinfection) of infected objects detected in e-mail messages by scanning.
- Filter e-mail traffic by the names and MIME types of attachments, and apply specified processing rules to the filtered objects.
- Check each message including attached objects for signs typical of spam.
- Check during anti-spam analysis the addresses of mail sender and recipient (envelope), message size and various headers (including *From* and *To*).
- Perform the following checks as a part of the anti-spam mail analysis:
  - Presence of the sender's IP address in a DNS-based real time black hole list (DNSBL).

> **Note**
>
> **DNSBL** (**DNS based black hole list**) is a database that lists IP addresses of mail servers used for uncontrolled mass mailing. Such servers receive mail from anyone and deliver it further to arbitrary recipients. Use of DNSBL allows automatic blocking of mail from such mail servers. Various services use different policies for generation of such lists. Please examine carefully the policy of each service before you start using it for mail filtration.

  - availability of a DNS record for the sending server (reverse DNS lookup);

  - a check of the sender's IP address for compliance with the list of addresses allowed for a domain, based on the Sender Policy Framework (SPF);

  - a check of addresses and links to web sites in the message text using the Spam URL Real-time Blocklists (SURBL) service.

- Scan also attached images, comparing them to the signatures of known spam messages, and take the comparison results into account to determine the status of the message.

- Maintain archives of all e-mail messages sent and/or received by the application, if required by the internal security policy of the company.

- Enable restrictions for SMTP connections, to provide protection against hacking attacks and to prevent the application being used as an open e-mail relay for unsolicited e-mail messages.

- Limit the load on your server by configuring the application's settings and SMTP parameters.

- Create white and black lists of senders and recipients applied during processing of e-mail traffic.

- Notify senders, recipients, and the administrator about disinfected letters, about messages containing infected, suspicious, or protected objects, and also about errors that have occurred during mail scanning.

- Quarantine messages identified as spam or probable spam, formal or blacklisted mail as well as messages containing infected and suspicious objects.

- Update the anti-virus and anti-spam databases of Kaspersky Mail Gateway. The application retrieves updates from Kaspersky Lab's update servers. You can also configure the application to update the databases from a local directory.

The application detects and cures infected objects using the anti-virus database. During scans, the contents of each file are compared to the sample code of known viruses contained in the database.

> **Attention!**
>
> Please remember that new viruses appear every day, and therefore you are advised to maintain the anti-virus databases in an up-to-date state. New updates are made available on Kaspersky Lab's update servers every hour.

The anti-spam databases are used during analysis of message contents (including *Subject* and other headers) and attached files. The application uses linguistic algorithms which compare the analyzed text with sample messages, and search for typical words and word combinations.

> **Attention!**
>
> Kaspersky Lab's Linguistic Laboratory continues to work on improving and supplementing the corpus of data used for spam detection. Efficient spam fighting requires that you regularly update the application's anti-spam databases. Updates for the databases are made available on Kaspersky Lab's update servers every three minutes.

The *keepup2date* component's function is to update the anti-virus and anti-spam databases (see section 5.1 on p. 46).

- Configure and manage Kaspersky Mail Gateway, either from a remote location using the Webmin web-based interface, or locally using standard operating system tools such as using command line options, signals, special command files or by modifying the application's configuration file.
- Monitor the antivirus protection, spam filtering status, application statistics and logs both locally and remotely using the Webmin interface.
- Obtain configuration data and statistics on application activity via SNMP and configure the application to generate and send SNMP traps upon occurrence of certain events.

## 1.1. What's new in Kaspersky Mail Gateway 5.6

Kaspersky Mail Gateway has the following additional features as compared to Kaspersky SMTP-Gateway 5.6:

- The application includes anti-spam module with the following features:
  - Increased performance and stability.
  - Low RAM requirements.
  - Low level of Internet traffic (updates to Kaspersky Mail Gateway databases).

- Improved filtration methods are used, namely:
    - Algorithms for parsing of HTML objects in e-mail messages (increasing the efficiency of protection against various spammer tricks devised to bypass filtration systems).
    - System for analysis of e-mail message headers.
    - System for analysis of graphical attachments (GSG).
    - Sender Policy Framework (SPF) and Spam URL Realtime Blocklists (SURBL) services.
    - Internal Urgent Detection System (UDS) service, which allows obtaining information about certain types of spam in real time.
- Individual settings available for user groups: certain scanning methods can be enabled/disabled separately for every group; you can also define the actions to be performed over e-mail messages.
- Collection of configuration data and statistics of application activity via SNMP; the application can be configured to send SNMP traps when certain events occur.
- Redesigned subsystem accepting incoming mail consumes fewer resources and supports more simultaneous incoming connections.

# 1.2. Licensing policy

The licensing policy for Kaspersky Mail Gateway 5.6 limits product use based on these criteria:

- **Number of users** protected by the application.
- **E-mail traffic** processed daily (MB/day).

Each type of license also has a time limit, typically one or two years from the date of purchase.

At the time of purchase, you can specify which type of license limitation you require (for example, by the daily e-mail traffic volume).

In addition, you can choose during product purchase whether your copy of Kaspersky Mail Gateway will only perform anti-virus scanning of e-mail traffic, or if it will also filter spam.

The application has slightly different configuration parameters depending on the type of license you purchased. For instance, if the license is issued for a certain **number of users**, you will have to create a list of addresses (domains) that will be protected by the application against viruses and spam. The application will notify the administrator when the license limitations are reached: in this case, when the number of protected accounts is exceeded.

## 1.3. Hardware and software requirements

The minimum system requirements for normal operation of Kaspersky Mail Gateway are as follows:

- Hardware requirements:
    - Intel Pentium® processor (Pentium III or Pentium IV recommended).
    - At least 256 MB of available RAM.
    - At least 100 MB of available space on your hard drive to install the application.

      **Attention!**

      Please note that the application's working queue, quarantine directory, and archives of incoming and outgoing e-mail are not included in the hard disk space required. If your network security policy requires the use of these features, additional disk space will be needed.

    - At least 500 MB of available space in the */tmp* file system.
- Software requirements:
    - One of the following operating systems for 32-bit platforms:
        - Red Hat Enterprise Linux Server 5.
        - Fedora 7.
        - SUSE Linux Enterprise Server 10.
        - OpenSUSE Linux 10.3.
        - Debian GNU/Linux 4 r1.
        - Mandriva 2007.
        - Ubuntu 7.10 Server Edition.
        - FreeBSD 5.5, 6.2.

- One of the following operating systems for 64-bit platforms:
    - Red Hat Enterprise Linux Server 5.
    - Fedora 7.
    - SUSE Linux Enterprise Server 10.
    - OpenSUSE Linux 10.3.
- *Perl* interpreter, version 5.0 or higher (www.perl.org), bzip2 utility for unpack spam filtration bases, and the which utility for application installation.
- *Webmin* version 1.070 or higher (www.webmin.com) to install the remote administration module (optional).

# 1.4. Distribution kit

You can purchase the product either from our dealers or at one of our online stores (for example, www.kaspersky.com/store – follow the **E-store** link).

If you purchase our application online, you will download it from Kaspersky Lab's website. Your product key will be sent to you by e-mail after payment.

The License Agreement constitutes a legal agreement between you and Kaspersky Lab, containing the terms and conditions under which you may use the purchased software.

# 1.5. Help desk for registered users

Kaspersky Lab offers an extensive service package enabling registered customers to boost the productivity of Kaspersky Mail Gateway.

After purchasing the product key, you become entitled to receive the following services for the validity period of your key:

- new versions of the application provided free of charge.
- phone or e-mail support on matters related to the installation, configuration, and operation of the product you have purchased. You can contact the Technical Support service for consulting using any of the following methods:
    - Make a phone call to Technical Support.

    - Create and send a request using the Technical Support web site (http://www.kaspersky.com/helpdesk) or your personal user cabinet.

- notifications about new software products from Kaspersky Lab, and about new virus outbreaks. This service is provided to users who subscribe to Kaspersky Lab's e-mail newsletter service.

**Note**

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

# CHAPTER 2. APPLICATION STRUCTURE AND TYPICAL DEPLOYMENT SCENARIOS

The correct configuration of the application, and its efficient operation, require knowledge of its structure and internal algorithms. It is also important for the application's deployment within an existing corporate e-mail system. This chapter discusses in detail the application's structure, architecture and operating principles, as well as typical deployment scenarios.

## 2.1. Application architecture

A review of the application's functionality must be preceded by a description of its internal architecture.

Kaspersky Mail Gateway is a fully-featured Mail Transfer Agent (MTA), able to receive and route e-mail traffic, which also scans e-mail messages for viruses, and filters spam.

The application uses SMTP protocol commands (RFC 2821), the Internet message format (RFC 2822), MIME format (RFC 2045-2049, 2231, 2646), and satisfies the requirements for e-mail relays (RFC 1123). In compliance with anti-spam recommendations (RFC 2505 standard), the application uses access control rules for SMTP clients to prevent the use of this application as an open relay. In addition, Kaspersky Mail Gateway supports the following SMTP protocol extensions:

- Pipelining – enhances performance of servers supporting this mode of operation (RFC 2920).
- 8-bit MIME Transport – processes code tables of national language characters (RFC 1652).
- Enhanced Error Codes – provides more informative explanations of protocol errors (RFC 2034).
- DSN (Delivery Status Notifications) – decreases bandwidth usage and provides more reliable diagnostics (RFC 1891, 3461-3464).
- SMTP Message Size – Decreases the server load and increases transfer rate (RFC 1870).

**Note**

The RFC documents mentioned above are available at: http://www.ietf.org.

The application includes these components:

- *mailgw* – the main application component – a fully-featured e-mail relay with built-in anti-virus protection and spam filtering.
- *licensemanager* – the component which manages product keys (their installation, removal, and statistics).
- *keepup2date* – the component that updates the anti-virus and anti-spam databases, by downloading the updates either from Kaspersky Lab's update servers or from a local directory.
- *Webmin* module – for remote administration of the application using a web-based interface (optional installation). This component allows the user to configure and manage the database updating process, specify the actions to be performed on detected objects, and monitor the application's operation.

The main application component (see Fig.1), in turn, consists of these modules:

- *Receiver*, which receives incoming e-mail.
- *Sender*, which sends out messages which have passed anti-virus scanning and spam filtering.
- *AS module* which performs anti-spam analysis of e-mail, its classification and processing.
- *AV module*, the anti-virus engine.
- *Scanning module,* which acts with the AS and AV modules to process messages, providing anti-virus scanning and spam filtering of e-mail traffic.

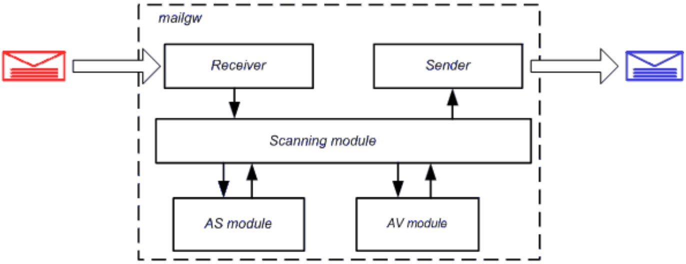


Figure 1. General architecture of Kaspersky Mail Gateway

# 2.2. The main application's algorithm

The application works as follows (see Fig. 2):

1. The e-mail agent receives e-mail messages via the SMTP protocol, and passes them to the Receiver module.

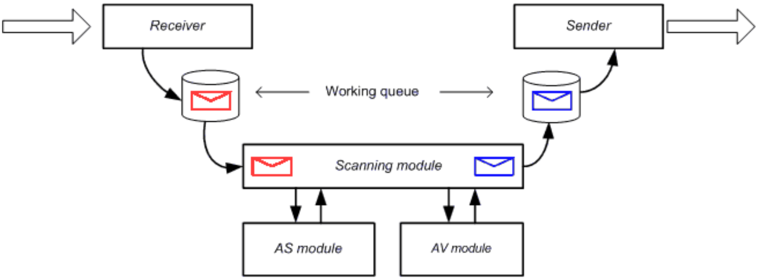


Figure 2. Working queue of Kaspersky Mail Gateway

2. The Receiver module performs preliminary e-mail processing using the following criteria:

   - presence of the sender's IP address in the list of blocked and/or trusted addresses including masks;
   - compliance with the access restrictions specified for SMTP connections (see section 5.5.2 on p. 70);
   - compliance of the message size (and the total number of messages within the session) with the limits specified in the application's settings;
   - compliance of the number of open sessions (both the total number from all IP addresses, and from a single IP address) with the limits specified in the application's settings.

   If the message satisfies the preliminary processing requirements, it is sent to the working queue to be processed by the scanning module.

   If the option to archive all incoming e-mail has been selected, a copy of any message added to the working queue will be automatically preserved in the archive of received messages.

   Blind carbon copies of each message can also be sent to a specified list of e-mail addresses before scanning of the received mail.

3. The Scanning module receives a message from the working queue and transfers it to the anti-spam module for inspection.

   The anti-spam module consists of the following components:

   - Filtration master process and filtering processes which perform actual mail analysis.
   - Licensing daemon which verifies the presence of a valid key file and compliance with the restrictions defined in the key.
   - Daemon processing SPF requests.
   - Auxiliary programs and scripts including the script compiling the anti-spam databases.

   The main component of the anti-spam module is the filtering master process (*mailgw-process-server*) performing the following functions:

   - Monitoring of requests for connection to filtering processes from the application Scanning module.
   - Launch of new filtering processes when there are no more available ones.
   - Control of the status of running filtering processes.
   - Termination of child processes upon an appropriate signal.

   Filtering process (*ap-mailfilter*) receives at launch message header and body, scans them and returns the results.

   If message sender should be checked for compliance with the existing SPF policy, the filtering process sends a request to SPF daemon (*mailgw-spfd*), which performs necessary queries to DNS server and returns the results to the filtering process.

   Message analysis and application of rules defined by the parameters in configuration file are only performed when a valid product key is present.

   All license-related checks are performed by the licensing module (*kas-license*) upon request from a filtering process.

   Having finished message processing, a filtering process keeps running expecting a new request. A filtering process is terminated after it has handled the maximum number of messages specified for a single process (usually 300) or if it remains idle for a long time.

   The AS module assigns to message a certain status based on the inspection results, and returns the message to the Scanning module, which breaks it into its components and passes them to the AV module for analysis.

> **Attention!**
>
> If you have only purchased a license for anti-virus scanning of e-mail traffic, spam filtering will not be performed. Messages will be delivered directly to the AV module for scanning, and any configuration parameters which apply to the anti-spam module are ignored.

4. The AV module scans the objects and, if this option is enabled, disinfects them when necessary.

5. The Scanning module handles messages according to the status (see section 4.2 on p. 36) assigned to each part of the message during analysis by the AS and AV modules. Possible actions include blocking message delivery, deleting infected objects, modifying message headers, and moving the message to the quarantine directory. The actions to be applied are specified in the application's configuration file. Each processed message is then added to the ready-to-send message queue.

6. If the application's configuration specifies that detected messages are to be saved in quarantine, a copy of the scanned message will be saved in the quarantine directory concurrently with its transfer to the ready-to-send queue. The application creates separate quarantine directories for messages identified as spam or probable spam (after anti-spam analysis), and for messages containing infected or suspicious objects (after anti-virus scanning).

> **Note**
>
> The creation of a copy of a message in backup storage or the quarantine directory does not block delivery of the original message to the recipient. An additional action blocking its delivery must be specified to prevent message delivery to the recipient.

7. The Sender module receives each message from the ready-to-send queue, and transfers it via the SMTP protocol to the onward e-mail agent to be delivered to local end users or rerouted to other mail servers.

8. If your network security policy requires logging of all outgoing e-mail traffic, a copy of each message will be automatically stored in the archive of sent messages after it is dispatched (see Fig. 3).

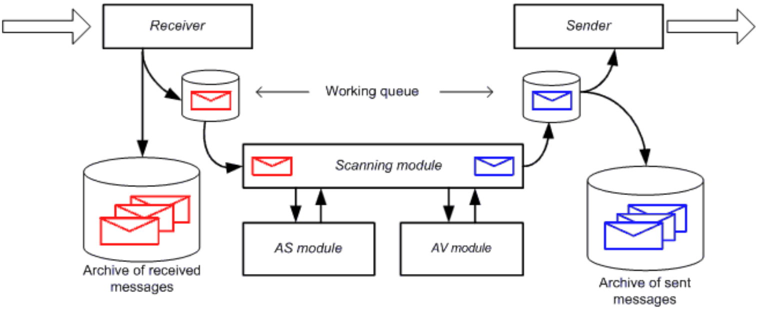


Figure 3. Saving messages to the archives of received / sent messages

# 2.3. Typical deployment scenarios

Depending upon the network architecture, there are two options for installing Kaspersky Mail Gateway:

- install the application within a demilitarized zone (DMZ) acting as a buffer between the internal corporate LAN and the external network;
- install the application inside the perimeter of the corporate network, as part of your existing e-mail system.

In each of the above cases the application can be installed:

- on the same server as the running e-mail system;
- on a dedicated server.

The sections below discuss these scenarios in detail and describe their advantages.

**Attention!**

The application, being an e-mail relay, does not include a local e-mail delivery agent (MDA). Therefore, **all deployment scenarios require an e-mail system (or e-mail systems)** to deliver e-mail messages to local users within protected domains.

## 2.3.1. Installing the application in a demilitarized zone

The main advantage of this deployment option is that it improves the overall performance of your e-mail system, by minimizing the number of transfer cycles for e-mail messages. It also provides additional protection for data, because the existing corporate mail server in that case has no connection to the Internet.

This is an overview of how to install the application and the e-mail system *on the same server*, so that they work together:

1. Configure all interfaces of Kaspersky Mail Gateway to listen on port 25 for incoming e-mail traffic from all IP addresses which match the relevant MX records for the protected domain.
2. The application will filter spam and scan e-mail, and then transfer processed messages to the corporate e-mail system via a different port (e.g., 1025).

   > **Attention!**
   >
   > You must set up restrictions for the e-mail transfer agent (MTA) receiving e-mail from Kaspersky Mail Gateway via port 1025, so that it accepts messages *exclusively* from Kaspersky Mail Gateway (e.g., configure mail receipt from the localhost (127.0.0.1) interface only). Otherwise, it will be possible to bypass the application with a connection established directly from the external network through port 1025.

3. The e-mail system, configured to use a local interface, delivers messages to users.

*Follow these steps to install the application and the e-mail system on the same server:*

- Configure the application to receive e-mail via port 25 on all the server's network interfaces. To do this, specify the following value in the **[mailgw.network]** section of the configuration file:

  ```
  ListenOn=0.0.0.0:25
  ```

- Specify in the routing table that all scanned messages will be transferred to the e-mail system via port 1025. To do this, specify the following value in the **[mailgw.forward]** section of the application's configuration file:

  ```
  ForwardRoute=<company_mask> [localhost:1025]
  ```

  where: `<company_mask>` is the mask for recipient addresses.

- Change the settings of the existing e-mail system to receive messages only from the application via port 1025. This will ensure that all incoming e-mail messages are received, and that they are delivered to local users within the protected domains of the company.
- Set up the existing e-mail system to transfer all the messages it receives to the application via port 25. This will ensure anti-virus scanning and anti-spam filtering of all outgoing e-mail messages from local users.
- Specify a list of all corporate local domains as the value for the **ProtectedDomains** option in the **[mailgw.forward]** section of the application configuration file ("*" and "?" wildcards can be used). E-mail messages for the specified domains will be scanned.

> **Attention!**
>
> These are the default application configuration settings for this deployment scenario, which will be stored in the configuration file by the installation process.

When the application is installed on a dedicated server, its operation algorithm is identical to when it is installed on the same server as the e-mail system, but the settings will differ. The IP address of the server on which the application is installed, must be included in MX records corresponding to the protected domain.

*To install the application on a dedicated server:*

- Configure the application to receive mail via port 25 on all the server's network interfaces, by specifying the following value in the **[mailgw.network]** section of the application's configuration file:

  ```
  ListenOn=0.0.0.0:25
  ```

- Specify in the routing table that all scanned messages must be transferred to the e-mail system via port 25, by setting the following value in the **[mailgw.forward]** section of the application's configuration file:

  ```
  ForwardRoute=<company_mask> [host:25]
  ```

  where: `<company_mask>` is the mask for recipient addresses, and will generally be of the form `*@company.com`

  `host` – name of your corporate e-mail server.

- Specify the list of all local corporate domains as the value for the **ProtectedDomains** option in the **[mailgw.network]** section of the application configuration file ("*" and "?" wildcards can be used). e-mail messages for the specified domains will be scanned.

**Attention!**

This is the most convenient deployment scenario, especially if Kaspersky Mail Gateway is installed at the same time as the network is deployed and the company's e-mail system is installed.

## 2.3.2. Installing the application inside the corporate network's perimeter

One advantage of installing the application inside the corporate perimeter is that there is no external access to the information that the application is running on the server, or to its configuration. Additionally, if the application is installed on a dedicated server, the load of performing anti-virus scanning can be distributed amongst several servers.

This is how the application and the e-mail system work together if they are installed *on the same server*:

1. Duplicate your e-mail system and configure one of the copies to listen on port 25, and receive e-mail messages via all available interfaces.
2. This e-mail system forwards all incoming messages through the local interface via a different port (port 1025, for instance) to the application for scanning and spam filtering.
3. The application filters spam, scans the e-mail messages for viruses and forwards scanned and processed messages to the second e-mail system copy, which receives e-mail on a different port (e.g., port 1026).
4. The second e-mail system delivers e-mail to the local users.

**Attention!**

This deployment scenario is recommended if you are sure of the reliability of your e-mail system. Installing the application in this configuration will not affect the stability of your e-mail system.

Installing the application on a *dedicated server* is similar to the above procedure. Additionally when installing the application on a dedicated server, you can create and run several copies of the application on different servers, enabling you to distribute the load of anti-virus processing and spam filtering amongst these several servers.

*To deploy the application on a dedicated server:*

Specify the list of all local corporate domains as a value for the **ProtectedDomains** option in the **[mailgw.network]** section of the application configuration file ("*" and "?" wildcards can be used). E-mail messages for the specified domains will be scanned.

**Attention!**

Deploying Kaspersky Mail Gateway may require changes to the settings for e-mail clients throughout company, to ensure that all outgoing e-mail messages are delivered to the application. These messages will be transferred to the external network after an anti-virus scan and spam filtration.

**Attention!**

If the network includes installed firewalls or demilitarized zones (DMZ’s), it is necessary to provide e-mail clients and internal and external network servers with access to the installed application to ensure joint operation and routing of the e-mail traffic.

# CHAPTER 3. INSTALLING THE APPLICATION

Before installing Kaspersky Mail Gateway, it is necessary to:

- Make sure that your system meets the hardware and software requirements (see section 1.3 on p. 12).
- Configure your Internet connection. The application distribution package does not contain the anti-virus and anti-spam databases, which are required to perform anti-virus protection and filter spam.
- Log on to the system as **root**, or as a privileged user.

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

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

To install the application under Linux Red Hat, Linux SUSE or Linux Mandriva, use the *rpm* package.

*To initiate installation of Kaspersky Mail Gateway from the rpm package, enter the following at the command line:*

```
# rpm -i <distribution_package_file_name>
```

**Attention!**

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/mailgw/lib/bin/setup/* directory (in Linux) and in the */usr/local/libexec/kaspersky/mailgw/setup* directory (in FreeBSD).

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

*To initiate installation of Kaspersky Mail Gateway from the deb package, enter the following at the command line:*

```
# dpkg -i <distribution_package_file_name>
```

After you enter the command, the application will be installed automatically.

**Attention!**

The procedure of application setup under Mandriva distributions has some peculiarities. You might have to perform some additional configuration to ensure the correct functioning of the application on such systems (please see Chapter 9 on p. 103 for details).

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

The distribution file for installing Kaspersky Mail Gateway on servers running FreeBSD OS is supplied as a *pkg* package.

*To initiate installation of Kaspersky Mail Gateway 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.

## 3.3. Installation procedure

**Attention!**

Installation errors can occur for a number of reasons. If an error message is displayed, firstly make sure that your computer satisfies the hardware and ware requirements (see section 1.3 on p. 12) and that you have logged on to the system as **root**.

The application installer script applies these steps:

### Step 1. Preparing the system

At this stage, the installation script creates the system group and user account for the application. The default group is **klusers** and the default user account is **kluser**. In future, the application will start under this user account (not **root**) to provide additional security for your system.

## Step 2. Copying application files to destination directories on your server

The installer starts copying the application files to the destination directories on your server. For a detailed description of the application's directories, see section B.1 on p. 149.

**Attention!**

If you installed the application from an *rpm* package, you should run the *postinstall.pl* script (present by default in the */opt/kaspersky/mailgw/lib/bin/setup/* directory in Linux and in */usr/local/libexec/kaspersky/mailgw/setup* in FreeBSD) to perform the next step, Post-installation tasks.

## Step 3. Post-installation tasks

The post-installation configuration includes these steps:

- Configuring the *main application component* (see section 3.4 on p. 28).
- Installing and registering the product key.

  If you do not have a product key at the time of installation (for example, if you purchased the application via the Internet and have not yet received the license key), you can activate the application after installation and before its first use: for details, see section 5.6 on p. 71. Please note that if the key is not installed, the anti-virus and anti-spam databases cannot be updated and the *main application component* cannot be started during the installation process. In this case it must be done manually, after the license key is installed.
- Configuring the *keepup2date* component.
- Installation (updating) of the anti-virus and anti-spam databases.

  You must install the anti-virus and anti-spam databases before using the application (see section 5.6 on p. 71). The procedure of detecting and disinfecting viruses relies on the anti-virus database which contains the descriptions of all currently known viruses, and the methods of disinfecting these viruses. Anti-virus scanning and processing of e-mail messages cannot be performed without the anti-virus database. The anti-spam database is used for spam detection, which analyzes the contents of messages and attached files to identify the signs of unsolicited e-mail.
- Installing the Webmin module.

  The Webmin module for remote management of the application can be installed correctly only if the Webmin application is located in the default directory. After the module is installed, you will receive detailed instructions on how to configure it to work with the application.

- Launching the *main application component*.

**Attention!**

If after installation, Kaspersky Mail Gateway has not started working as required, check the configuration settings. Pay special attention to the port number you specified for receiving e-mail traffic. You should also view the application log file for error messages.

After these steps are properly completed, a message on the server console will indicate that installation has been successful.

# 3.4. Configuring the application

Immediately after the application's files have been copied to your server, the system configuration process will start. The configuration process will either be started automatically or, if the package manager (such as *rpm*) does not allow the use of interactive scripts, some additional actions will have to be performed by the administrator. All settings are stored in the *mailgw.conf* file which is installed by default in the */etc/opt/kaspersky/* directory in Linux, and in the */usr/local/etc/kaspersky/* directory in FreeBSD.

**Attention!**

If you are using the *rpm* installation package, enter the following command to start post-installation configuration (in Linux):

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

In FreeBSD:

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

The configuration procedure includes the following tasks:

- Specifying (by the administrator) the full domain name of the server that will be used to identify the application in SMTP commands when creating the DSN and notifications: this is the **Hostname** parameter in the **[mailgw.network]** section of the *mailgw.conf* configuration file.
- Assigning addresses to be used by the application:
  - Assign the **Postmaster** address (**[mailgw.network]** section, **Postmaster** parameter).

  - Assign the sender's return address for notifications (**[mailgw.policy]** section, **NotifyFromAddress** parameter).
  - Define the administrator's address (**[mailgw.policy]** section, **AdminNotifyAddress** parameter).
  - Allow incoming e-mail to the specified domain (**[mailgw . access]** section, **RelayRule** parameter).

- Defining the interface and port on which to listen for incoming e-mail traffic (**[mailgw.network]** section, **ListenOn** parameter). The port name and the IP address should be entered in the format `<x.x.x.x:z>`,

  where:

  x.x.x.x is the IP address, and

  z is the port number.

- Specifying local network identifiers. This value is used to assign rules for message delivery and processing (**[mailgw.access]** section, **RelayRule** parameter), for example, rules specific to your organization concerning e-mail processing, or blocking e-mail messages from certain domains. Specify the values using the following formats: `<x.x.x.x>` or `<x.x.x.x/y.y.y.y>`, or `<x.x.x.x/y>,`

  where:

  x.x.x.x is the IP address, and

  y.y.y.y or y is the subnet mask.

- Specifying (when necessary) the server to which all processed messages will be forwarded (**[mailgw.forward]** section, the **ForwardRoute** parameter). Type the host name in the format: `<x.x.x.x:z>`,

  where:

  `x.x.x.x` is the IP address, and

  `z` is the port number.

- Specifying the proxy server name (**[updater.options]** section, **ProxyAddress** parameter). This option is necessary for computers connected to the Internet via a proxy server.

- Confirmation of UDS installation and use.

  UDS service allows blocking spam in a timely manner before updates to Kaspersky Mail Gateway databases are downloaded. You are advised to disable UDS checks only if the method considerably decreases the filtration server performance or if the server cannot contact the UDS servers of Kaspersky Lab. Please refer to section 4.3.4 on page 41 for details on UDS service.

> **Attention!**
>
> To increase UDS efficiency, specify regular launch of the task that determines the time for access to UDS servers (see section 5.2.4 on page 55).

- Modifying the application configuration file to fine-tune the operation of the AV and AS modules (optional).

If all the above steps have been successfully completed, the configuration file will contain all settings that are required to start working with the application.

During Kaspersky Mail Gateway 5.6 installation you can choose to use saved settings of previous product version 5.5.139 installed earlier. In that case you will be offered to:

- Specify the path to the configuration file of an earlier version.
- Move or copy files from the queue, archives and Quarantine of the earlier version to the corresponding directories of the new one.
- Use UDS because that feature was introduced in version 5.6 (see above).

Application databases will be downloaded as well.

If the configuration file of an earlier version is not available or if you do not wish to use it, post-install setup will consist of the steps described above.

> **Attention!**
>
> After the system is installed and configured, it is recommended that you check the settings for Kaspersky Mail Gateway and test its performance. For more details, see Chapter 7 on page 97.

## 3.5. Installing the Webmin module to manage Kaspersky Mail Gateway

The activity of Kaspersky Mail Gateway 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.

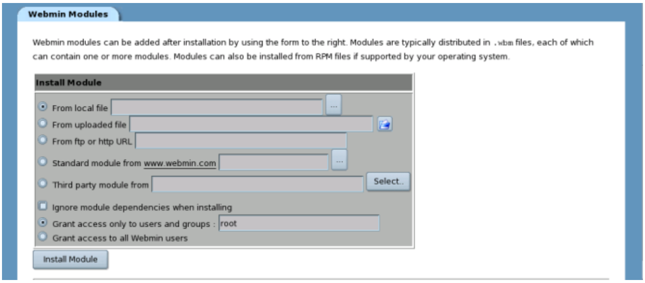
Kaspersky Mail Gateway's distribution package contains a Webmin module that can either be connected during the application's post-installation configuration (see section 3.3 on p. 26) if the system already has Webmin installed, or at any time later after Webmin is installed.

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

If the default settings were used during Webmin installation, the program can be accessed from a web browser using HTTP / HTTPS to connect to port 10000, as soon as the installation procedure is finished.

*To install the Webmin module to control Kaspersky Mail Gateway:*

1. Use your web browser to access Webmin with administrative 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 4).



Webmin modules can be added after installation by using the form to the right. Modules are typically distributed in .wbm files, each of which can contain one or more modules. Modules can also be installed from RPM files if supported by your operating system.

Install Module

● From local file

○ From uploaded file

○ From ftp or http URL

○ Standard module from www.webmin.com

○ Third party module from

□ Ignore module dependencies when installing

● Grant access only to users and groups : root

○ Grant access to all Webmin users

Figure 4. **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/mailgw/share/contrib/* directory (for Linux distributions), or the */usr/local/share/mailgw/contrib/* 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 Mail Gateway by clicking its icon within the **Others** tab (see Figure 5).

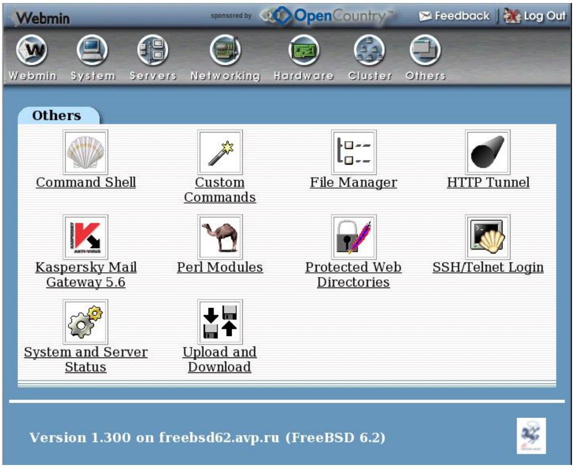


Figure 5. The icon of Kaspersky Mail Gateway in the **Others** tab

# CHAPTER 4. THE PRINCIPLES OF THE APPLICATION'S OPERATION

This chapter describes in more detail how the application works and the interaction between its components, and gives information required for correct software setup.

## 4.1. Creating groups of recipients/senders

A Recipients/Senders group is defined as a specified list of recipient/sender e-mail addresses. A particular e-mail message may be assigned to a particular group depending on whether this group contains the message sender's address (or sender IP) or the recipient's address, which are specified in the MAIL FROM and RCPT TO parts of the message header.

The administrator can specify individual rules for processing each e-mail message depending on the group of recipients/senders. Therefore, it is particularly important that the addresses are associated with the correct group.

When processing a message, the application searches through the list of addresses for each specific group. If it finds a matching combination of sender/recipient addresses, the rules defined for this group will be applied to the e-mail message.

**Note**

The anti-virus and spam filtering functionality of Kaspersky Mail Gateway depends on the configuration file settings. Changes to the configuration file can be made either locally or remotely (using the Webmin remote administration module).

The configuration file contains the **[mailgw.policy]** section that implicitly defines the **policy** group, which determines the default rules for processing e-mail messages.

**Attention!**

Both the section **[mailgw.policy]**, and all the parameters specified in the section, are mandatory.

The **[mailgw.policy]** section does not contain names of senders and recipients. The section **[mailgw.policy]** defines the default rules which are applied to all messages which **do not** belong to other groups explicitly described in **[mailgw.group:group_name]** sections.

All parameters in **[mailgw.group:group_name]** sections are optional. If a parameter value in such a section is not specified, it will be taken from the corresponding parameter in the **[mailgw.policy]** section.

The configuration file included in the application's installation package contains the following rules in the **policy** group. Messages which are not assigned to another group will be processed using the following rules (defined in the **[mailgw.policy]** section):

- Check all e-mail messages for indications of spam.
- Scan all e-mail messages for viruses.
- Deliver to recipients just messages which contain clean or disinfected objects only.
- Remove the following from messages: infected objects, objects which caused errors during their analysis, suspicious objects and password-protected and damaged objects.
- Notify recipients and the administrator about infected, suspicious, protected or filtered objects in messages and any objects which caused errors during analysis.

The parameters of the **policy** group can be altered, and new groups created. To process e-mail messages belonging to different groups of recipients/senders using different rules, you will have to create several groups.

*To create a new group of user addresses:*

1. Create a section **[mailgw.group:group_name]** in the configuration file.
2. Specify sender addresses (address masks, IP addresses, host names, masks for host names, subnets) and recipient addresses (address masks) as the values of **Senders** and **Recipients** parameters. To define several addresses or address masks, each record must be entered in a new line:

```
Senders=user1@example.com
Senders=*@internal.local
Senders=ip 192.168.0.1
```

```
Senders=ip 192.168.0.0/255.255.0.0
Senders=host example.com
Senders=network MyNetwork
Recipients=*@management.local
Recipients=help@helpdesk.local
```

"*" and "?" wildcards may be used to define masks. If a group description contains no **Recipients** or **Senders** parameter, the application will use the default value, "***@***" (i.e. all addresses)**.** At least one of the **Senders** or **Recipients** parameters must be specified.

**Attention!**

If you leave the **Senders** or **Recipients** parameter in a group description empty, e.g.:

**Senders=**

then no messages will be processed using the rules specified for that group. To use the default value for a parameter, delete (or place a comment mark before) the corresponding parameter from the group description.

If you have added other groups to the configuration file, the application will process messages from these groups as follows:

1. The application first compares the message address(es) with addresses in the groups created by the administrator. If the recipient/senders addresses pair is found in a specific group, the rules defined for that group will be applied to the message.

   **Attention!**

   If a sender/recipient address fits several groups, the application will use the rules for the first of those groups.

2. If the message addresses do not match any group created by the administrator, the message will be processed according to the rules described in the **policy** group.

   **Attention!**

   If a message has several recipients belonging to different groups, virtual copies of the initial message will be created to match the number of such groups. Each copy will be processed individually, according to the rules specified by the particular group.

Figure 6 demonstrates the sequence of actions applied by the application to a received e-mail message.

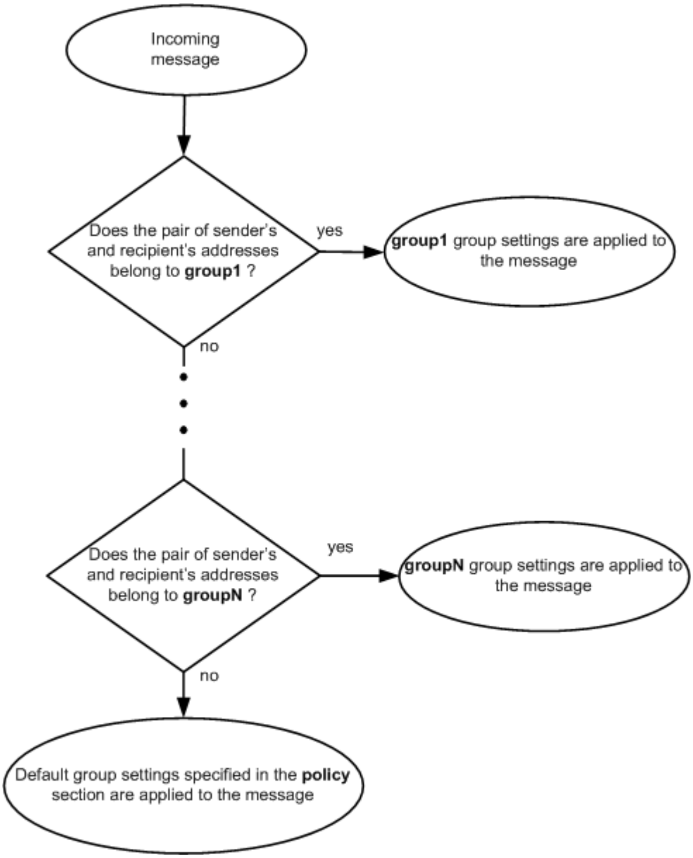

Incoming message
Does the pair of sender's and recipient's addresses belong to **group1** ?
yes
**group1** group settings are applied to the message
no
Does the pair of sender's and recipient's addresses belong to **groupN** ?
yes
**groupN** group settings are applied to the message
no
Default group settings specified in the **policy** section are applied to the message

Figure 6. Message processing

## 4.2. General message processing algorithm

This section discusses how the application processes e-mail messages. When the server receives an e-mail message, the scanning module:

1. Determines which group of recipients this message belongs to.
2. If the message has multiple recipients belonging to different groups, several virtual copies of this message are created to match the number

of groups, so that the respective group rules for anti-spam filtering and anti-virus scanning can be applied to each of the copies.

3. Then the application transfers the message for analysis by the anti-spam module.

   **Attention!**

   If you have only purchased a license for anti-virus scanning of e-mail traffic, spam filtering will not be performed. Messages will be immediately delivered to the AV module for scanning (Step 4). The application will ignore any configuration parameters which apply to the anti-spam module.

   Please refer to section 4.3 on page 38 for details on the operation of the anti-spam module.

   After processing, the anti-spam filter returns messages to the scanning module*.*

   If a message has been assigned the status of **Spam, Probable Spam**, **Formal** or **Blacklisted** and the application is configured to block such messages (the **BlockMessage** parameter is assigned the **as**/**spam**, **as**/**probable**, **as**/**formal**, **as**/**blacklisted** value), then anti-virus message scanning will be skipped. Further actions of the application are described in Step 8.

4. Using a built-in MIME format identifier (MIME, RFC2822, UUE), the application divides the message into its components: headers, message body and attachments.

5. If the application is configured to filter objects by name and/or attachment type, it will apply the specified filtering rules for this message. If the message meets the filter conditions, the object will be assigned the **Filtered** status and will not be subjected to further anti-spam scanning.

6. Each of the received objects is then sent to the AV module that analyzes each object and returns the status assigned to it.

7. Depending on the status assigned to each object, the application performs actions as specified in the settings for the respective group (please see section 4.4 on page 44 for basic actions of the AV module) in the configuration file.

8. After the anti-virus scan of all the message's components, and the execution of required actions on those components, an additional action can be performed on the message as a whole:

   - Add label to the message title (Subject) in accordance with the results of its anti-spam analysis (see section 4.3.5 on page 42).

- Append additional informational fields to the message's header or body (see section 6.12 on p. 93).
- Block delivery of messages to the recipients; see section 5.2.7 on p. 57 for an example of blocking the delivery of spam messages, and section 5.3.3 on p. 61 for messages containing infected objects.
- Create and send notifications to the sender, administrator, and recipient (see example in section 5.3.4 on p. 62).
- Quarantine a message; see section 5.2.8 on p. 58 for an example of quarantining spam messages, and section 5.3.6 on p. 64 for messages containing infected objects.

## 4.3. Operation of the anti-spam module

Spam filtration by the anti-spam module is performed during the third step of the procedure described in section 4.2 on p. 36. This section contains a brief overview of the spam detection technologies implemented in the application, namely:

- Analysis of formal signs (see section 4.3.1 on page 39).
- Content filtration (see section 4.3.2 on page 40).
- Checks involving external services (see section 4.3.3 on page 41).
- Urgent Detection System technology (see section 4.3.4 on page 41).
- During all inspection stages, message analysis is performed according to the required filtering intensity, defined in the application configuration file (**SpamRateLimit** option in the **[mailgw.policy]** or **[mailgw.group:group_name]** section).

The following degrees of filtering intensity are available:

- Minimum (**SpamRateLimit=minimum**).
- Standard (**SpamRateLimit=standard**).
- High (**SpamRateLimit=high**).
- Maximum (**SpamRateLimit=maximum**).

The application decides if a message contains spam based on several signs detected in mail by the anti-spam module. The higher is filtering intensity, the smaller is the number of signs required to recognize a message as spam. When the specified filtering intensity is lower, the same set of signs can only result in message identification as suspicious (**Probable Spam**) or even normal.

**Note**

The **Standard** level of filtering intensity is recommended.

Higher level of filtering intensity can be used in cases, when the application does not detect spam or when it recognizes spam as suspicious mail (**Probable Spam**). However, the probability of false positives in that case also becomes higher and normal mail can be recognized as spam.

Lower intensity degree decreases the probability of false positives but it increases the possibilities for spam to bypass the filter.

**Note**

Apart from the intensity level, filtering result is also affected by the methods used for spam recognition. When false positives occur you should consider the methods employed for spam recognition.

## 4.3.1. Analysis of formal signs

The method uses a set of rules based on examination of certain message headers and their comparison with sets of headers typical of spam messages. In addition to header analysis, the application takes into account message structure, size, presence of attachments and other similar signs.

The method also provides for analysis of data transmitted by the sender during an SMTP session. In particular, the following information is estimated:

- IP address of the server that has sent the message, and whether it is included into black list of recipients;
- IP addresses of intermediate relay servers obtained from the *Received* headers;
- e-mail addresses of message sender and recipients transmitted in SMTP session commands;
- presence of the sender's and recipients' addresses in white or black lists;
- conformity of the addresses transmitted during SMTP session to the set of addresses specified in message headers and a number of other checks.

## 4.3.2. Content filtration

Message analysis employs the algorithms of *content filtering*: the application uses artificial intelligence technologies to analyze the actual message content (including the *Subject* header), and its attachments (attached files) in the following formats:

- plain text (ASCII, not multibyte)
- HTML (2.0, 3.0, 3.2, 4.x, XHTML 1.0).

**Note**

The purpose of spam filtering is to decrease the volume of unwanted messages in the mailboxes of your users. It is impossible to guarantee detection of all spam messages, because too strict criteria would inevitably cause filtering of some normal messages as well.

The application uses three main groups of methods to detect spam messages:

- **Text comparison with semantic samples** of various categories (based on the search for key terms (words and word combinations) in message body and their subsequent probabilistic analysis). The method provides for heuristic search for typical phrases and expressions in text.
- **Fuzzy comparison of a message being examined with a collection of sample messages** based on comparison of their signatures. The method helps detect modified spam messages.
- **Analysis of attached images**.

All the data employed by Kaspersky Mail Gateway for content filtering: *classification index* (a hierarchical list of categories), message samples, typical terms, etc. are stored in the application databases.

**Note**

The group of spam analysts at Kaspersky Lab works nonstop to supplement and improve Kaspersky Mail Gateway databases. Therefore, you are advised to update the databases regularly.

You can also send to Kaspersky Lab samples of spam messages, which Kaspersky Mail Gateway has failed to recognize as well as the samples of messages erroneously classified as spam. The data will help us improve Kaspersky Mail Gateway databases and react in a timely manner to new types of spam. Please refer to Appendix C on page 191 for details on forwarding sample messages.

## 4.3.3. Checks using external services

In addition to the analysis of message text and headers, Kaspersky Mail Gateway allows a number of the following checks involving external network services:

- availability of a DNS record for message sender's IP (reverse DNS lookup);
- the presence of the sender's IP address in a DNS-based real time black hole list or lists (DNSBL);
- a check of the sender's address for compliance with SPF (Sender Policy Framework) policy for the domain containing the server used to send the message;
- a check of addresses and links to sites in message text for the presence in the Spam URL Realtime Blocklists database – www.surbl.org;
- recognition of e-mail messages using the UDS (Urgent Detection System) technology.

All the checks listed above, except for UDS, are based on the use of the DNS protocol and as a rule they require no additional network configuration.

## 4.3.4. Urgent Detection System

Urgent Detection System is an original technology of spam detection developed and supported by Kaspersky Lab. It is based on the following principles:

- A message being analyzed is used to select a collection of properties, which can be used to identify the message. The set of properties may include header information, text fragments and other information about the message being processed.

- Filtration server uses the properties thus collected to generate a small UDS request and sends it to one of UDS servers of Kaspersky Lab.

> **Note**
>
> Since the product does not transmit to external servers any data that could allow viewing the recipients or the text of the processed mail, the use of this method does not pose any risk to the safety or confidentiality of your information.

- The UDS server checks the received request against a database of known spam. If the request matches a known spam sample, a message will be sent to the filtration server informing that the e-mail is very likely to be spam. The information will be taken into account during assignment of a certain status to e-mail.

> **Note**
>
> The UDS technology allows filtering of known spam before updates to Kaspersky Mail Gateway databases become available.

A filtration server interacts with UDS servers of Kaspersky Lab via UDP using port 7060 for communication. In order to use UDS, a filtration server must be able to establish outgoing connections through that port.

Information about available UDS servers is added to Kaspersky Mail Gateway databases. The choice of an individual UDS to be used for message analysis is performed automatically on the basis of the response time of accessible UDS servers.

## 4.3.5. Recognition results and actions over messages

The analysis procedure results in assignment of one of the following statuses to a message:

- **Spam** – message recognized as spam.
- **Probable Spam** – message contains some spam signs; however, it cannot be unambiguously identified as spam.
- **Formal** (automatically generated letter) – message is formal, for example, it is a mail server notification informing about mail delivery or inability to deliver it or about message infection with a virus. The category includes messages sent automatically by mail clients. Such messages are usually not considered to be spam.

- **Blacklisted** – message received from an address present in a black list.
- **Not detected** – a message that has no sufficient spam signs to be recognized as spam. No actions are specified for messages with such status.

Messages that have received the **Not detected** status (the message has not been recognized as spam), are always transferred to the specified recipient. In that case the letter must also contain no infected or suspicious objects revealed during anti-virus scanning.

**Note**

Although the product is being constantly developed in order to improve spam recognition and decrease the number of false positives from the filter, it is not possible to eliminate altogether the probability of recognizing normal messages as spam. Therefore, you are advised to use with caution the actions deleting messages.

Each e-mail message can be assigned just one of the above statuses. The application records the status assigned to a message after analysis to a special **X-SpamTest-Status-Extended** header. Please refer to section B.18 on page 183 for details about the headers added to mail messages after filtering.

After recognition, the application may perform one of the following actions over a message:

- add a text mark in the message subject field;
- append special headers to the message;
- delete message.

System administrator can define which of the listed actions will be performed over messages with a specific status.

**Attention!**

Preservation of all useful mail must be the top priority task for the system administrator because the loss of a single important message may cause more trouble for the end user than receipt of a dozen of spam messages. To avoid the loss of necessary mail, you are advised to use only non-destructive actions with mail identified after content analysis as spam or probable spam.

In addition to actions related to mail routing, the administrator can specify the actions for message modification, which can be helpful both for visualization of recognition results and for use in combination with the filters in client e-mail software of end users:

- Add a label to the message subject field.

- Add to message special X-SpamTest-* headers. The headers can be used later for automatic mail processing by the e-mail software of end users. Please refer to section B.18 on page 183 for details about the headers added to mail messages after filtering.

## 4.4. Operation of the anti-virus scanning module

The AV module checks message components for the presence of viruses.

During the scanning and disinfection of detected infected objects the AV module uses the anti-virus databases, which contain descriptions of all currently known viruses and methods for disinfecting objects containing them.

**Attention!**

You are advised to update the anti-virus databases regularly, to maximize the efficiency of anti-virus functionality with respect to new viruses. Updates for the anti-virus databases are made available on Kaspersky Lab's update servers every hour.

By default, the application's AV module *only scans* your e-mail traffic; it does not cure infected objects.

To enable disinfection, set the **AVCure** parameter in the **[mailgw.group: group_name]** section of the configuration file to **true**. If disinfection has been successful, the object is assigned **Disinfected** status.

An object may be assigned one of the following statuses in the process of its scanning:

- **Clean** – object is clean.
- **Infected** – object is infected and cannot be disinfected or its disinfection has not been attempted.
- **Disinfected** – infected object has been successfully disinfected.

**Note**

An object can be assigned the **Disinfected** status only if the cure mode has been enabled for infected objects.

- **Suspicious** – object is suspected of being infected with an unknown virus or with a new modification of a known virus.

- **Protected** – scanning failed because the object is password-protected (e.g., it is an archive).
- **Error** – object is an error occurred during the scan.
- **Not_checked** – object has not been scanned because anti-virus checks have been disabled.

The actions performed by the AV module on an object which has passed scanning are determined by the corresponding options in the configuration file (**ActionInfected**, **ActionSuspicious**, etc.). Each message status has a corresponding option. The following actions are available:

- **cure** – replace the infected object in a message with a disinfected one;

**Attention!**

The action can only be defined for objects with **Disinfected** status (**ActionDisinfected** parameter).

- **pass** – transfer the object without modifications, no actions will be applied to the object;
- **remove** – remove the object from the e-mail message;
- **placeholder** – replace the object with a notification generated from a template.

# CHAPTER 5. ANTI-VIRUS PROTECTION AND SPAM FILTRATION

Kaspersky Mail Gateway can provide anti-virus protection and spam filtering for e-mail traffic transferred through your organization's mail server.

The tasks implemented by Kaspersky Mail Gateway may be divided into three major groups:

1. Updates of the anti-spam and anti-virus databases used for spam filtering, anti-virus scanning and disinfection of objects.
2. Spam filtering.
3. Anti-virus protection of e-mail traffic.

Each of these groups comprises more specific tasks. In this chapter, we will discuss some typical tasks that the administrator can combine and enhance in accordance with the needs of his/her organization.

**Attention!**

To perform the tasks described, some changes must be made to the application's configuration file, following which the application must be restarted to apply the modifications.

This guide describes how to locally configure and start tasks from the command line. Issues related to starting and managing tasks from remote computers using the Webmin application are not discussed in this document.

**Attention!**

In the examples below, it is assumed that the administrator has completed all required post-installation tasks and the application operates correctly.

## 5.1. Updating the anti-virus and anti-spam databases

Kaspersky Mail Gateway uses the anti-virus and anti-spam databases while processing e-mail traffic.

The anti-spam database is employed for spam filtering, which requires the analysis of the contents of message bodies and attached files to identify unsolicited e-mail.

The anti-virus databases are employed during scanning and disinfection of infected objects; they contain descriptions of all currently known viruses and the methods of disinfection for objects affected by those viruses.

The *keepup2date* component is included in Kaspersky Mail Gateway to provide for software updates. The updates are retrieved from Kaspersky Lab's update servers, e.g.:

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

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

ftp://downloads1.kaspersky-labs.com/ etc.

The *updcfg.xml* file included in the installation package lists the URLs of all available update servers.

**Note**

The *keepup2date* component supports Basic authentication for connections through a proxy server.

To update the anti-virus and content filtration databases, the *keepup2date* component selects an address from the list of update servers and tries to download updates from that server. If the server is currently unavailable, the application connects to another server on the list, until it succeeds.

**Note**

Updates for the anti-spam databases are made available on Kaspersky Lab's update servers every three minutes. Updates for the anti-virus databases of Kaspersky Mail Gateway are made available on Kaspersky Lab's update servers every hour.

After connection to an update server, *keepup2date* identifies available updates and downloads them.

**Attention!**

We strongly recommend that the *keepup2date* component is configured to update the databases every three minutes!

After a successful update, the command specified by the value of the **PostUpdateCmd** parameter in the **[updater.options]** section of the configuration file will be executed. By default, this command starts compilation of the anti-spam module databases and automatically restarts the application. The restart is necessary to make the application use the updated anti-spam databases. Kaspersky Mail

Gateway anti-virus databases are loaded without restart. Incorrect modification of this parameter may prevent the application from using the updated databases or cause it to function erroneously.

**Note**

All settings of the *keepup2date* component are stored in the **[updater.*]** sections of the configuration file.

If you have purchased a license for Kaspersky Mail Gateway to provide only anti-virus scanning of e-mail traffic, downloading of updates for the anti-spam databases can be disabled. To do so, assign the values **AVS, AVS_OLD, CORE, Updater**, and **BLST** to the **UpdateComponentsList** parameter in the **[updater.options]** section:

```
[updater.options]
UpdateComponentsList=AVS, AVS_OLD, CORE, Updater,
BLST
```

If your network has a complicated structure, you are advised to download updates from Kaspersky Lab's update servers every three minutes and place them in a network directory. Other networked computers can be configured to copy their updates from that directory. For detailed instructions on how to implement this scenario, see section 5.1.3 on p. 50.

The updating process can either be scheduled to run automatically using the **cron** utility (see section 5.1.1 on p. 48), or started manually from the command line (see section 5.1.2 on p. 49). Starting the *keepup2date* component requires **root** user privileges.

## 5.1.1. Automatic updating of the anti-virus and anti-spam databases

Regular automatic updates for the anti-virus and anti-spam databases can be scheduled using the **cron** utility.

Example:

> Configure the **cron** utility to update automatically your anti-virus and anti-spam databases every three minutes. An update server should be selected from the *updcfg.xml* file by default. Only errors occurring in the component operation should be recorded in the system log. Keep a general log of all task starts. Output no information to the console.

To perform the above task, do the following:

1. In the application's configuration file, specify the following values for these parameters:

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

2. Edit the cron task file for the **root** user by typing this command: **crontab -u root -e** and add the following line:

   In Linux:

   ```
   */3 * * * * /opt/kaspersky/mailgw/bin/mailgw-
   keepup2date
   ```

   In FreeBSD:

   ```
   */3 * * * * /usr/local/bin/mailgw-keepup2date
   ```

## 5.1.2. Manual updating of the anti-virus and anti-spam databases

You can start updating your anti-virus and anti-spam databases from the command line at any time.

Example:

> start updating of the anti-virus and anti-spam databases, save the results of updating in the */tmp/updatesreport.log* file.

To accomplish the task, log in as **root** (or any other privileged user) and enter at the command line:

```
# mailgw-keepup2date -l /tmp/updatesreport.log
```

If you need to update the anti-virus and anti-spam databases on several servers, it may be more convenient to download the updates from an update server once, save them to a shared directory, and mount the directory within the file system of every server running Kaspersky Mail Gateway. Then it will be sufficient to launch the update script, having first specified the mounted directory as the source of updates. Please see section 5.1.3 on p. 50 for details of how to create a shared directory for updates.

Example:

> start updating the anti-virus and anti-spam databases from the local directory */home/kluser/bases*. If the directory is inaccessible or empty, update the databases from Kaspersky Lab's update servers. Save the results to the */tmp/updatesreport.log* file.

To accomplish the task, log in as **root** (or any other privileged user) and do the following:

1. Mount the shared directory containing the anti-virus database updates as the local directory */home/kluser/bases*.
2. In the application configuration file, specify the following values for these parameters:

   ```
   [updater.options]
   UpdateServerUrl=/home/kluser/bases
   UseUpdateServerUrl=true
   UseUpdateServerUrlOnly=false
   ```

3. Enter the following at the command line:

   ```
   # mailgw-keepup2date -l /tmp/updatesreport.log
   ```

**Attention!**

These and other similar tasks can be accomplished remotely using the Webmin remote administration module.

## 5.1.3. Creating a network directory to store and share updates

Kaspersky Mail Gateway supports copying of updates to databases and application modules into a network directory for sharing and storage. That directory can be specified as the source of updates for the Kaspersky Mail Gateway 5.6 installations on network computers as well as other applications of Kaspersky Lab (versions 6.0 and 7.0).

To ensure that local computers are correctly updated from the shared directory, the directory must have the same file structure as Kaspersky Lab's update servers. This task deserves a detailed explanation.

**Note**

Please keep in mind that for Kaspersky Mail Gateway 5.6 only anti-virus and anti-spam databases will be updated.

Example:

> create a shared local directory which local computers will use as the source to update their anti-virus and anti-spam databases.

To accomplish the task, log in as **root** (or any other privileged user) and do the following:

1. Create a local directory.
2. Define the following parameter values in the application configuration file:

```
[updater.options]
RetranslateComponentsList =KAS303, AVS, AVS_OLD,
CORE, Updater, BLST
```

3. Run the *keepup2date* component as follows:

```
# mailgw-keepup2date -u <rdir>
```

where `<rdir>` is the full path to the directory created.

> **Note**
>
> If other applications (versions 6.0 and 7.0) of Kaspersky Lab will be updated from the shared directory, the *keepup2date* component must be started as follows:
>
> ```
> #mailgw-keepup2date -x <rdir>
> ```

4. Grant read-only access to the directory for local computers on your network.

# 5.2. Spam filtration

This section contains sample tasks demonstrating the application's functionality related to spam filtering. The examples show the main mechanisms used by the application to combat spam, and in particular:

- spam filtration and organization of user groups;
- marking of messages identified as spam, probable spam, formal or blacklisted mail with special labels in the *Subject* header;

> **Note**
>
> Users may set up their e-mail clients to transfer labeled messages to corresponding directories.

- blocking of delivery for messages identified as spam, probable spam, formal or blacklisted mail;

- saving of messages identified as spam, probable spam, formal or black-listed mail in the quarantine directory.

The section also includes information about the procedure used by the anti-spam module components and about the parameters controlling the anti-spam module.

## 5.2.1. Starting and managing the components of the anti-spam module

The main components of the anti-spam filtration server including:

- the filtering master process (*mailgw-process-server*)
- licensing daemon (*mailgw-kas-license*)
- the SPF daemon (*mailgw-spfd*)

are launched at the operating system start-up by a special script, which is named and located differently in Linux and FreeBSD operating systems. The Linux operating system uses the *mailgw* script located in the */opt/kaspersky/mailgw/lib/bin/* directory (the */etc/init.d/mailgw* link can be used, too), while the FreeBSD operating system employs the *mailgw.sh* script in the */usr/local/etc/rc.d/* directory*.*

The administrator can use the said scripts with the command line parameters described below to start, stop or restart the main components of the filtration server:

- **start** – start the main components of the filtration server.
- **stop** – stop operation of the main components of the filtration server.
- **restart** – restart the main components of the filtration server; the action is identical to running the **stop** and **start** actions one after another.

## 5.2.2. Managing the filtration process

The main purpose of the anti-spam module is detection of unwanted messages in e-mail stream. The module has an advanced system of settings for configuration of spam recognition and its further processing:

- The level of spam recognition intensity (**SpamRateLimit** parameter in the **[mailgw.policy]** section). The application decides whether a message contains spam on the basis of several signs revealed in it by the scanning module (please refer to section 4.3 on page 38 for details).

- Addition of **ProbableSpam** or **Obscene** marks to the header of messages recognized as mail belonging to the corresponding category after checks (**SpamMarkProbable** and **SpamMarkObscene** parameters respectively).
- Verification of information about message sender in DNS and DNS-based services: DNSBL, SPF, etc (**SpamUseDNS** parameter).
- Checks of the sender IP address using a set of DNSBL services (**SpamCheckDNSBL** parameter).
- Check of sender IP presence in DNS (**SpamCheckHostInDNS** parameter).
- Check of sender IP using SPF (Sender Policy Framework) (**SpamCheckSPF** parameter).
- Check of sender IP address presence using SURBL (Spam URL Real-time Blocklists) (**SpamCheckSURBL** parameter).
- Analysis of message headers checking them for:
  - List of undisclosed recipients in message headers (**SpamHeadersToUndisclosed** parameter).
  - Groups of digits in the sender's or recipient's address (**SpamHeadersFromOrToDigits** parameter).
  - Missing domain part in address (**SpamHeadersFromOrToNoDomain** parameter).
  - Long text in message subject (**SpamHeadersSubjectTooLong** parameter).
  - Multiple spaces and dots in message subject (**SpamHeadersSubjectWSOrDots** parameter).
  - Digital identifier or time label in message subject (**SpamHeadersSubjectDigitIDOrTimestamp** parameter).
  - Text in Chinese, Korean, Thai or Japanese in message headers (**SpamHeadersMarkAllChinese**, **SpamHeadersMarkAllKorean**, **SpamHeadersMarkAllThai**, **SpamHeadersMarkAllJapanese** parameters).
- Addition to message header of a prefix describing its status assigned by the anti-spam module after scanning (**MarkSubject** parameter).
- Maximum size (Kb) of messages scanned for spam presence (**SpamCheckSizeLimit** parameter**)**.

- Definition of individual groups of senders/recipients whose mail will be handled using custom rules (the **[mailgw.group:group_name]** section is used), including filtration based on black or white lists (please refer to section 5.2.3 on page 54 for details).

## 5.2.3. Mail filtration using black and white lists

White list of senders is used to specify explicitly the addresses that provide mail which should not be scanned for presence of spam signs. The list can include, for example, IP addresses of e-mail servers used to relay mail in corporate LAN or the addresses of internal mailing lists.

During application configuration, white lists are created using specifically defined groups for which anti-spam and/or anti-virus scanning is disabled (**CheckSpam=false**, **CheckAV=false**).

Black list of senders has the opposite meaning. Administrators of the filtration server can add to the list addresses which spammers use to distribute their mail and computers spreading viruses.

Black lists are implemented through definition of an appropriate set of **ConnectRule** rules with specified **deny** action.

Example:

Task:

- Create a group of senders whose mail will be treated as belonging to a white list. Criterion including senders in white list: any host of the 10.10.0.0/16 subnet.
- Create a group of senders whose mail will be treated as belonging to a black list. Criterion including senders in black list: host with the 10.10.138.99 address.
- Messages from the white list should not be scanned for presence of spam or viruses; they should be forwarded to recipients unchanged.
- Messages from the black list should not be accepted.

To accomplish the task, perform the following steps:

- Define the level of spam filtration intensity setting the corresponding parameter in the **[mailgw.policy]** section of the configuration file to the following value:

```
SpamRateLimit=standard
```

2. In the **[mailgw.access]** section specify a **ConnectRule** to reject a session when connection is established with the 10.10.138.99 address:

   ```
   [mailgw.access]
   ...
   ConnectRule=deny for ip 10.10.138.99
   ...
   ```

3. Create the **[mailgw.group:whitelist]** section which defines the following rules for processing of mail for the users included into the **whitelist** group:

   ```
   [mailgw.group:whitelist]
   Recipients=*
   Senders=ip 10.10.0.0/16
   CheckSpam=false
   CheckAV=false
   ```

## 5.2.4. Managing the UDS service

### Checking the access time of UDS servers

The application uses the *uds-rtts.sh* script to check the time required for access to the UDS servers of Kaspersky Lab. Collected data is used to select the most suitable server for UDS queries.

Script launch command in Linux:

```
# /opt/kaspersky/mailgw/lib/bin/kas-filter/uds-rtts.sh -q
```

In FreeBSD:

```
# /usr/local/libexec/kaspersky/mailgw/kas-filter/\
uds-rtts.sh -q
```

To increase the efficiency of the UDS server you should configure the task checking the UDS server access time to run regularly, for example, using **cron**.

The recommended interval between task starts is every 10-15 minutes.

### Checking UDS server availability

To check if a UDS server is available (i.e. it can be accessed), run the *uds-rtts.sh* script with the *-a* option as follows*:*

In Linux:

```
# /opt/kaspersky/mailgw/lib/bin/kas-filter/uds-rtts.sh -a
```

In FreeBSD:

```
# /usr/local/libexec/kaspersky/mailgw/kas-filter/\
uds-rtts.sh -a
Restarting as kluser
uds-rtts: OK, updated 1 records.
uds-rtts: uds.kaspersky-labs.com available rtt=4103
uds-rtts finished successfully.
```

## 5.2.5. Managing the list of enabled DNSBL services

Checks of sender IP address presence in DNSBL are performed on two levels:

- When an incoming connection is established (provided that an appropriate rule is specified in the **ConnectRule** parameter), please see section Appendix A on page 107.
- When the anti-spam module checks a message (the check also includes verification of IP addresses mentioned in the *Received* header of the message). You can define for each group of users whether the application will run checks involving DNSBL services for that group.

Management of the DNSBL services used by the application belongs to general settings of the anti-spam module. The list of available services is common for all user groups.

Each DNSBL service is defined through its address where queries are sent and its corresponding rating.

Service rating determines how trustworthy the service is in the opinion of the administrator. When a sender's IP address is checked in DNSBL, Kaspersky Mail Gateway sends a query to all the services included in the list. When the results are returned, it sums up the rating values of the services, which have recognized the specified IP address as a source of spam mail.

When IP address presence in DNSBL is checked at connection establishment, the **in_dnsbl** rule (see section Appendix A on page 107) will be applied if the sum of ratings of the triggered DNSBL services reaches 100 or exceeds the value.

If IP address presence in DNSBL is checked by the anti-spam module, message sender is assumed to be included in the black list and the letter receives blacklisted status if the sum of ratings of triggered DNSBL services reaches 100 or exceeds the value. The status is assigned irrespectively of the results returned by checks using other methods.

If the sum of ratings of the triggered DNSBL services exceeds 100, the sender will be assumed to be included into black list and the corresponding message will receive the **blacklisted** status independently from the results returned by other checks based on various methods. Some filtering intensity levels also allow situations when the sum of ratings for services which contain the sender in their black lists is less than 100. In that case information about sender presence in black lists is used as an additional sign only and such mail is recognized as spam if there are more signs revealed by other checks.

## 5.2.6. Marking of messages containing spam

Example:

- Filter spam using the **standard** degree of filtering intensity.
- Modify the *Subject* header of messages identified as spam or probable spam.

To perform the above task, do the following:

Specify the level of spam filtering intensity, by setting the **SpamRateLimit** parameter value in the **[mailgw.policy]** section of the configuration file. Then define the mail processing rules:

```
SpamRateLimit=standard
CheckSpam=true
MarkSubject=spam,probable
```

## 5.2.7. Blocking delivery of spam messages

Example:

- Filter spam; specify the **standard** degree of filtering intensity.
- Block the delivery of messages identified as spam or probable spam, for users in the **managers** group.
- Block the delivery of spam messages only, for all other users.

To perform the above task, do the following:

1. Specify the level of filtering intensity. To do so, specify the following parameter value in the **[mailgw.policy]** section of the configuration file:

```
SpamRateLimit=standard
```

2. Create the **[mailgw.group:managers]** section, which will define the rules for processing the e-mail of users included in the **managers** group:

   ```
   [mailgw.group:managers]
   Recipients=*@managers.example.com
   CheckSpam=true
   BlockMessage=as/spam,as/probable
   ```

   Mail processing rules for all other users will also be defined by the **[mailgw.policy]** section:

   ```
   [mailgw.policy]
   CheckSpam=true
   BlockMessage=as/spam
   ```

## 5.2.8. Storage of spam message copies in the quarantine directory

Storing message copies in the quarantine directory can be combined with blocking e-mail delivery, but not necessarily. In the first case messages identified as spam or probable spam will not reach the mailboxes of recipients, but are saved in the quarantine directory. In the second case, the messages will be delivered to end users **and** message copies will be preserved in quarantine.

Example:

- Filter spam; specify the **standard** degree of filtering intensity.
- Copy all messages identified as spam, probable spam, formal or blacklisted mail to the quarantine directory.
- Block the delivery of messages identified as spam or probable spam.

To perform the above task, do the following:

1. Specify the level of filtering intensity, by setting the following parameter value in the **[mailgw.policy]** section of the configuration file:

   ```
   SpamRateLimit=standard
   ```

2. Specify the following parameter values in the **[mailgw.policy]** section of the configuration file:

   ```
   [mailgw.policy]
   CheckSpam=true
   BlockMessage=as/spam,as/probable
   ```

```
QuarantineMessage=as/spam,as/probable,as/formal,
as/blacklisted
```

**Attention!**

Blocked and quarantined messages that have been assigned the status **Spam**, **Probable Spam, Formal** or **Blacklisted** by the anti-spam module may contain viruses, as their anti-virus scanning will be skipped after performance of these actions.

# 5.3. Anti-virus protection of e-mail traffic

This section contains examples of Kaspersky Mail Gateway's anti-virus protection of e-mail traffic. The settings described in the examples can be combined to produce more sophisticated e-mail traffic protection schemes.

## 5.3.1. Delivery of messages with clean or disinfected objects only

Example:

- Scan all the server's incoming and outgoing e-mail traffic for viruses.
- Cure infected objects.
- Remove from e-mail messages all infected objects which could not be cured.
- Deliver messages to recipients containing clean and disinfected objects only.

To perform the above task, specify the following parameter values in the **[mailgw.policy]** section:

1. Define the anti-virus scanning mode for all e-mail messages:

   ```
   CheckAV=true
   ```

2. Enable disinfection mode for infected objects:

   ```
   AVCure=true
   ```

3. Specify the operations, which must be performed with the objects:

   ```
   ActionDisinfected=cure
   ActionInfected=remove
   ```

```
ActionSuspicious=remove
ActionProtected=remove
ActionError=remove
BlockMessage=
```

**Note**

Notifications can be delivered to the administrator, message recipient and sender, informing them of the detection of infected or suspicious objects (see section 5.3.4 on p. 62). Also, messages containing infected, suspicious or password-protected objects can be saved in the quarantine directory (see section 5.3.6 on p. 64).

## 5.3.2. Replacement of infected objects by standard notifications

Task:

- Scan all e-mail traffic on the server for viruses, and cure infected objects in e-mail messages.
- Objects which cannot be cured, and suspicious, damaged or password-protected objects, must be deleted and replaced with a standard notification.

Solution: To perform the above task, specify the following parameter values in the **[mailgw.policy]** section:

1. Define the anti-virus scanning mode for all e-mail messages:

   ```
   CheckAV=true
   ```

2. Enable disinfection mode for infected objects:

   ```
   AVCure=true
   ```

3. Specify the operations, which must be performed with the objects:

   ```
   ActionDisinfected=cure
   ActionInfected=placeholder
   ActionSuspicious=placeholder
   ActionProtected=placeholder
   ActionError=placeholder
   BlockMessage=
   ```

**Note**

In addition to replacing infected and suspicious objects with standard sages, the application can deliver notifications to the administrator with information about the detection of the objects (see section 5.3.4 on p. 62) and save the messages containing the objects in the quarantine directory (see section 5.3.6 on p. 64).

## 5.3.3. Blocking delivery for messages containing suspicious objects

Example:

- Scan all e-mail traffic on the server for viruses, and cure infected objects in e-mail messages;
- Block the delivery of messages containing objects which cannot be cured, and suspicious, damaged or password-protected objects.

**Attention!**

While implementing this task, please note that if a message contains several objects, one of which cannot be disinfected or is suspicious or password

protected, the delivery of the whole message will be blocked.

To perform the above task, specify the following parameter values in the **[mailgw.policy]** section:

1. Define the anti-virus scanning mode for all e-mail messages:

```
CheckAV=true
```

2. Enable disinfection mode for infected objects:

```
AVCure=true
```

3. Specify the operations, which must be performed with the objects:

```
ActionDisinfected=cure
ActionInfected=pass
ActionSuspicious=pass
ActionProtected=pass
ActionError=pass
BlockMessage=av/infected,av/suspicious,
av/protected,av/error
```

**Note**

The application can also be configured to send notifications to the administrator with information about the detection of infected or suspicious objects (see tion 5.3.4 on p. 62) and save the messages containing those objects in the quarantine directory for later delivery to Kaspersky Lab for examination (see section 5.3.6 on p. 64).

## 5.3.4. Delivery of notifications to the sender, administrator and recipients

Example:

- Scan all e-mail traffic on the server for viruses, and cure all infected objects.
- Deliver messages to recipients containing only clean and disinfected objects.
- Delete all objects which cannot be cured, as well as suspicious, damaged or password-protected objects.
- Notify the senders, recipients and the administrator about cured, incurable, deleted and suspicious and damaged objects in e-mail messages.

To perform the above task, specify the following parameter values in the **[mailgw.policy]** section:

1. Enable disinfection mode for infected objects:

   ```
   AVCure=true
   ```

2. Specify the operations, which must be performed with the objects:

   ```
   ActionDisinfected=cure
   ActionInfected=remove
   ActionSuspicious=remove
   ActionProtected=remove
   ActionError=remove
   BlockMessage=
   ```

3. Specify the cases in which notifications should be sent, and their recipients:

   ```
   NotifyAdmin=av/disinfected,av/infected,
   av/suspicious,av/protected,av/error
   ```

```
NotifyRecipient=av/disinfected,av/infected,
av/suspicious,av/protected,av/error
NotifySender=av/disinfected,av/infected,
av/suspicious,av/protected,av/error
```

### 5.3.5. Additional filtering of objects by name and type

E-mail messages frequently contain objects for which virus infection is highly probable (e.g., executable files). To avoid infection, you are advised to configure the application to filter e-mail by name and/or attachment types, and save these objects in a separate directory.

There are also objects which cannot be infected with viruses (e.g., plain text files). To reduce the load on the server during anti-virus scanning of e-mail messages, you are advised to specify the types and/or the names of such attachments in advance so that the application does not scan them.

Filtering of objects is performed using name masks (**IncludeByName**, **ExcludeByName** parameters) and MIME types (**IncludeByMime**, **ExcludeByMime** parameters).

Example:

- Delete *.exe* and *.reg* attachments from the e-mail of users in the **managers** group.
- For users in the **accounts** group, delete all attached objects except for *.doc* files .
- For users in the **sales** group, block messages containing attached *.exe* files.

To perform the above task, do the following:

Create in the application's configuration file three **[mailgw.group:group_name]** sections, which will contain processing rules for the e-mail of users in the **managers**, **accounts** and **sales** groups respectively:

```
[mailgw.group:managers]
Recipients=*@managers.example.com
IncludeByName=*.exe
IncludeByName=*.reg
ActionFiltered=remove
...
```

```
[mailgw.group:accounts]
Recipients=*@accounts.example.com
ExcludeByName=*.doc
ActionFiltered=remove
…
[mailgw.group:sales]
Recipients=*@sales.example.com
IncludeByName=*.exe
BlockMessage=av/filtered
```

## 5.3.6. Saving messages in the quarantine directory

Kaspersky Mail Gateway can be configured to store messages with specified statuses in the quarantine directory.

This feature may be used, for example, if an infected attachment containing important data was detected during anti-virus scanning. Attempting to disinfect the file may corrupt the data. The message can be isolated in a separate directory and subsequently sent to Kaspersky Lab for analysis. Our experts will probably be able to disinfect the file, and preserve the data's integrity.

Example:

- Scan all e-mail traffic on the server for viruses and cure all infected objects.
- Deliver messages to the recipients containing only clean and disinfected objects.
- Messages with incurable attachments or suspicious, damaged or password-protected objects must be saved in the quarantine directory */opt/quarantine*; delivery of these messages must be blocked.

To perform the above task, do the following:

1. Create the directory */opt/quarantine*, which will be used to store blocked messages, and grant the right to write to that directory to the account used to run the application (**kluser** by default).
2. Enable the cure mode for infected objects, by setting the following parameter value in the **[mailgw.policy]** section of the configuration file:

   ```
   AVCure=true
   ```

3. Specify these parameter values in the **[mailgw.policy]** section of the configuration file:

```
ActionDisinfected=cure
ActionInfected=pass
ActionSuspicious=pass
ActionProtected=pass
ActionError=pass
BlockMessage=
av/infected,av/suspicious,av/protected,av/error
QuarantineMessage=av/infected,av/suspicious,
av/protected,av/error
AVQuarantinePath=/opt/quarantine
```

## 5.4. Combining spam filtration and anti-virus protection

The choices of application mode, of level of anti-virus scanning and of spam filtering intensity depend both on the volume of e-mail traffic to be processed by the application, and the corporate security policy. Three modes demonstrated in this section illustrate methods for combining spam filtration with anti-virus protection of e-mail traffic.

> **Note**
>
> The application settings described in this section are provided as examples only; the administrator should adapt them as necessary.

### 5.4.1. Maximum speed

The mode allows high performance anti-virus scanning and spam filtration, which may be necessary for processing a large volume of e-mail messages. The security level in this case is reduced, because the application does not cure infected objects, but just sends notifications about their detection.

In this mode, the application:

- filters e-mail traffic for spam; the degree of filtering intensity is **minimum**;
- blocks messages identified as spam;

- marks messages identified as probable spam, formal or blacklisted mail using special labels in the *Subject* header;
- performs anti-virus scanning of e-mail attachments, but does not attempt to cure infected objects;
- filters and blocks delivery of messages containing the most dangerous attachment types (an external file is used to define the list of dangerous objects) and for messages containing infected attachments;
- notifies recipients about messages which have been blocked.

*To enable this mode:*

1. Specify the following parameter value in the **[mailgw.policy]** section of the configuration file:

   ```
   SpamRateLimit=minimum
   ```

2. Create a file *List1* which contains a list of the most likely sources of viruses, for example:

   ```
       *.exe
       *.bat
       *.com
       *.scr
       *.bin
       *.dll
   ```

3. Specify the following parameter values in the **[mailgw.policy]** section of the configuration file:

   ```
   AVCure=false
   AVScanArchives=false
   AVScanMailBases=false
   CheckAV=true
   CheckSpam=true
   IncludeByName=file:<path to file>/List1
   MarkSubject=probable,formal,blacklisted
   ActionFiltered=pass
   ActionInfected=pass
   ActionSuspicious=pass
   ActionProtected=pass
   ActionError=pass
   BlockMessage=av/infected,av/filtered,as/spam
   ```

```
NotifyRecipient=av/infected,av/filtered
```

> **Note**
>
> The presence of several groups of senders/recipients (**[mailgw.group:group_name]** sections) slows down processing of e-mail traffic. When high performance is required, you are advised to use the default group only (**[mailgw.policy]** section) to specify the e-mail processing rules.

## 5.4.2. Recommended mode

The mode gives the optimal balance between server performance and security. In this mode, the application:

- filters e-mail traffic looking for spam; the degree of filtering intensity is **standard**;
- marks messages identified as spam, probable spam, formal or blacklisted mail using special labels in the *Subject* header;
- performs anti-virus scanning and disinfection of e-mail attachments;
- replaces suspicious objects, and infected objects which cannot be cured, with a standard notification;
- blocks delivery for messages containing password-protected attachments and attached objects that cause errors during scanning; these attachments are added to the quarantine directory;
- notifies recipients about blocked messages.

*To enable this mode:*

1. Specify the following parameter value in the **[mailgw.policy]** section of the application’s configuration file:

   ```
   SpamDetection=standard
   ```

2. Specify the following parameter values in the **[mailgw.policy]** section:

   ```
   AVCure=true
   AVScanArchives=true
   AVScanMailBases=true
   CheckAV=true
   CheckSpam=true
   MarkSubject=spam,probable,formal,blacklisted
   ActionDisinfected=cure
   ```

```
ActionInfected=placeholder
ActionSuspicious=placeholder
ActionProtected=pass
ActionError=pass
BlockMessage=av/protected,av/error
QuarantineMessage=av/protected,av/error
NotifyRecipient=av/protected,av/error
```

## 5.4.3. Maximum protection

In the maximum protection mode the speed of e-mail traffic processing is lower. However, the mode provides the best protection for users against spam and viruses. In this mode the application:

- filters e-mail traffic looking for spam; the degree of filtering intensity is **maximum**;
- blocks delivery for messages identified as spam, probable spam, formal or blacklisted mail and adds them to the quarantine directory;
- performs anti-virus scanning and disinfection of e-mail attachments;
- removes the following from messages: infected attachments which cannot be cured; suspicious or password-protected objects, and objects which caused errors during scanning;
- notifies message recipients and the administrator about infected, suspicious and password-protected attachments, and objects which caused errors during scanning.

*To enable that mode:*

1. Specify the following parameter value in the **[mailgw.policy]** section of the configuration file:

   ```
   SpamRateLimit=maximum
   ```

2. Specify the following parameter values in the **[mailgw.policy]** section of the configuration file:

   ```
   AVCure=true
   AVScanArchives=true
   AVScanMailBases=true
   CheckAV=true
   CheckSpam=true
   MarkSubject=spam,probable,formal,blacklisted
   ```

```
ActionDisinfected=cure
ActionInfected=remove
ActionSuspicious=remove
ActionProtected=remove
ActionError=remove
BlockMessage=as/all
QuarantineMessage=as/all
NotifyRecipient=av/infected,av/suspicious,
av/protected,av/error
NotifyAdmin=av/infected,av/suspicious,
av/protected,av/error
```

# 5.5. Additional features of Kaspersky Mail Gateway

In addition to its main functions, of spam filtering and anti-virus scanning of e-mail traffic, the application can also perform these tasks:

- logging of received and sent e-mail;
- forwarding of all received e-mail;
- enabling restrictions for SMTP connections, preventing both hacker attacks and the use of the application as an open relay for sending unauthorized e-mail.

## 5.5.1. Automatically add incoming and outgoing e-mail to archives

If the security policy of your organization includes archiving e-mail traffic processed by the server, the application can be configured to add all e-mail messages to archives. If necessary, the administrator can view all messages in archives.

If the auto archiving option is enabled, copies of the following messages will be archived:

- All incoming messages including spam or infected objects, without additionally notifying the administrator. Archiving these messages is enabled when the path to the archive directory is specified as the value of the **IncomingArchivePath** parameter in the **[mailgw.archive]** section).

- All outgoing messages, including messages delivered to recipients, messages blocked because of a virus or spam, and notification messages generated by the application. Archiving these messages is enabled when the path to the archive directory is specified as the value of the **OutgoingArchivePath** parameter in section **[mailgw.archive]**).

- All received messages before their scanning. The application starts adding mail to archive if you specify a list of e-mail addresses (address) where blind carbon copies of the mail will be sent (**IncomingBcc** option in the **[mailgw.archive]** section).

**Attention!**

Before you enable automatic archiving, make sure that there is enough space in your server's file system to accommodate the archive.

Do not forget to purge this directory occasionally to remove old messages, and to compress necessary files (the frequency at which this is required depends on the intensity of e-mail traffic in your network).

## 5.5.2. Protection from hacker attacks and spam

To provide the highest level of security for your e-mail system, you are advised to modify the configuration file to extend the application's anti-virus functionality. To protect your server from hacker attacks or, for example, to prevent spam being relayed through your server, configure the following options:

- **ConnectRule** in the **[mailgw.access]** section. The parameter defines application behaviour during establishment of an SMTP session.

- **HeloRule** in the **[mailgw.access]** section. The parameter defines the application response to HELO/EHLO commands received from a client.

- **MailfromRule** in the **[mailgw.access]** section. The parameter defines the application's behaviour in response to an attempt to send a message from a source (passed with the MAIL FROM command) with a domain name which does not match the actual IP address or MX host corresponding to that domain.

- **RelayRule** in the **[mailgw.access]** section. The parameter defines rules for client access to the gateway. The correct settings of this option are essential to prevent the application's use as a publicly open e-mail relay.

**Attention!**

A detailed discussion of the syntax of these parameters is provided in the description of the configuration file (see Appendix A on p. 107).

You are also advised to enable restrictions for SMTP connections (see section 6.1.2 on p. 78).

Application version 5.6 supports the technology of DNS black lists. This technology allows the blocking of incoming e-mail sent from unsafe servers registered in the DNSBL database as servers sending spam. The list of DNSBL services is specified in the **DNSBlackList** parameter, in the **[mailgw.access]** section of the application configuration file.

**Attention!**

**DNSBL** service **(DNS-based Blackhole List**) is a database that lists IP addresses of mail servers used for uncontrolled mass mailing. Such servers receive mail from anyone and deliver it further to arbitrary recipients. Use of DNSBL allows automatic blocking of mail from such mail servers. Various services use different policies for generation of such lists. Please examine carefully the policy of each service before you start using it for mail filtration.

If a certain address is constantly used for sending spam and the administration of the server used for spam distribution takes no preventive steps, you can inform RBL about the spammer. The latter will be added to the database and the record will allow automatic blocking of incoming e-mail sent from that mail server.

# 5.6. Managing product keys

The right to use Kaspersky Mail Gateway is determined by the product *key*. The key is included in the application's distribution kit and entitles you to use the application from the day on which you purchased it and installed the key.

**Attention!**

Kaspersky Mail Gateway WILL NOT work without a key!

After the key expires, the application will continue to work as before, except that the anti-virus and anti-spam databases will no longer be updated. That is, the application will still be able to scan e-mail messages for viruses, filter spam and disinfect infected objects, but will be unable to use databases issued after the key expiration date. Therefore, you may not be protected against new viruses that appear after the license expired, and the anti-spam module will be unable to filter new spam types.

To protect your network's computers against new viruses and efficiently filter spam, you are advised to renew the key for Kaspersky Mail Gateway.

The key gives you the right to use the application. It contains information related to the license you have purchased, including the type of license, the key expiry date, and information about dealers.

In addition to the right to use the application during the period of key validity, you will have the following benefits:

- twenty-four-hour technical support;
- hourly updates of the anti-virus databases, and updates to the anti-spam database made available every three minutes;
- timely notifications about new virus threats.

For all these reasons, it is essential to extend your product key before it expires. One way to manage licenses is to install an additional key, which the application will start to use as soon as the current active key expires (see section 5.6.2 on p. 74).

## 5.6.1. Viewing information about product keys

You can view information about installed product keys in the reports of the *mailgw* component. Each time the main application component starts it loads the license key and displays its contents in the report.

More detailed information about the status of license keys may be obtained using *licensemanager*, a special component of the application.

All information about keys may be viewed either on the server's console, or remotely from any networked computer that has access to the Webmin module.

*To view information about all installed product keys, enter the following in the command line:*

```
In Linux:
# /opt/kaspersky/mailgw/bin/mailgw-licensemanager -s
in FreeBSD:
# /usr/local/bin/mailgw-licensemanager -s
```

The server console will display information similar to the following:

```
Kaspersky license manager for Linux. Version
5.6.0/RELEASE
Copyright (C) Kaspersky Lab, 1997-2008.
```

```
Portions Copyright (C) Lan Crypto

License info:
Product name: Kaspersky Mail Gateway
Expiration date: 02-06-2008, expires in 34 days

Active key info:
Product name: Kaspersky Mail Gateway
Key file     00086CA1.key
Type:      Commercial
Expiration date: 02-06-2008
Serial:     0007-000487-00086CA
```

*To view information about a particular key, enter, the following in the command line:*

in Linux:

```
# /opt/kaspersky/mailgw/bin/mailgw-licensemanager -k
00053E3D.key
```

where `00053E3D.key` is the name of the product key file.

*in FreeBSD:*

```
# /usr/local/bin/mailgw-licensemanager -k
00053E3D.key
```

The server console will display information similar to the following:

```
Kaspersky license manager. Version 5.6.0/RELEASE
Copyright (C) Kaspersky Lab. 1998-2008.
Portions Copyright (C) Lan Crypto
Product name: Kaspersky Mail Gateway
Creation date: 02-12-2007
Expiration date: 02-06-2008
Serial      0007-000487-00086CA
Serial 02B1-000454-00053E3
Type: Commercial
Lifespan:    91
```

## 5.6.2. Renewing your product key

Renewing the Kaspersky Mail Gateway key gives you the right to re-enable full product functionality, and to resume the additional services listed in section 5.6 on p. 71.

The validity period of the key depends on the product you bought, and the type of the license you purchased. The license for Kaspersky Mail Gateway is usually issued for one year.

*To renew the Kaspersky Mail Gateway key:*

Contact the company that sold you the application and renew your key for Kaspersky Mail Gateway.

*or*:

Purchase a key directly from Kaspersky Lab. Write a letter of request to the Sales Department of our company at sales@kaspersky.com or fill in the corresponding form on our website (www.kaspersky.com), in the section **E-Store → Renew Your License**. After your payment is received, we will send a license key to the e-mail address indicated in the corresponding field of your license renewal form.

*To install a new license key, enter the following in the command line:*

*in Linux:*

```
# /opt/kaspersky/mailgw/bin/mailgw-licensemanager -a
00053E3D.key
```

where `00053E3D.key` is the name of the product key file .

*in FreeBSD:*

```
# /usr/local/bin/mailgw-licensemanager -a
00053E3D.key
```

If the installation is successful, the server console will display information similar to the following:

```
Kaspersky license manager. Version 5.6.0/RELEASE
Copyright (C) Kaspersky Lab. 1998-2008.
Portions Copyright (C) Lan Crypto
Key file 00053E3D.key is successfully registered
```

You are advised to update the anti-virus database after the installation.

If you want to install a new key before the current one expires, it can be added as a backup key. The backup key will be activated immediately after the current one expires. The term of validity for the additional key starts from the activation date. You can install only one backup key.

If you have installed two keys (the current and an additional one), information about both of them can be viewed on the server console.

## 5.6.3. Removing a key

*To remove the current license key and the backup key (if it is installed), enter the following in the command line:*

*in Linux:*

```
# /opt/kaspersky/mailgw/bin/mailgw-licensemanager -da
```

*in FreeBSD:*

```
# /usr/local/bin/mailgw-licensemanager -da
```

If the component removes the license key(s) successfully, the server console will display the following (or similar) information:

```
Kaspersky license manager. Version 5.6.0/RELEASE
Copyright (C) Kaspersky Lab. 1998-2008.
Portions Copyright (C) Lan Crypto
Active key was successfully removed
```

*To remove a backup key, enter the following in the command line:*

*in Linux:*

```
# opt/kaspersky/mailgw/bin/mailgw-licensemanager -dr
```

*in FreeBSD:*

```
# /usr/local/bin/mailgw-licensemanager -dr
```

The server console will display the following (or similar) information:

```
Kaspersky license manager. Version 5.6.0/RELEASE
Copyright (C) Kaspersky Lab. 1998-2008.
Portions Copyright (C) Lan Crypto
Additional key was successfully removed
```

# CHAPTER 6. ADVANCED APPLICATION SETTINGS

This chapter discusses in detail the advanced settings of Kaspersky Mail Gateway. In contrast to the main settings that provide the application functionality, advanced settings can be configured optionally at the administrator's discretion.

**Attention!**

Restart the application to apply modified settings.

## 6.1. Configuring anti-virus protection of e-mail traffic

Application parameters in the **[mailgw.policy]** section define modes for message scanning and disinfection. They also and enable/disable the scanning of archives and e-mail attachments (the **AVScanArchives** and **AVScanMailBases** parameters respectively).

### 6.1.1. Setting up application timeouts

**Attention!**

All timeout settings are located in the **[mailgw.timeouts]** section of the application configuration file.

By setting up various timeouts, the administrator can:

- Limit the maximum period during which the application will attempt to deliver unsent outgoing messages (**MaximalBackoffTime** parameter, in seconds).
- Limit the minimum time which should elapse before the application will attempt to re-send undelivered messages (**MinimalBackoffTime** parameter).
- Specify the interval during which the application will try to deliver messages, at the frequency defined by the **MinimalBackoffTime** and **MaximalBackoffTime** parameters (**MaximalQueueLifetime** option).

After this period elapses, the unsent message will be removed from the ready-to-send queue. If necessary, a DSN message about the initial message delivery failure will be generated.

- Specify timeouts for intercepting various network operations (for the Sender and Receiver modules), such as:
  - Network reading timeout (**ReadTimeout** option**).** The default timeout specified in the application's configuration file is the optimal value for most cases and it is advisable not to alter it.
  - Network writing timeout (**WriteTimeout** option**).** The default timeout specified in the application‘s configuration file is the optimal value for most cases, and it is advisable not to alter it.
- Specify timeouts used by the application to send messages:
  - Maximum time for receiving data from the remote server when establishing an SMTP session (**SendingInitialTimeout** option).
  - Maximum time to start an e-mail session (command HELO/EHLO) (**SendingHelloTimeout** option).
  - Timeout for receiving a response from the remote server to the MAIL FROM command (**SendingMailTimeout** option).
  - Timeout for defining the recipient (RCPT TO command) (**SendingRcptTimeout** option).
  - Timeout for initiating data transfer (DATA command) (**SendingDataInitiationTimeout** option).
  - Timeout for stopping data transfer (CRLF.CRLF sequence) to the remote server (**SendingDataTerminationTimeout** option).
  - Timeout for quitting the current e-mail session (QUIT command) (**SendingQuitTimeout** option).
- Specify timeouts used by the application to receive messages:
  - Timeout for starting the DATA command (**ReceivingDataInitiationTimeout** option).
  - Timeout for stopping data transfer by the remote server (**ReceivingDataTerminationTimeout** option).
  - Timeout for waiting for the HELO/EHLO, MAIL FROM, RCPT TO and QUIT commands from the remote server (**ReceivingCommandTimeout** option).
- Timeout for object processing by the AV module (**ScanTimeout** option).

- Specify timeouts used by the application during communication with DNS servers:
  - Timeout for sending a query to DNS server and arrival of its response (**DNSNetworkTimeout** option).
  - Timeout for the total time it takes to receive response from DNS server for all attempts (**DNSResolveTimeout** option).
  - Timeout for storage of a DNS record in DNS cache (**DNSCacheMaximalTTL** option).
  - Timeout for storage of a DNS record for unreachable servers in mailgw cache (**UnreachableCacheTTL** option).

## 6.1.2. Setting performance restrictions

Kaspersky Mail Gateway allows the administrator to set certain limits when working with the application, which may reduce the load on the server and increase performance. In addition, the application of network restrictions may prevent some types of virus outbreaks and DOS attacks, which attempt to paralyze mail servers with huge volumes of e-mail traffic.

**Attention!**

You can find all restriction settings in the **[mailgw.limits]** section of the application's configuration file.

You can set the following restrictions:

- Number of objects simultaneously processed by the Receiver, Sender and AV modules (the **IncomingSessions**, **OutgoingSessions**, and **AntiviralSessions** options, respectively).
- Maximum number of message hops (**MaximalIncomingHops** option). Set this parameter to avoid looping due to incorrect configuration of the routing table.
- Limit the maximum size for messages received by the server (**MaximalIncomingMessageSize** option), and the total number of messages received during one e-mail session (**MaximalIncomingMessagesPerSession** option).
- Limit the number of recipients of a single message (**MaximalIncomingRcptsPerMessage** option). This parameter prevents spam addressed to your users).
- Maximum size of a single e-mail session (**MaximalIncomingSessionSize** option).

- Maximum number of simultaneous connections from the same IP (or host) that are processed by the Receiver and by the Sender modules (**MaximalIncomingSessionsPerIP** and **MaximalOutgoingSessionsPerHost** options respectively).
- Minimum size of available disk space on the partition where the application's working queue is stored (the **MinimalQueueFreeSpaceSize** option). If during the application's operation the queue size increases to the point that the available space is below this value, the application will temporarily suspend receipt of new messages until the value returns to the specified limits.

If the e-mail traffic at your server exceeds the specified limits, you are advised to decrease the number of objects being simultaneously processed by the AV module (**AntiviralSessions** parameter) and the number of hops for a single message (**MaximalIncomingMessageSize** option). This will increase the application's performance and the message processing speed.

If your server has a low-speed Internet connection, the following actions are recommended:

- Decrease the number of objects being simultaneously processed by the Receiver and Sender modules (**IncomingSessions** and **OutgoingSessions** options).
- Decrease the maximum number of incoming messages received during a single session (**MaximalIncomingMessagesPerSession** option).

# 6.2. Setting up connection receiving interfaces

The set of interfaces and ports on which the application receives connections is defined by the **ListenOn** parameter in the **[mailgw.network]** section of the application's configuration file. By default, Kaspersky Mail Gateway listens for connections on port 25 using all available interfaces.

If a particular interface is to be used, rather than all available interfaces, or if it is necessary to use a port other than 25, additional settings configuration must be performed.

*For instance, To make the application wait for connections on port 1025 of interface 192.168.0.1:*

assign the following value to the **ListenOn** parameter in the **[mailgw.network]** section:

```
ListenOn=192.168.0.1:1025
```

To use several particular interfaces, create several **ListenOn** parameter records in the configuration file, for instance:

```
ListenOn=192.168.0.1:25
ListenOn=10.0.0.1:25
```

# 6.3. Setting up the routing table

The application does not include a local agent for message delivery, and therefore all incoming e-mail messages must be transferred to the local host on which the agent is installed.

The rules for transferring (routing) are set by the **ForwardRoute** parameter in the **[mailgw.forward]** section.

This parameter is specified using one of the following formats:

```
ForwardRoute=<address_mask> <recipient>
ForwardRoute=<address_mask> [<recipient>]
ForwardRoute=<address_mask> [<recipient>:<port>]
```

where:

<address_mask> – the address of the recipient of the messages (wildcards "*" and "?" can be used; if the parameter is assigned the value **any**, then any recipient's address may be used).

`<recipient>` is the name of the domain containing the mail server, to which (according to MX records) the e-mail must be sent.

[<recipient>:<port>] is the delivery point, using the recipient's IP address or host name, and port number.

For example, if you create the following record in section **[mailgw.forward]**:

```
ForwardRoute=*@example.com [localhost:1025]
```

then all e-mail messages to **example.com** will be sent to port 1025 of the local host after processing by the application.

If several routing rules must be specified, create several copies of the **ForwardRoute** parameter in the configuration file.

For example, if the section **[mailgw.forward]** contains these entries:

```
ForwardRoute=*@example.com [localhost:1025]
ForwardRoute=*@example.net [somehost.example.com]
ForwardRoute=*@example.org example.com
```

the following processing rules will be followed:

- forward all e-mail messages for domain **example.com** to port 1025 of the local host after processing by the application.
- forward all e-mail messages for domain **example.net** to port 25 of host **somehost.example.com** after processing by the application.
- forward all e-mail messages for domain **example.org** to MX-host of domain **example.com** after processing by the application (the domain will be determined at the time the message is sent).
- forward all other messages to the corresponding MX-hosts after anti-virus scanning and spam filtering.

**Attention!**

When more than one rule applies to a message, the rule used is the first one where the specified domain matches the domain of the message recipient.

## 6.4. Checking the configuration file syntax

Use the `-k` or `--check-config` key in the command line of the *mailgwd* application component to check the syntax of its configuration file.

If the configuration file contains no errors, no information will be output to the server console.

If the check reveals errors, the list of errors will be displayed in the console.

## 6.5. Syntax check in notification templates

The application allows syntax checks of notification templates to be made by the *mailgw-tlv* utility, which is installed by default in the directory */opt/kaspersky/mailgw/bin/* (in Linux distributions) or in */usr/local/bin/* (for FreeBSD distributions).

*To check the syntax of a notification template, enter the following in the command line:*

*in Linux:*

```
> /opt/kaspersky/mailgw/bin/mailgw-tlv ./dsn.tmpl
```

*in FreeBSD:*

```
> /usr/local/bin/mailgw-tlv ./dsn.tmpl
```

The utility will output, to the server console, a report similar to the example below:

```
Kaspersky Template Language Verifier, version
5.6.12/RELEASE,
Copyright (C) Kaspersky Lab, 1997-2008


Parsing error: Unexpected end of line in the declara-
tion, line 63
```

If a template check is successful, the utility will report that template syntax is correct. In case of errors it will display a description of possible failure. The utility's return codes are described in section B.13 on p. 170.

# 6.6. Work with e-mail archives and the quarantine directory

The *mailgw-maila* utility allows the management of objects stored in the quarantine directories, or in the archives of incoming/outgoing messages. The *mailgw-maila* utility is installed by default to the */opt/Kaspersky/mailgw/bin/* directory (in Linux) or */usr/local/bin/* directory (in FreeBSD).

It has the following functionality:

- Reviewing the whole storage contents, or information about certain messages, for example, in Linux:

  ```
  > /opt/kaspersky/mailgw/bin/mailgw-maila
  --show-all --archive-
  path=/var/opt/kaspersky/mailgw/arch_in
  ```

  In FreeBSD:

  ```
  > /usr/local/bin/mailgw-maila
  --show-all --archive-
  path=/var/db/kaspersky/mailgw/arch_in
  ```

The following (or similar) information will be output to console:

```
Kaspersky Mail Archives Manager, version
5.6.12/RELEASE,
Copyright (C) Kaspersky Lab, 1997-2008


--QueueID--Status-Size-------ArrivalTime---------
------Sender.../Recipient...
jFDpgGKo70777 av/suspicious 1065 Mon, 12 Dec 2007
15:13:51 +0300 10.0.0.1 <test@example.com> ->
<test2@example.com>
jFDpg4Hc04120 av/error 1056 Mon, 12 Dec 2007
15:13:51 +0300 10.0.0.1 <test@example.com> ->
test2@example.com


Total: 2 archived messages, 11425 bytes.
```

The utility outputs information about messages in a storage directory in the following format:

```
ID STATUS SIZE DATE IP <SENDER> -> <RECIPIENT>
```

where:

ID – identification number of a stored message

STATUS – message status reflecting its current state.

A stored message may have any of the following statuses:

- incoming – message from the archive of incoming mail;
- outgoing – message from the archive of outgoing mail;
- as/spam – message with the **Spam** status, assigned by the anti-spam module;
- as/probable – message with the status **Probable Spam**, assigned by the anti-spam module;
- as/formal – message with the **Formal** status assigned by the anti-spam module;
- as/blacklisted – message with the **Blacklisted** status assigned by the anti-spam module;
- av/clean – message with the **Clean** status, assigned by the AV module;

  - `av/disinfected` – message with the **Disinfected** status, assigned by the AV module;
  - `av/infected` – message with the **Infected** status, assigned by the AV module;
  - `av/suspicious` – message with the **Suspicious** status, assigned by the AV module;
  - `av/protected` – message with the **Protected** status, assigned by the AV module;
  - `av/error` – message with the **Error** status, assigned by the AV module;
  - `av/filtered` – message with the **Filtered** status, assigned by the AV module.

  `SIZE` – message size (may be specified in bytes, kilobytes, or megabytes as determined by the respective prefixes);

  `DATE` – time and date that the message was received by the application;

  `IP` – IP address of message sender;

  `SENDER` – message sender's address;

  `RECIPIENT` – message recipient's address (the field may contain several values).

- Removal of all messages, or a specified message, from storage, for example, in Linux:

```
> /opt/kaspersky/mailgw/bin/mailgw-maila
--remove-all=jHrWPC7s86253 --archive-
path=/var/opt/kaspersky/mailgw/arch_in
```

  In FreeBSD:

```
> /usr/local/bin/mailgw-maila
--remove-all --archive-
path=/var/db/kaspersky/mailgw/arch_in
```

  The following (or similar) information will be output to console:

```
Kaspersky Mail Archives Manager, version
5.6.12/RELEASE,
Copyright (C) Kaspersky Lab, 1997-2008

Total: 4586 archived messages have been removed.
```

- Sending of all messages/certain messages from storage directories to their original recipients, for example, in Linux:

  ```
  > /opt/kaspersky/mailgw/bin/mailgw-maila
  --send-id=jHrWPC7s86253 --archive-
  path=/var/opt/kaspersky/mailgw/arch_in
  ```

  In FreeBSD:

  ```
  > /usr/local/bin/mailgw-maila
  --send-id=jHrWPC7s86253 --archive-
  path=/var/db/kaspersky/mailgw/arch_in
  ```

  The following (or similar) information will be output to console:

  ```
  Kaspersky Mail Archives Manager version
  5.6.19/RELEASE,
  Copyright (C) Kaspersky Lab, 1997-2008

  Message with QueueID jHrWPC7s86253 will be sent
  asap.
  ```

**Attention!**

If the --*send-id* command line option is specified, the selected message must pass anti-virus scanning and anti-spam filtering procedure before it is delivered to the recipient. To send a message from storage without anti-virus scanning and anti-spam filtration, use the *-send-id-without-check* command line option.

**Note**

Descriptions of command line options for *mailgw-maila* utility can be found in section B.16 on p. 181, and its return codes are described in section B.17 on p. 182.

# 6.7. Management of application working queue

While the application is running, it creates a working queue of messages for processing by the anti-spam and AV modules.

The *mailgw-mailq* utility, which is installed by default in the directory */opt/kaspersky/mailgw/bin/* (in Linux distributions) or in */usr/local/bin/* (for

FreeBSD distributions) allows the management of messages in the working queue.

It has the following functionality:

- Reviewing the contents of the working queue, or supplying information on specific messages in it.

  *To display information about all messages in the working queue, enter the following in the command line (in Linux):*

  ```
  > /opt/kaspersky/mailgw/bin/mailgw-mailq
  --show-all
  ```

  *In FreeBSD:*

  ```
  > /usr/local/bin/mailgw-mailq --show-all
  ```

  The utility will output to the server console a report similar to the example below:

  ```
  Kaspersky Mail Queue Manager, version
  5.6.12/RELEASE,
  Copyright (C) Kaspersky Lab, 1997-2008


  --QueueID--Status-Size-------ArrivalTime---------
  ------Sender.../Recipient...
  iAgUF4Oi21098 WFS 1570 Tue, 12 Feb 2007 10:42:30
  +0000 10.0.0.28 <test2@scmsmtpgw1.example.com> ->
  <test1@scmsmtpgw1.example.com>
  iAgVF4Qs38118 WFC 897 Tue, 12 Feb 2007 10:42:31
  +0000 10.0.0.16
  <test2@scmsmtpgw1.example.com> ->
  <test1@scmsmtpgw1.example.com>
  iAgTF45Y97588 SND 1048 Tue, 12 Feb 2007 10:42:29
  +0000 10.0.0.16 <test2@scmsmtpgw1.example.com> ->
  <test1@scmsmtpgw1.example.com>


  Total: 3 queued messages, 3515 bytes.
  ```

  The application outputs information, about messages in the working queue, in the following format:

  ```
  ID STATUS SIZE DATE IP <SENDER> -> <RECIPIENT>
  ```

  where:

  ID – identification number of a queued message;

  `STATUS` – message status reflecting its current state.

A message in working queue may have any of the following statuses:

- `WFC` – message waiting for anti-spam filtration and anti-virus scanning;
- `CHK` – message being scanned for virus presence;
- `WFS` – message waiting for creation of its virtual copies;
- `SPL` – message being used for creation of virtual copies;
- `QUE` – message waiting to be sent to its recipient;
- `SND` – message being sent.

`SIZE` – message size, which may be specified in bytes, kilobytes, or megabytes as determined by the respective prefixes;

`DATE` – time and date that the message was added to the queue;

`IP` – IP address of message sender;

`SENDER` – message sender's address;

`RECIPIENT` – message recipient's address (the field may contain several values).

- Removal of all messages, or a specified message, from the working queue.

  *To remove all messages from the working queue, enter the following in the command line (in Linux):*

  ```
  > /opt/kaspersky/mailgw/bin/mailgw-mailq --
  remove-all
  ```

  *In FreeBSD:*

  ```
  > /usr/local/bin/mailgw-mailq --remove-all
  ```

  The utility will output to the server console a report similar to the example below:

  ```
  Kaspersky Mail Queue Manager, version
  5.6.12/RELEASE,
  Copyright (C) Kaspersky Lab, 1997-2008


  Total: 12 queued messages have been removed.
  ```

**Attention!**

A message can only be removed from the queue if its status is WFC, WFS or QUE .

- Send all or selected messages ahead of the general queue, for example, in Linux:

  ```
  > /opt/kaspersky/mailgw/bin/mailgw-mailq
  --send-id=jHrWPC7s86253
  ```

  In FreeBSD:

  ```
  > /usr/local/bin/mailgw-mailq
  --send-id=jHrWPC7s86253
  ```

  The following (or similar) information will be output to console:

  ```
  Kaspersky Mail Queue Manager, version
  5.6.12/RELEASE,
  Copyright (C) Kaspersky Lab, 1997-2008


  Message with QueueID jHrWPC7s86253 will be sent
  asap.
  ```

**Attention!**

A message can be sent ahead of the general queue only if it has the status QUE (expects delivery to the recipient).

**Note**

Descriptions of command line options for *mailgw-mailq* utility can be found in section B.15 on p. 180, and its return codes are described in section B.17 on p. 182.

# 6.8. Managing the application

While Kaspersky Mail Gateway is running, it can be managed using scripts, signals, and the command line.

This section describes how to manage the application using scripts. For management options using signals, see section B.3 on p. 155, and for information about using files, see B.4 on p. 155).

**Attention!**

Application management using scripts requires privileged user (**root**) rights.

*If you use the Linux distribution package to run the management script, enter the following at the command line:*

```
# /opt/kaspersky/mailgw/lib/bin/mailgw <action>
```

or use the link:

```
# /etc/init.d/mailgw <action>
```

*If you use the FreeBSD distribution package, run the management script by entering the following:*

```
# /usr/local/etc/rc.d/mailgw.sh <action>
```

Table 1 contains possible values of the `<action>` parameter:

Table 1. Management script parameters

| **Value** | **Meaning** |
|---|---|
| start | Start the application. |
| stop | Stop the application. |
| restart | Stop and then start the application. |
| reload | Reinitialize the main application component, reload the anti-virus database and the configuration file, and restart the anti-spam module. |
| reload-bases | Reload the anti-virus databases and restart the anti-spam module. |
| status | Request the application's status. |
| stats | Request the application's statistics. |
| recv-off | Suspend the operation of the Receiver module. |
| recv-on | Resume the operation of the Receiver module. |
| send-off | Suspend the operation of the Sender module. |

| Value | Meaning |
|---|---|
| send-on | Resume the operation of the Sender module. |
| check-off | Suspend the operation of the scanning module. |
| check-on | Resume the operation of the scanning module. |
| clear-stats | Reset statistics. |
| post-update | Load Kaspersky Mail Gateway databases after their successful downloading. |

When the Receiver module is suspended, mail servers will be unable to establish connection with Kaspersky Mail Gateway to transfer messages to recipients within your e-mail system. Messages already added to the work queue will be treated as normal, that is scanned for viruses and spam signs, processed in accordance with the existing rules and forwarded to the recipients (unless the rules block their delivery).

When the Sender module is suspended, the application stops transmitting processed messages. Processed messages will be preserved in the work queue of outgoing messages. Suspension of the Sender module does not affect the Receiver module. Receipt of messages from mail servers will not be suspended.

When the scanning module is suspended, e-mail messages accepted by the Receiver module will be transferred directly to the Sender module for subsequent delivery to recipients. Anti-virus scanning, spam filtering and message processing will not be performed.

## 6.9. Control of application activity

A special watchdog process ensures that individual application modules function correctly while the software is running. As soon as the application starts, it creates a child process to monitor the application. If after a specified interval the parent process receives no confirmation of correct operation from any module, the watchdog process restarts the application.

**Attention!**

You can control timeouts of the watchdog process using the application command line options. See section B.6 on p. 163 for details.

# 6.10. Customizing date and time formats

Kaspersky Mail Gateway generates reports on the activity of every component. This information always contains the date and time of report generation.

By default, Kaspersky Mail Gateway displays the date and time using the strftime standard:

- **%H:%M:%S** – displayed time format.
- **%d-%m-%Y** – displayed date format.

The administrator can customize how time and date information are displayed in the **[locale]** section of the application configuration file. You can specify one of the following formats:

- **%I:%M:%S %P** – display time in 12-hour format (**TimeFormat** parameter).
- **%y/%m/%d** or **%m/%d/%y** – display date (**DateFormat** parameter) as **yy/mm/dd** or **mm/dd/yy**, respectively).

# 6.11. Reporting options

The performance of the main application component is recorded either in the application log file in plain text format (**LogFilename** option in the **[mailgw.options]** section) or in the system log (syslog). The data is not logged if the **LogFilename** option is not defined (**LogFilename=**).

To customize the output data, change the *report detail level* (**LogLevel** option in the **[mailgw.options]** section).

**Report detail level** is a number that defines the level of reported details for application performance data. Each subsequent level of detail contains all the details from the previous level, and adds new information.

Table 2 below lists the possible report detail levels.

Table 2. Report detail levels

| Level | Level description | Letter symbol | Meaning |
|---|---|---|---|
| 0 | Fatal Errors | F | Only information regarding critical errors which terminate the program, due to the impossibility of |

| Level | Level description | Letter symbol | Meaning |
|---|---|---|---|
| | | | executing an action. For example, a component is infected, or scanning, database loading, or product key loading failed. |
| 1 | Errors | E | Information about other errors that may or may not lead to application shutdown, for example, file scan errors. |
| 2 | Warning | W | Notifications about errors that may lead to the application shutdown (product key expiration warning, out-of-disk-space warning, etc.). |
| 3 | Info, Notice | I | Important informational messages, such as whether a component is running or inactive, the path to the configuration file, latest changes in the scan area, database updates, product keys, statistics summary. |
| 4 | Activity | A | Messages on scanning of files according to the report detail level. |
| 9 | Debug | D | All debug messages. |

Information about fatal errors is always displayed, regardless of the report detail level. The optimal level is level **4**, which is also the default level.

Information messages may be divided into the following types:

- Messages about actions on e-mail messages.
- Notifications about system events.
- Other messages (component start, loading of databases, return codes, etc.).

The output format for each of the levels of detail listed above is as follows:

- for messages about actions on e-mail messages:

```
[date time detail_level] envelope-id: MESSAGE;
```

- for all other types of message:

  `[date time detail_level]: MESSAGE,`

  where:

  - `[date time detail_level]` gives the date and the time (in the format specified by the administrator in the **[locale]**) section, and the letter indicating the report detail level.
  - `envelope-id` – e-mail message identifier in the working queue of the application, which identifies the e-mail message.
  - `MESSAGE` – message text that may have different formats depending on the message type.

For the text of report messages containing information about actions on e-mail messages, see section B.20 on p. 187.

# 6.12. Adding supplementary information to messages

The application supports two methods of adding supplementary information to e-mail messages:

- Adding an extension header field to the e-mail message.

  The information may describe the application's version, date when the anti-virus databases were last updated, or the time and result of anti-virus and anti-spam scanning of the message (determined by the **AddXHeaders** parameter in the **[mailgw.policy]** section of the application configuration file).

  Header format:

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

  `where:`

  `YYYY` – year indicated in four-digit format;

  `MM – month;`

  `DD – date;`

  `HH – hour;`

  `MM – minute;`

```
    SS - second.
```

E.g.:

```
X-Anti-Virus:   Kaspersky   Mail   Gateway,   version
5.6.12/RELEASE,   bases:   20080118T085614   #519212,
check: 20080118 clean
```

For detailed information about the headers added to messages by the anti-spam module, please see section B.18 on page 183.

- Adding a disclaimer text to the e-mail message’s body.

  The information will be added as plain text; it may contain a statement generated in accordance with the security policy (or other rules) of a specific organization (the **AddDisclaimer** parameter in the **[mailgw.policy]** section). The default message text notifies that the message has been scanned by Kaspersky Mail Gateway. The administrator can modify the information format (e.g., generate disclaimer message as a HTML text).

## 6.13. Control of application activity via SNMP

Beginning with version 5.6, the application provides read-only access to the following information via Simple Network Management Protocol (SNMP):

- *Configuration of the application* – information about all parameters from all sections of the program configuration file.
- *Activity statistics* – statistical information about application operations.

Availability of the information via SNMP is defined by the **SNMPServices** parameter in the **[mailgw.snmp]** section of the configuration file.

The application provides the following data accessible through SNMP:

- Information about configuration of the application.
- Statistics of application activity:
  - Date of application launch (in ISO 8601 format).
  - Time (seconds) passed since application start.
  - Date of the last successful update (in ISO 8601 format).
  - Total number of records in the current databases of Kaspersky Mail Gateway.

- Release date of the current application database update (in ISO 8601 format).

Interaction via SNMP is implemented in Kaspersky Mail Gateway using an SNMP subagent, which works in turn with SNMP master agent. Interaction parameters are listed in the **[mailgw.snmp]** section of the configuration file:

- **ConnectTo** – the option defines the socket for interaction. A local file or a network socket can be used. E.g.:

  **ConnectTo=unix:/path/to/dir/**

  or

  **ConnectTo=127.0.0.1:705**

- **PingInterval** – interval (seconds) that the subagent will use between attempts to connect to the master agent in case of disconnection.
- **Timeout** – timeout (seconds) for sending a request to the master agent.
- **Retries** – number of attempts to send a request to master agent.

The application can use as master any agent that supports the *AgentX* protocol. In this section the *NET-SNMP* agent is used as an example. Interaction is performed through a local socket.

**Attention!**

To ensure correct interaction with the application via AgentX, you are advised to use the NET-SNMP version 5.1.2 or later.

The following steps are necessary for configuration of the agent:

1. Modify the *snmpd.conf* configuration file adding the following lines to it:

   ```
   master agentx
   AgentXSocket tcp:localhost:705
   rocommunity public
   trapsink localhost
   ```

2. Modify the *snmp.conf* configuration file adding the following lines to it:

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

   The */opt/kaspersky/mailgw/share/snmp-mibs* (in Linux) or */usr/local/share/mailgw/snmp-mibs* path (in FreeBSD) defines the location of MIB files of Kaspersky Mail Gateway. If you have installed the application to a different directory, specify the path corresponding to your configuration.

3. Restart NET-SNMP.

> **Note**
>
> Detailed information regarding configuration of the *NET-SNMP* agent is available at its official site http://www.net-snmp.org/. To display information about snmpd.conf and snmp.conf use the program manual pages.

During data access via SNMP the following OID (object identifier) is used:

**1.3.6.1.4.1.23668.1159**

Administrator can configure the application to send SNMP traps when certain events occur. Generation of SNMP traps is regulated by the **SNMPTraps** option in the **[mailgw.snmp]** section of the configuration file. SNMP traps are generated when the following events occur:

- Reloading of the anti-virus databases (*TrapBasesReloaded, TrapBasesReloading*) or application configuration (*TrapConfigReloaded, TrapConfigReloading*).
- Application start/stop (*TrapStart*, *TrapStarting, TrapStop, TrapStopping*), critical error (*TrapError*).

# CHAPTER 7. TESTING APPLICATION OPERABILITY

After you install and configure Kaspersky Mail Gateway**,** it is recommended that you test its settings and operability by using the following three methods:

- Telnet program.
- Mail messages containing test phrases in the *Subject* header.
- Templates GTUBE.
- EICAR test virus.

## 7.1. Testing mail receipt and delivery using Telnet

*To test the application operation using Telnet it is necessary to:*

1. Connect to the server on which the application is installed using Telnet. To do so, enter the following at the command line:

   ```
   telnet <mailgw host address> <port>
   ```

   where the `<mailgw host address>` and `<port>` are the values assigned to the **ListenOn** option in the **[mailgw.network]** section of the application configuration file.

2. After the connection is established, wait for a response from the main application component. You will see the following information:

   ```
   220 example.org ESMTP
   ```

   where mailgw.company.com is the name of the server being tested.

3. After the connection to the server is confirmed, type the following at the command line:

   ```
   EHLO <fqdn>
   ```

   where `<fqdn>` stands for a full domain name of the host, which establishes connection.

You will see the following (or similar) information:

```
250-example.org hello user [127.0.0.1]
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE 10485760
250 DSN
```

where:

`mailgw.company.com` is the name of the server being tested

`user` is the client host name

`[127.0.0.1]` is the client IP address.

- Enter at the command line:

  ```
  MAIL FROM: <sender_address>
  ```

  You will see the following (or similar) information:

  ```
  250 2.1.0 OK
  ```

- Enter at the command line:

  ```
  RCPT TO: <recipient_address>
  ```

  You will see the following (or similar) information:

  ```
  250 2.1.0 OK
  ```

- Enter in the command line:

  ```
  DATA
  ```

  You will see the following (or similar) information:

  ```
  354 Start mail input; end with <CRLF>.<CRLF>
  ```

  Enter in the command line:

  ```
   From:xz@example.com
   To: xz@example.com
   Subject: test
   test
  .
  ```

  You will see the following (or similar) information:

  ```
  250 2.1.0 OK
  ```

4. If the response is `250 2.1.0 OK`, the test message has been successfully accepted by the server. After this, the message will be checked by the anti-spam module, scanned for viruses and then sent to the recipient in accordance with the routing table. You are advised to check message delivery. To verify the results, view the application statistics. One message will be added to the totals for scanned and sent messages.

# 7.2. Testing the anti-spam filtration

To test the Spamtest filter functionality, you must create e-mail messages containing specific phrases in the *Subject* header. Table 3 below contains a summary of test phrases and the corresponding Spamtest responses.

Table 3. Test messages

| **Test phrase in the *Subject* header** | **Response of the anti-spam module** |
|---|---|
| *Subject*: spam is bad do not send it<br>or<br>*Subject*: `t h i s i s n o t s p a m` | Based on the analysis, the message will be assigned the **Spam** status. |
| *Subject*: News and special events May | Based on the analysis, the message will be assigned the **Probable Spam** status. |
| *Subject*: Out of Office AutoReply | Based on the filter's analysis, the message will be assigned the status **Not detected**. The label **[--Formal Messages--]** will be added to its *Subject* header |
| Text of the *Subject* header contains invective. | Based on the filter's analysis, the message will be assigned the status **Not detected**. The label **[--Obscene--]** will be added to its *Subject* header/ |

Having sent a message with a test phrase in the Subject, you should check that the message has been processed in accordance with the specified rules: for instance, that the application has changed the specified message headers; or that the message has been added to the quarantine directory. If the application does not function properly, you should consult Kaspersky Lab's Technical Support.

Furthermore, you can test filtration using a special **GTUBE** (Generic Test for Unsolicited Bulk E-mail) template. Test of spam filtration using GTUBE is identical to the tests of anti-virus software based on EICAR test virus.

Create an e-mail message containing the following string (without spaces or hyphenation):

```
XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-E-
MAIL*C.34X
```

and send it to an e-mail account protected by Kaspersky Mail Gateway. After analysis the message will receive the **SPAM** status and the application will apply to it the action specified in the policy assigned for the account.

# 7.3. Testing the application using EICAR

This test "virus" has been developed by eicar (The European Institute for Computer Anti-Virus Research) specifically to verify the functioning of anti-virus software.

It IS NOT A VIRUS and contains no code that may harm your computer. However, most anti-virus products identify it as a virus, according to The European Institute for Computer Antivirus Research.

**Attention!**

Never use real viruses to test the operation of your anti-virus application!

The test "virus" can be downloaded from the official **EICAR** site at: http://www.eicar.org/anti_virus_test_file.htm. If you have no Internet access, you can create a test "virus" manually, by entering the line below into any text editor and save the file as **eicar.com**:

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-
FILE!$H+H*
```

The file that you downloaded from the **EICAR** website, or created in a text editor as described above, contains the body of a standard test "virus". The anti-virus application will detect it, flag it as **Infected** and perform the specified action for objects with this status.

To test the application's response to objects with other statuses, modify the body of the standard test "virus" by adding one of the prefixes below (see Table 4).

**Note**

You can verify the proper operation of Kaspersky Mail Gateway using modifications of the EICAR "virus" only if your anti-virus databases were last updated on or after October 24, 2003, or has the cumulative updates for October 2003.

Table 4. Test "virus" modifications

| **Prefix** | **Object type** |
|---|---|
| No prefix, standard test "virus"" | **Infected**. An error occurs during disinfection. The object will then be deleted. |
| SUSP- | **Suspicious** (unknown virus code). |
| WARN- | **Suspicious** (modified code of a known virus). |
| ERRO- | **Causes a scanning error** |
| CURE- | **Infected**. The object will be disinfected and the whole text of the infected file will be changed to CURE. |
| DELE- | **Infected**. The object will be deleted automatically. |

The first column of the table contains prefixes that should be added at the beginning of the line in the standard test "virus" file (e.g., `DELE-X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*`).

After adding a prefix to the test “virus”, save it to a file with another name, for example *eicar_dele.com*.

The second column of the table indicates how the application will analyze the prefixed file. The actions for each type of object are defined by the application's settings as customized by the administrator.

**Note**

You are advised to test Anti-Virus operation with objects in message body and attachments using both incoming and outgoing mail. To verify detection of viruses in message body, add the standard or modified “virus” text to the message body.

# CHAPTER 8. UNINSTALLING THE APPLICATION

To uninstall Kaspersky Mail Gateway from the server, you must be a privileged (**root**) user. If you are currently logged on under a user account with lesser privileges, log on as **root**.

**Attention!**

The uninstallation process will automatically stop the application!

When you are uninstalling the product, the application will be stopped, and all files and directories created during installation will be deleted. However, files and directories created or modified by the administrator, such as the application configuration file, notification templates, the quarantine directories, archives of received and sent messages, anti-virus and anti-spam databases, and the product key file, will remain.

There are several different ways to run the uninstall procedure, depending on the package manager you used. Below is a detailed description of these options.

*If you installed the application from the rpm package, type the following at the command line to uninstall Kaspersky Mail Gateway:*

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

*If you installed the application from the deb package, type the following at the command line to uninstall Kaspersky Mail Gateway:*

```
# 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 remove the application but preserve its configuration files.

*If you installed the application from the pkg package, type the following at the command line to uninstall Kaspersky Mail Gateway:*

```
# pkg_delete <package_name>
```

A message will inform you if the application was successfully removed from your server.

If the Webmin plug-in module has been installed for remote management of Kaspersky Mail Gateway, its removal can be performed manually using standard means for Webmin.

# CHAPTER 9. FREQUENTLY ASKED QUESTIONS

This chapter contains a discussion of questions most frequently asked by our users regarding the installation, configuration and operation of the application.

**Attention!**

Constantly growing Knowledge Base containing answers to frequently asked questions is available at Kaspersky Lab web site: http://support.kaspersky.com/mail_gateway. You can use it to find answers to the questions that are not mentioned below. You can also use online HelpDesk form to send a request to the Technical Support service (http://www.kaspersky.com/helpdesk).

*Question: Is it possible to use Kaspersky Mail Gateway with anti-virus products of other vendors?*

No. You are advised to uninstall anti-virus products of other vendors before installing Kaspersky Mail Gateway, to avoid software conflicts.

*Question: Why does Kaspersky Mail Gateway cause a certain decrease in server performance, noticeably loading the CPU?*

Virus detection is a computationally intensive mathematical problem requiring structural analysis, checksum calculation and mathematical data conversions. Processor time is therefore the main resource consumed by the program, and each new virus added to the anti-virus databases increases the overall scanning time. This is a necessary sacrifice for the security and safety of your data.

Other anti-virus products speed up scanning by excluding both viruses which are less easily detectable or less frequent in the geographic location of the anti-virus vendor, and file formats that require complicated analysis (e.g. PDF) from their databases.

In contrast, Kaspersky Lab believes that the purpose of its products is to establish real and complete security for its users.

Kaspersky Mail Gateway gives its users maximum protection. Experienced users can accelerate anti-virus scanning to the detriment of overall security by disabling scanning of various file types, but we do not recommend this practice to users who want the best protection.

For maximum user protection, Kaspersky Mail Gateway recognizes more than 700 formats of archived and compressed files. This is essential for anti-virus security, because harmful executable code may be hidden inside files of any recognized format. However, despite the daily growth in the number of viruses detected by *Kaspersky Mail Gateway* (approximately 30 new viruses appear daily) as well as the ever increasing number of recognized file formats, each subsequent version of our product functions faster than the previous one. This is achieved through the use of new, exclusive technologies, developed at Kaspersky Lab.

*Question: Why do I need the key file? Will my Kaspersky Mail Gateway work without it?*

No, Kaspersky Mail Gateway will not work without a key.

If you are still deciding whether or not to purchase Kaspersky Mail Gateway, we can provide you with a temporary key file (trial key), which will only work for either two weeks or a month. When this period expires, the key will be blocked.

*Question: What happens when the key validity period expires?*

After expiration of the key, Kaspersky Mail Gateway will continue operating, but updating of the anti-virus and content filtration databases will be disabled. The application will continue to clean infected objects and filter out spam, but using the old anti-virus and spam content databases.

If this happens, notify your system administrator, or contact the company from which you purchased *Kaspersky Mail Gateway,* or contact Kaspersky Lab directly for key renewal.

*Question: The application does not work. What should I do?*

If you have encountered a problem while using the application, first of all, please make sure that the solution to this problem is not described in this document (in particular, in this section) or at the **Services/Knowledge base** section of the Kaspersky Lab's web site (http://support.kaspersky.com/mail_gateway).

If you have not found the solution to your problem in the relevant documentation and the Knowledge base on the web site, we recommend that you contact Kaspersky Lab's Technical Support.

For solution of urgent issues please call us using the phone numbers in the **Contact Us** part of this document (see section D.2 on page 205). User support is available 24 hours a day in the Russian, English, French and German languages. Please note that you have to be a registered user to receive assistance and you must provide to the support technician your registration number (received with a retail box) or information about your purchase (in case if you have bought the product online).

In addition, you can contact the Technical Support service by filling a special form (http://support.kaspersky.com/helpdesk.html).

Please fill in the web form carefully. Enter precise information about the product of Kaspersky Lab that you are using, your registration data and try to describe your problem clearly. Specify the following information in mandatory fields:

- Request type. Select the category to which your request belongs.
- Name of the product of Kaspersky Lab that you are using (e.g., **Kaspersky Mail Gateway 5.6**).
- Request text. Describe the problem that you have encountered while using the product of Kaspersky Lab.
- Registration information. Specify the registration type: **license key** (if you have purchased a retail box) or **online order** (in case if you have bought the product online). Depending upon the selected registration type, use the field below to specify the serial number of your license or the number of your Internet order.
- E-mail address that the specialists of our Technical Support service can use to contact you.

In the next window of the web form enter your contact information, type the code of protection against automatic registration and click the **Submit** button. Experts at the Technical Support service will carefully examine your problem and help you as soon as possible.

*Question: What are the daily updates for?*

A few years ago viruses were transmitted on floppy disks, and adequate computer protection could be achieved by installation of an anti-virus program followed by rare updates to its anti-virus database. However, recent virus epidemics spread around the world in several hours, and anti-virus protection with old databases may be helpless against a new threat. To resist new viruses, you should update the anti-virus databases every hour.

Spam is a serious problem for all network users being a direct and obvious threat to businesses. According to the latest data, the volume of spam in the Internet is about 75-80 percent of the total e-mail volume, and new types of spam appear constantly. Fast response to the appearance of such unwanted messages, and blocking their spread, requires timely updates to the anti-spam database used for spam filtering. New updates to the anti-spam database are made available on Kaspersky Lab‘s update servers every three minutes.

*Question: Can an intruder deliberately replace the anti-virus or anti-spam databases?*

Every anti-virus and anti-spam database has a unique signature which is verified by Kaspersky products while accessing the database. If the signature does not correspond to the one assigned at the Kaspersky Lab, or if the date of the database is later than that of the product key expiry, Kaspersky Mail Gateway will not use it.

*Question: The application cannot be started, the Sender module does not work, etc. What should I do?*

If, due to incorrect settings, the number of running processes (threads) exceeds the maximum number permitted by the system, application performance may be affected or your system may freeze.

To solve this problem, you are advised to decrease the number of concurrently active incoming and outgoing e-mail sessions, and the number of objects being simultaneously scanned by the anti-virus module (**AntiviralSessions, IncomingSessions,** and **OutgoingSessions** parameters in the **[mailgw.limits]** section of the application configuration file).

*Question: What should I do to make man pages of the application available by the command* **man <name>***?*

To make the application's man pages available for the man <name> command, do the following:

- For Debian Linux, Ubuntu Linux, SUSE Linux distributions, add the following line to the **/etc/manpath.config** file:

  ```
  MANDATORY_MANPATH /opt/kaspersky/mailgw/share/man
  ```

- For Red Hat and Mandriva Linux distributions, add the following line to the **/etc/man.config** file:

  ```
  MANPATH /opt/kaspersky/mailgw/share/man
  ```

- For FreeBSD distributions, add the following line to the **/etc/manpath.config** file:

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

**Attention!**

In addition, if the MANPATH variable is used in your system, add to it the path to the directory containing the application's man pages, by running the following command:

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

# APPENDIX A. KASPERSKY MAIL GATEWAY CONFIGURATION FILE

This section contains a detailed description of the configuration file used by the application.

The default installation package of Kaspersky Mail Gateway includes the *mailgw.conf* file containing the default application settings.

This configuration file is divided into sections that contain the parameters of all the individual groups of application features.

Each section is described in the following way: the first line contains the heading **[section name]**, followed by lines containing the description of the parameter represented as **parameter=description**. The description of each section of the configuration file is completed by the header of the next section.

**Note**

Instead of **true|false** values for Boolean settings in the configuration file you may also use equivalent values: **yes|no y|n** or **1|0**.

**Attention!**

The options described as "required parameters" are critical for normal functioning of the application. Without these parameters the configuration file is invalid!

## A.1. Section *[path]*

The **[path]** section contains options that define the path to critical files, which are necessary for the application to work properly:

- **BasesPath** – full path to the directory containing the anti-virus and anti-spam databases of Kaspersky Mail Gateway.

  Required parameter.

  Default parameter value: **/var/opt/kaspersky/mailgw/bases** (in Linux) and **/var/db/kaspersky/mailgw/bases** (in FreeBSD).

- **LicensePath** – full path to the directory containing license keys. Required parameter.

  Default parameter value: **/var/opt/kaspersky/mailgw/licenses** (in Linux) and **/var/db/kaspersky/mailgw/licenses** (in FreeBSD).

- **TempPath** – full path to the directory for temporary files.

  Default parameter value: **/tmp**.

  If the parameter is not specified, the application uses the value of the **TMP** environment variable, if it is not available, that of the **TEMP** variable and the **/tmp** directory if the latter variable is not defined.

# A.2. Section *[locale]*

The **[locale]** section contains date and time formats:

- **DateFormat** – notation used by the components to display date in the report in **strftime** format.

  If the parameter is not specified, the default format is used: **%d-%m-%Y**.

- **TimeFormat** – notation used by the components to display date in the report in **strftime** format.

  If the parameter is not specified, the default format is used: **%H:%M:%S**.

  > **Note**
  >
  > You can alter the time presentation to 12-hour format (a.m., p.m.) using the string: **%I:%M:%S %P**.

# A.3. Section *[options]*

The **[options]** section contains various application parameters that do not belong to other groups:

- **User** – system account used to run the application components.

  The default value is **kluser**.

- **Group** – system group used to run the application components.

The default value is **klusers**.

# A.4. Section *[mailgw.access]*

The **[mailgw.access]** section includes the following options used to control access for SMTP clients:

- **ConnectRule** – rule that defines whether the application will accept or reject a session during connection establishment.

```
ConnectRule=(allow|deny) for any
ConnectRule=(allow|deny) for has_hostname
ConnectRule=(allow|deny) for no_hostname
ConnectRule=(allow|deny) for in_dnsbl
ConnectRule=(allow|deny) for out_dnsbl
ConnectRule=(allow|deny) for host <hostname_mask>
ConnectRule=(allow|deny) for ip <ip>
ConnectRule=(allow|deny) for ip <ip>/<netmask>
ConnectRule=(allow|deny) for ip
<ip>/<netmask_cidr>
ConnectRule=(allow|deny) for network <net-
work_name>
```

where:

- o **any** keyword allows any incoming SMTP connection;
- o **has_hostname|no_hostname** corresponds to the situation when the program can/cannot obtain the name of the calling host at its specified IP address;
- o **in_dnsbl|out_dnsbl** corresponds to the situation when the host's address is included/not included in the black lists of DNSBL services specified by the **DNSBlackList** parameter;

> **Note**
>
> Each DNSBL service is assigned a rating reflecting the level of its trustworthiness in the administrator's opinion (see section 5.2.5 on page 56).
>
> The **in_dnsbl** rule will be applied if the sum of ratings of the triggered DNSBL services reaches 100 or exceeds the value. Otherwise, if the sum of ratings is less than 100, the **out_dnsbl** rule will be applied.

- **<hostname_mask>** corresponds to a host name mask. Wild-cards can be used for mask definition:

  ***** (asterisk) – an arbitrary string of characters.

  ? (question mark) – a single arbitrary character.

- **<ip>** corresponds to the host IP address (in x.x.x.x format). E.g., 192.168.10.1;
- **<ip>/<netmask>** – mask of the IP addresses of recipients (in x.x.x.x/x.x.x.x format). E.g.: 192.168.0.0/255.255.0.0;
- **<ip>/<netmask_cidr>** – mask of IP addresses of recipients in CIDR format (recorded as x.x.x.x/y). E.g.: 192.168.0.0/16;
- **<network_name>** – corresponds to a network with appropriate name defined by the **NetworkName** parameter in the **[mailgw.network] section**.

You can specify several rules as a list. Rules are applied when an SMTP connection is established. The application uses the first rule from the list matching the specified conditions.

If a rule has been applied, the establishment/termination of an e-mail session will be determined by the specified **allow|deny** value.

If none of the rules has been triggered, the session will be accepted during connection establishment.

- **HeloRule** – rule that defines whether the application will accept or reject a session based on the information in HELO / EHLO command received from a client.

  **HeloRule** supports all rule recording methods available for **ConnectRule** as well as the following:

```
HeloRule=(allow|deny) helo eq <helo_string_mask>
HeloRule=(allow|deny) helo neq <helo_string_mask>
HeloRule=(allow|deny) helo has_ip
```

```
HeloRule=(allow|deny) helo no_ip
HeloRule=(allow|deny) helo same_ip
HeloRule=(allow|deny) helo diff_ip
```

where:

- **eq <helo_string_mask>|neq <helo_string_mask>** corresponds to the situation when the HELO / EHLO command value matches/ does not match the specified mask. Wildcards can be used for mask definition:

  ***** (asterisk) – an arbitrary string of characters.

  ? (question mark) – a single arbitrary character.

- **has_ip|no_ip** corresponds to the situation when it is possible/impossible to receive an address from the host name transferred by the client as a parameter for the HELO/EHLO SMTP command;
- **same_ip|diff_ip** corresponds to the situation when the address received from that name matches/does not match the actual address of the client that has established the connection.

You can specify several rules as a list. Rules are applied when an HELO / EHLO command is processed. The application uses the first rule from the list matching the specified conditions.

If a rule has been applied, the establishment/termination of an e-mail session will be determined by the specified **allow|deny** value.

If none of the rules has been triggered, the session will be accepted during connection establishment.

- **MailfromRule** defines the application response to an attempt to send a message from a certain source (i.e. based on the information in MAIL FROM SMTP command received from client).

  **MailfromRule** supports all rule recording methods available for **ConnectRule** and **HeloRule** as well as the following:

```
MailfromRule=(allow|deny) mailfrom eq <e-
mail_mask>
MailfromRule=(allow|deny) mailfrom neq <e-
mail_mask>
MailfromRule=(allow|deny) mailfrom in_dns
MailfromRule=(allow|deny) mailfrom out_dns
MailfromRule=(allow|deny) mailfrom is_empty
```

```
MailfromRule=(allow|deny) mailfrom not_empty
```

where:

- **in_dns** corresponds to the situation when it is possible to identify one of the following parameters: IP address of the sender's host (using the name transferred by the MAIL FROM SMTP command or MX record for the domain in sender's address);
- **out_dns** corresponds to the situation when none of these parameters could be identified;

> **Note**
>
> The above rules will not be applied if the argument value for the MAIL FROM command is empty.

- **is_empty** corresponds to the situation when the argument value of the MAIL FROM command is empty (< >);
- **not_empty** corresponds to a situation when the argument value of the MAIL FROM command is not empty (not <>).

You can specify several rules as a list. Rules are applied when a MAIL FROM command is processed. The application uses the first rule from the list matching the specified conditions.

If a rule has been applied, the establishment/termination of an e-mail session will be determined by the specified **allow|deny** value.

If none of the rules has been triggered, the session will be started.

- **RelayRule** defines the rules for e-mail routing.

  **RelayRule** supports all rule recording methods available for **ConnectRule**, **HeloRule** and **MailfromRule** in the following format:

  ```
  RelayRule=<rule> to <rcpt_mask>
  ```

  where

  - **<rule>** stands for a rule for **ConnectRule**, **HeloRule** or **MailfromRule** in format described above;
  - **<rcpt_mask>** corresponds to the mask of e-mail addresses of recipients. Wildcards can be used for mask definition:

    ***** (asterisk) – an arbitrary string of characters.

    ? (question mark) – a single arbitrary character.

You can specify several rules as a list. Rules are applied when RCPT TO commands received from client are processed. The application uses the first rule from the list matching the specified conditions.

If a rule has been applied, the establishment/termination of an e-mail session will be determined by the specified **allow|deny** value.

If none of the rules has been triggered, client access to the relay will be allowed.

Examples:

```
RelayRule=allow for ip 192.168.0.0/16 to *
RelayRule=allow for any to *@example.org
RelayRule=allow for any to *@example.org
RelayRule=deny for any to *
```

These example rules allow messages to be sent for clients from network 192.168.x.x to any recipient addresses; it allows sending of messages from all clients to recipients from `example.com` or `example.com` domains, blocking all other actions.

**Attention!**

Incorrect access settings for clients may allow the application to be used as an open e-mail relay.

- **DNSBlackList** – a list of DNS Black List servers (services). Specify the list of DNSBL services to be used during receipt of e-mail messages. If you are using several services, each service should have its own record, in the following format:

```
DNSBlackList=<service> N1
...
DNSBlackList=<service> Nn
```

where N1,…, Nn stand for the service rating (see section 5.2.5 on page 56). E.g.:

```
DNSBlackList=mail-abuse.org 70
DNSBlackList=bl.spamcop.net 30
DNSBlackList=block.blars.org 50
```

# A.5. Section *[mailgw.antispam]*

The **[mailgw.antispam]** section contains settings for the anti-spam filter:

- **ConnectTo** – path to the socket used for anti-spam module connection.

  The default value is: **unix:/var/run/mailgw/kas-filter.socket.**

  **ConnectTimeout=0, 60…3600** – timeout (seconds) for a connection to the anti-spam module.

  If the parameter is set to **0**, the timeout for connection establishment is not limited.

  The default value is **0** (it is used if the parameter is not defined).

- **RWTimeout=0, 60…3600** – timeout (seconds) for read/write operations during data exchange with the anti-spam module.

  If the parameter is set to **0**, the timeout for read/write operations during data exchange is not limited.

  The default value is **60** (it is used if the parameter is not defined)

- **FilterReceivedHeadersLimit=0…10** – the number of *Received* headers analyzed according to the lists of IP addresses and using DNSBL services.

  The default value is **2** (it is used if the parameter is not defined).

- **FilterUseUDS=true|false** – parameter that enables/disables mail checks involving UDS service.

  The default value is **false** (it is used if the parameter is not defined).

- **PidPath** – full path to the directory containing PID files.

  The default parameter value is: **/var/run/mailgw**.

- **LogLevel=0|1|2|3|4|5** – the level of details added to the system log (syslog).

  The default value is **2** (it is used if the parameter is not defined).

- **LogFacility=mail|user|local0|local1|local2|local3|local4|local5|local6|local7** – facility used to add records to the syslog.

  The default value is **mail** (it is used if the parameter is not defined).

- **FilterDNSTimeout=1...30** – timeout (seconds) for performance of all possible checks involving DNS.

The default value is: **10** (it is used if the parameter is not defined).

- **FilterUDSTimeout=1...30** – timeout for establishment of a connection between filtration server and UDS server. If the filtration server does not receive response from UDS server within the specified time interval, it attempts to connect to another UDS server of Kaspersky Lab.

  The default value is **10** (it is used if the parameter is not defined).

# A.6. Section *[mailgw.forward]*

The **[mailgw.forward]** section contains the following options for relaying messages through the application:

**ForwardRoute** – the routing table, containing message routing options. It explicitly defines, for specified domains or recipient addresses, the mail server to which the application will deliver messages addressed to the specified domains/addresses. The possible values include a mask of recipient addresses ("*" and "?" wildcards can be used) and the name / IP address of the mail server to which the application will connect for e-mail delivery. You may optionally specify the port to be used, if it is different from the standard one (port 25).

```
ForwardRoute=<address_mask> <recipient>
ForwardRoute=<address_mask> [<recipient>]
ForwardRoute=<address_mask> [<recipient>:<port>]
```

where:

- **<address_mask>** – address of message recipient («*» and «?» wildcards can be used; **any** value assigned to the parameter means that any recipient address is allowed);
- **<recipient>** – name of the domain whose e-mail server must receive the mail (according to MX records);
- **[<recipient>]** – destination (IP address or host name); e-mail traffic will be sent to port 25 of the specified host;
- **[<recipient>:<port>]** – destination (IP address or host name, port).

Please refer to section 6.3 on page 80 for details on configuration of the routing tables.

# A.7. Section *[mailgw.limits]*

The **[mailgw.limits]** section includes options that limit application functionality when e-mail traffic processing is performed:

- **AntiviralSessions=1…200** – maximum number of concurrently running anti-virus sessions.

  The default value is **10** (it is used if the parameter is not defined).

- **IncomingSessions=1…1024** – maximum number of open incoming sessions.

  The default value is **100** (it is used if the parameter is not defined).

- **ReceiverThreads=1…1024** – the number of threads handling incoming connections in the application process.

  The default value is **10** (it is used if the parameter is not defined).

  > **Attention!**
  >
  > Total number of e-mail connections processed simultaneously is defined by the **IncomingSessions** parameter. **ReceiverThreads** defines the size of threads pool implemented in the Receiver module. Each thread in the pool can handle a large number of incoming connections. Modification of the parameter value is recommended for configuration of application performance only.

- **OutgoingSessions=1…1024** – maximum number of open outgoing sessions.

  The default value is **20** (it is used if the parameter is not defined).

- **MaximalIncomingHops=1…100** – maximum number of intermediate hosts for a single message.

  The default value is **20** (it is used if the parameter is not defined).

- **MaximalIncomingMessageSize=64…204800** – maximum size (Kb) of an incoming message.

  The default value is **10240** (it is used if the parameter is not defined).

- **MaximalIncomingMessagesPerSession=1…1024** – maximum number of messages that can be received during one e-mail session.

  The default value is **20** (it is used if the parameter is not defined).

- **MaximalIncomingRcptsPerMessage=1…1024** – maximum number of recipients of a single message.

  The default value is **100** (it is used if the parameter is not defined).

- **MaximalIncomingSessionSize=64…2048000** – maximum size (KB) of incoming messages transferred within a single e-mail session.

  The default value is **102400** (it is used if the parameter is not defined).

- **MaximalIncomingSessionsPerIP=1…1024** – maximum number of open connections for e-mail receipt from a single IP address.

  The default value is **4** (it is used if the parameter is not defined).

- **MaximalOutgoingSessionsPerHost=1…1024** – maximum number of simultaneous connections for sending messages to a single host.

  The default value is **4** (it is used if the parameter is not defined).

- **MinimalQueueFreeSpaceSize=0…1000000** – minimum amount of available disk space on the partition where the application's working queue is located (MB). If the queue size increases, reducing the available space below the specified limit, the application will temporarily suspend receipt of new messages until the value returns to the defined minimum.

  If the parameter is set to **0**, the requirement for available free space on disk is disabled.

  The default value is: **0** (it is used if the parameter is not defined).

- **MaximalOutgoingMessagesPerSession=1…1024** – maximum number of messages that can be sent within a single e-mail session.

  The default value is **32** (it is used if the parameter is not defined).

# A.8. Section *[mailgw.network]*

The **[mailgw.network]** section includes the application's network settings:

- **ListenOn** – this option defines the interfaces and ports used by the *Receiver* module to receive e-mail traffic. It is specified as a table (list of values). Please refer to section 6.2 on page 79 for details on configuration of the interfaces for incoming connections.

  If the parameter is set to **0.0.0.0**, all available interfaces will be used.

The default value is **0.0.0.0:25** (it is used if the parameter is not defined).

- **Hostname** – host name that identifies the server on which the application is installed.

  Required parameter.

  The installer sets the parameter to **localhost** during standard application setup.

- **Postmaster** – e-mail address used by the application as the <postmaster> address.

  Required parameter.

  The installer sets the parameter to **postmaster@localhost** during standard application setup.

- **ProtectedDomains** – the list of domains for which anti-virus scanning and spam filtering of e-mail traffic is required. Wildcards can be used for mask definition:

  - * (asterisk) – an arbitrary string of characters.
  - ? (question mark) – a single arbitrary character.

  Required parameter.

**Attention!**

Make sure you have defined the list of protected domains, which traffic will be protected against malware and spam.

- **NetworkName** – the option defines the names of subnets, which will be used in access rules **(ConnectRule**, **HeloRule**, **MailfromRule**, **RelayRule**) and individual user groups.

  ```
  NetworkName = <networkname> ip <ip>
  NetworkName = <networkname> ip <ip>/<netmask>
  NetworkName = <networkname> ip <ip>/<netmask
  cidr>
  NetworkName = <networkname> host <hostname>
  ```

  where

  - **<ip>** corresponds to a host IP address (record format: x.x.x.x). E.g.: 192.168.10.1;
  - **<ip>/<netmask>** corresponds to a mask for IP addresses of recipients (record format: x.x.x.x/x.x.x.x). E.g.: 192.168.0.0/255.255.0.0;

- **<ip>/<netmask_cidr>** corresponds to a mask for IP addresses of recipients in CIDR format (record format: x.x.x.x/y). E.g.: 192.168.0.0/16;

  **<hostname>** corresponds to a mask for host names. Wildcards can be used for mask definition:

  * (asterisk) – an arbitrary string of characters.

  ? (question mark) – a single arbitrary character.

  You can define several values for the parameter as a list.

> **Note**
>
> You can specify a network with a special **Trusted** name. A subnet with such name is used to determine SMTP clients, incoming mail from which the application will not check for the signs of a DOS attack.
>
> E.g.: **NetworkName=Trusted ip 10.10.0.0/16**

# A.9. Section *[mailgw.options]*

The **[mailgw.options]** section contains miscellaneous settings of the main application daemon not included in other sections:

- **LogFilename** – full path and name of the log file to which results of *mailgw* component operation are written, in text format. If parameter value is an empty string (**LogFilename=**), the information is not logged. Information can also be written to system log (**LogFilename=syslog**).

  The default value is **syslog** (it is used if the parameter is not defined).

- **LogLevel=0|1|2|3|4|9** – the level of detail in the application work report (**0** – Fatal, **1** – Error, **2** – Warning, **3** – Info, **4** – Activity, **9** – Debug).

  The default value is **4** (it is used if the parameter is not defined).

- **StatFilename** – full path and name of the file that stores the application's performance statistics.

  The default value is **/var/opt/kaspersky/mailgw/stats/mailgw.stat** (in Linux) or **/var/db/kaspersky/mailgw/stats/mailgw.stat** (in FreeBSD).

- **PrependReceived=true|false** – if this option is enabled, the *mailgw* component will add the **Received:** header to scanned messages. If the parameter is set to **true**, the header will not be added.

  The default value is **true** (it is used if the parameter is not defined).

- **DSNTemplateFilename** – full path and name of the file used as a template for Delivery Status Notification messages.

  Required parameter.

  The default value is **/etc/opt/kaspersky/mailgw/templates/dsn.tmpl (in Linux) or /usr/local/etc/kaspersky/templates/dsn.tmpl** (in FreeBSD).

- **DSNEntireMessage=true|false** – if this option is enabled, the original message will be entirely included in the corresponding DSN messages.

  If the parameter is set to **false**, the application will only include the original message headers.

  The default value is **false** (it is used if the parameter is not defined).

- **DSNOnRelaying=true|false** – the option controls generation of a DSN notification upon successful delivery of a message.

  The default value is **false** (it is used if the parameter is not defined).

- **MessageStatistics** – full name of the file to which the application's statistical data will be logged.

  If parameter value is an empty string, the statistical data will not be logged.

  The default value is an empty string (it is used if the parameter is not defined).

- **DropMalformedMail=true|false** – an option that determines the delivery or removal of e-mail messages that do not meet the RFC standards.

  If the parameter is set to **false,** the application will make an attempt to make the message compliant with the standards, after which certain actions as per the configuration file settings will be performed. If the option is set to **true**, then messages that do not comply with the standards will not be delivered.

  The default value is **false** (it is used if the parameter is not defined).

- **Recode8BitMessages=true|false** – the option defines whether messages will be recoded during dispatch. When set to **true**, the option means that mail will only be recoded if mail gateway server has received during its receipt a direct indication that a message contains 8-bit characters while the server where it will be relayed after scanning does not support such functionality.

  The default value is **true** (it is used if the parameter is not defined).

- **Recode8BitMalformed=true|false** – the option determines the handling method for 8-bit characters which should be left unchanged during mail recoding. The option is only used if the **Recode8BitMessages is set to true**.

  When **Recode8BitMalformed** is set to **true,** it means that all 8-bit characters that should not be recoded will be replaced with "**?**". If the parameter value is **false**, 8-bit characters that should not be recoded will be skipped unchanged.

  The default value is **false** (it is used if the parameter is not defined).

# A.10. Section *[mailgw.path]*

The **[mailgw.path]** section contains paths used in the operation of the main application daemon:

- **QueuePath** – path to the directory containing the working queue of objects to be processed by the application.

  Required parameter.

  The default parameter value is **/var/spool/kaspersky/mailgw/** (both in Linux and FreeBSD).

- **ControlSocket** – complete name of the application control socket.

  Required parameter.

  The default parameter value is **/var/run/mailgw/mailgw.socket** (both in Linux and FreeBSD).

- **QueueBackupPath** – path to the directory where the application stores objects produced after cleaning of its working queue.

  Required parameter.

  The default parameter value is
  **/var/opt/kaspersky/mailgw/arch_spool** (in Linux)
  **/var/db/kaspersky/mailgw/arch_spool** (in FreeBSD).

- **CorePath** – directory for storage of core files containing RAM image and created in case of emergency shutdown of the application.

  Empty value disables RAM image creation. To enable it, define the path to the directory where RAM image files will be stored as the parameter value.

  The default value is empty (it is used if the parameter is not defined).

# A.11. Section *[mailgw.timeouts]*

The **[mailgw.timeouts]** section contains the application's timeout options:

- **MaximalBackoffTime=60…64800** – maximum period of time (seconds) that must elapse before the application will try to re-send an undelivered message.

  The default value is **21600** (it is used if the parameter is not defined).

- **MinimalBackoffTime=60…64800** – minimum time (seconds) that must elapse before the application will try to re-send an undelivered message.

  The default value is **1800** (it is used if the parameter is not defined).

- **MaximalQueueLifetime=1…14** – period (days) during which the application will try to send a message that was not delivered. If the message could not be delivered during the specified time, it will be deleted and a notification about failed delivery will be generated for its sender.

  The default value is **5** (it is used if the parameter is not defined).

- **ReadTimeout=1…1200** – timeout (seconds) for the *Receiver* module to read network data.

  The default value is **120** (it is used if the parameter is not defined).

- **WriteTimeout=1…1200** – timeout (seconds) for the *Sender* module to write network data.

  The default value is: **120** (it is used if the parameter is not defined).

- **ReceivingCommandTimeout=1…1200** – timeout (seconds) for waiting for SMTP protocol commands from a host: HELO/EHLO, MAIL FROM, RCPT TO (first such command) and QUIT SMTP protocol.

  The default value is **300** (it is used if the parameter is not defined).

- **ReceivingDataInitiationTimeout=1…2400** – timeout (seconds) for the DATA command of the SMTP protocol from a remote host. Note that the timeout for the first command RCPT TO is defined by the above parameter, whereas all subsequent RCPT TO commands must be transferred by the client within the time specified as **ReceivingDataInitiationTimeout**.

  The default value is **600** (it is used if the parameter is not defined).

- **ReceivingDataTerminationTimeout=1…7200** – timeout (seconds) for terminating data transfer (CRLF.CRLF sequences).

  The default value is **1800** (it is used if the parameter is not defined).

- **SendingInitialTimeout=1…1200** – timeout (seconds) for waiting for the response from a remote server when establishing an SMTP session.

  The default value is **300** (it is used if the parameter is not defined).

- **SendingHelloTimeout=1…1200** – timeout (seconds) for waiting for the response from a remote server to the HELO/EHLO command of the SMTP protocol.

  The default value is **300** (it is used if the parameter is not defined).

- **SendingMailTimeout=1…1200** – timeout (seconds) for waiting for the response from a remote server to the MAIL FROM command.

  The default value is **300** (it is used if the parameter is not defined).

- **SendingRcptTimeout=1…1200** – timeout (seconds) for waiting for the response from a remote server to the RCPT TO command of the SMTP protocol.

  The default value is **300** (it is used if the parameter is not defined).

- **SendingDataInitiationTimeout=1…2400** – timeout (seconds) for waiting for the response from a remote server to the DATA command of the SMTP protocol.

  The default value is **600** (it is used if the parameter is not defined).

- **SendingDataTerminationTimeout=1…7200** – timeout (seconds) for termination of the data transfer (CRLF.CRLF sequences).

  The default value is **1800** (it is used if the parameter is not defined).

- **SendingQuitTimeout=1…1200** – timeout (seconds) for waiting for the response from a remote server to the QUIT command of the SMTP protocol.

  The default value is **300** (it is used if the parameter is not defined).

- **DNSNetworkTimeout=1…10** – timeout (seconds) that defines the time interval allocated to send a request to DNS server and receive its response.

  The default value is **2** (it is used if the parameter is not defined).

- **DNSResolveTimeout=1…100** – timeout (seconds) that defines the total time allocated for receipt of DNS server response for all attempts.

  The default value is **10** (it is used if the parameter is not defined).

- **DNSResolveRetries=1…10** – maximum number of attempts to receive response from a DNS server.

  The default value is **5** (it is used if the parameter is not defined).

- **DNSCacheMaximalTTL=0…259200** – maximum time (seconds) during which a DNS record will be preserved in DNS cache.

  The default value is 4**3200** ((it is used if the parameter is not defined).

- **UnreachableCacheTTL=0…3600** – maximum time (seconds) during which a DNS record for an inaccessible server will be preserved in DNS cache.

  The default value is **600** (it is used if the parameter is not defined).

- **ScanTimeout=0, 60…3600** – time (seconds) during which the AV module can process a single object. If the scan remains unfinished after the specified time interval, the object receives the **Error** status (scan error).

  If the option is set to **0**, the duration for object processing by the AV module is not limited.

  The default value is **0** (it is used if the parameter is not defined).

# A.12. Section *[mailgw.archive]*

The **[mailgw.archive]** section contains paths to archives and the list of bcc addresses.

- **IncomingArchivePath** – path to the directory where an archive of all received e-mail messages is stored.

  If an empty value is defined or the parameter is missing, received mail will not be archived.

  The default parameter value is empty.

- **OutgoingArchivePath** – path to the directory where the archive of all outgoing mail is stored. If an empty value is specified or the parameter is not defined, the application will not archive sent mail.

  The default parameter value is empty.

- **IncomingBcc** – list of e-mail addresses where blind carbon copies of each received message will be sent prior to its scanning.

  The default parameter value is empty.

# A.13. Section *[mailgw.snmp]*

The **[mailgw.snmp]** section contains the settings for the SNMP network management protocol.

- **SNMPServices=true|false** – the option enables or disables control of application activity using SNMP. If the parameter is set to true, the data on application activity within LAN will be provided to the network administrator via the protocol.

  The default parameter value is **false**.

- **SNMPTraps=true|false** – the option controls generation of SNMP traps. The traps are used to inform network administrators about occurrence of specific situations.

  The default parameter value is **false**.

- **ConnectTo** – the option defines the socket for interaction of an SNMP subagent with SNMP master agent using the *AgentX* protocol; a local file or a network socket can be used (**unix:/path/to/dir/** or **127.0.0.1:705**).

  The default parameter value is **127.0.0.1:705**.

- **PingInterval=0…100000** – the option defines the interval (seconds) that the subagent will use to inform the master agent that it is online.

  The default parameter value is **30**.

- **Timeout=0…100000** – the option defines the timeout (seconds) for delivery of a request to master agent.

  The default parameter value is **5**.

- **Retries=0…100000** – the option defines the number of attempts to send a request to master agent. The attempts are performed once a second.

  The default parameter value is **10**.

# A.14. Section *[mailgw.policy]*

The **[mailgw.policy]** section contains the default settings for processing e-mail messages:

- **CheckAV=true|false** – defines the anti-virus scanning mode for all e-mail messages included in the particular group of recipients/senders. To disable the mode (i.e., configure the application to bypass the anti-virus scanning of messages), set the option to **false**.

  Required parameter.

  The default parameter value is **true**.

- **AVCure=true|false** – the mode for disinfection of revealed objects. Set the parameter to **true** to enable disinfection.

  Required parameter.

  The default parameter value is **false**.

- **AVScanArchives=true|false** – the mode for scanning of archives. Set the parameter to **false** to disable the mode.

  Required parameter.

  The default parameter value is **true**.

- **AVScanMailBases=true|false** – the mode for scanning of e-mail bases. Set the parameter to **false** to disable the mode.

  Required parameter.

  The default parameter value is **true**.

- **AVUseBasesSet=standard|extended** – the set of AV databases in Kaspersky Mail Gateway. The **extended** set contains in addition to the records from the **standard** set also signatures of riskware, such as adware, remote administration utilities, etc.

  Required parameter.

  The default parameter value is **standard**.

Parameters that define actions applied to objects, which have passed anti-virus scanning based on the status assigned after AV scan (see section 4.4. on page 44):

- **ActionDisinfected=cure|pass|remove|placeholder** – action to be applied to objects which have been disinfected successfully.

Required parameter.

The default parameter value is **cure**.

- **ActionInfected=pass|remove|placeholder** – action to be applied to infected objects.

  Required parameter.

  The default parameter value is **remove**.

- **ActionSuspicious=pass|remove|placeholder** – action to be applied to objects that are suspected of being infected with an unknown virus.

  Required parameter.

  The default parameter value is **remove**.

- **ActionProtected=pass|remove|placeholder** – action to be applied to objects that the application has failed to scan because they are password-protected.

  Required parameter.

  The default parameter value is **pass**.

- **ActionFiltered=pass|remove|placeholder** – action to be applied to objects filtered by name or MIME type.

  Required parameter.

  The default parameter value is **remove**.

Parameters that control the anti-spam module:

- **CheckSpam=true|false** – parameter which enables/disables spam filtering. Required parameter.

  Required parameter.

  The default parameter value is **true**.

- **SpamRateLimit=minimum|standard|high|maximum** – spam detection intensity. Four intensity levels are supported: **minimum**, **standard**, **high**, **maximum**. The **standard** level is recommended.

  Required parameter.

  The default parameter value is **standard**.

- **SpamMarkProbable=true|false** – the option enables addition of the **ProbableSpam** label to the headers of messages if scanning reveals that they belong to the corresponding category. If the parameter is set to **false**, the application will not recognize messages as probable spam.

Required parameter.

The default parameter value is **true**.

- **SpamMarkObscene=true|false** – the option enables addition of the **Obscene** label to the headers of messages if scanning reveals that they belong to the corresponding category. If the parameter is set to **false**, the application will not recognize messages as Obscene.

  Required parameter.

  The default parameter value is **true**

Parameters that define spam filtration criteria:

- **SpamUseDNS=true|false** – verification of information about message sender in DNS and DNS-based services: DNSBL, SPF, etc. The parameter controls the use of DNS services by the filtering module; individual services are enabled/disabled by their corresponding separate parameters.

  Required parameter.

  The default parameter value is **true**.

- **SpamCheckDNSBL=true|false** – the option defines whether the application will check the sender's IP address using the specified of DNSBL services.

  Required parameter.

  The default parameter value is **true**.

- **SpamCheckHostInDNS=true|false** – the option defines whether the application will check the presence of the sender's IP address in DNS.

  Required parameter.

  The default parameter value is **true**.

- **SpamCheckSPF=true|false** – the option defines whether the application will check the existence of the sender's IP address using SPF (Sender Policy Framework).

  Required parameter.

  The default parameter value is **true**.

- **SpamCheckSURBL=true|false** – the option defines whether the application will check the existence of the sender's IP address using SURBL (Spam URL Realtime Blocklists).

  Required parameter.

The default parameter value is **true**.

- **SpamHeadersToUndisclosed=true|false** – the option defines whether the application will check message headers for the presence of undisclosed recipients lists.

  Required parameter.

  The default parameter value is **true**.

- **SpamHeadersFromOrToDigits=true|false** – the option defines whether the application will check sender or recipient addresses for the presence of groups of digits. You are advised to set the parameter to **true**, if your e-mal addresses do not contain groups of digits.

  Required parameter.

  The default parameter value is **true**.

- **SpamHeadersFromOrToNoDomain=true|false** – the option defines whether the application will check the address for the missing domain part. You are advised to set the parameter to **false** for recipients that allow delivery of e-mail with incomplete addresses.

  Required parameter.

  The default parameter value is **true**.

- **SpamHeadersSubjectTooLong=true|false** – the option defines whether the application will check message subject for the presence of too long text strings. You are advised to set the parameter to **false** if such mail is allowed in your e-mail system.

  Required parameter.

  The default parameter value is **true**.

- **SpamHeadersSubjectWSOrDots=true|false** – the option defines whether the application will check message header for multiple spaces and dots. You are advised to set the parameter to **false** if such mail is allowed in your e-mail system.

  Required parameter.

  The default parameter value is **true**.

- **SpamHeadersSubjectDigitIDOrTimestamp=true|false** – the option defines whether the application will check message header for the presence of a digital identifier or time label.

  Required parameter.

  The default parameter value is **true**.

- **SpamHeadersMarkAllChinese=true|false** – the option defines whether the application will check message header for the presence of words in the Chinese language.

  Required parameter.

  The default parameter value is **true**.

- **SpamHeadersMarkAllKorean=true|false** – the option defines whether the application will check message header for the presence of words in the Korean language.

  Required parameter.

  The default parameter value is **true**.

- **SpamHeadersMarkAllThai=true|false** – the option defines whether the application will check message header for the presence of words in the Thai language.

  Required parameter.

  The default parameter value is **true**.

- **SpamHeadersMarkAllJapanese=true|false** – the option defines whether the application will check message header for the presence of words in the Japanese language.

  Required parameter.

  The default parameter value is **true**.

- **MarkSubject=spam|probable|formal|blacklisted** – the option adds to message header a prefix describing its status assigned to the letter by the anti-spam module after scanning (see section 4.2 on page 36).

  Required parameter.

  The default parameter value is **spam probable formal blacklisted**.

- **SpamCheckSizeLimit** – the option defines maximum size (Kb) of messages checked for the presence of spam. The anti-spam module will not scan larger messages.

  Required parameter.

  The default parameter value is **200**.

Parameters that define the addresses of notification sender and recipient:

- **AdminNotifyAddress** – e-mail address to which the application will send notifications for the administrator, regarding processing results for messages included in this group.

Required parameter.

The default parameter value is **postmaster@localhost**.

- **NotifyFromAddress** – e-mail address from which the application will send notifications regarding the scan results for messages of this group.

  Required parameter.

  The default parameter value is **MAILER-DAEMON@localhost**.

Parameters that define the paths to template files:

- **NotifyAdminTemplateFilename** – path to the file used as the template for notifications sent to the administrator.

  Required parameter.

  The default parameter value is:
  **/etc/opt/kaspersky/mailgw/templates/notify.tmpl** (in Linux)

  **/usr/local/etc/kaspersky/mailgw/templates/notify.tmpl** (in FreeBSD).

- **NotifyRecipientTemplateFilename** – path to the file used as the template for notifications sent to the recipient.

  Required parameter.

  The default parameter value is:
  **/etc/opt/kaspersky/mailgw/templates/notify.tmpl** (in Linux)

  **/usr/local/etc/kaspersky/mailgw/templates/notify.tmpl** (in FreeBSD)

- **NotifySenderTemplateFilename** – path to the file used as the template for notifications sent to the sender.

  Required parameter.

  The default parameter value is:
  **/etc/opt/kaspersky/mailgw/templates/notify.tmpl** (in Linux)

  **/usr/local/etc/kaspersky/mailgw/templates/notify.tmpl** (in FreeBSD).

- **PlaceholderTemplateFilename** – path to the file used as the template for replacement of attached infected objects.

  Required parameter.

  The default parameter value is:
  **/etc/opt/kaspersky/mailgw/templates/placeholder.tmpl** (in Linux)

  **/usr/local/etc/kaspersky/mailgw/templates/placeholder.tmpl** (in FreeBSD).

- **DisclaimerTemplateFilename** – path to the file used as the template for addition to appended messages.

  The default parameter value is: **/etc/opt/kaspersky/mailgw/templates/disclaimer.tmpl** (in Linux)

  **/usr/local/etc/kaspersky/mailgw/templates/disclaimer.tmpl** (in FreeBSD).

**Attention!**

Use of the template is defined by the **AddDisclaimer** option. You should modify the default template included in the distribution package to reflect the security policy of your company.

Parameters that define the paths to quarantine directories:

- **AVQuarantinePath** – path to the quarantine directory for messages containing objects that have been assigned the status **Infected**, **Disinfected**, **Suspicious**, **Protected** or **Error** while scanning. The directory may also contain objects with the status **Filtered**.

  Required parameter.

  The default parameter value is **/var/opt/kaspersky/mailgw/quarantine/av** (in Linux) or **/var/db/kaspersky/mailgw/quarantine/av** (in FreeBSD).

- **SpamQuarantinePath** – path to the quarantine directory for messages identified by the Spamtest filter as spam or probable spam.

  Required parameter.

  The default parameter value is **/var/opt/kaspersky/mailgw/quarantine/spam** (in Linux) or **/var/db/kaspersky/mailgw/quarantine/spam** (in FreeBSD).

Parameters regulating application actions after an object that has failed to pass anti-virus or anti-spam scanning has been assigned a certain status:

- **BlockDSN** – the list of statuses assigned to a message after its scanning by the anti-spam and AV components, for which the application will block generation and delivery of DSN notifications.

  Required parameter.

  The default parameter value is **infected**, **spam**.

- **BlockMessage=spam|as/spam**,**probable|as/probable**, **formal|as/formal**,**blacklisted|as/blacklisted**,**disinfected|** **av/disinfected**,**infected|av/infected**,**suspicious|av/suspicious**, **protected|av/protected**,**error|av/error**,**filtered|av/filtered**,

**av/all|as/all|all|none** – list of statuses assigned to a message after its processing by the anti-spam module, and statuses assigned to message objects according to the anti-virus scan results. Delivery of original messages with these statuses is blocked.

Required parameter.

The default parameter value is **av/disinfected**, **av/infected**, **av/suspicious**, **av/protected**, **av/error**, **av/filtered**.

**Attention!**

Each of the statuses used as values for the **BlockMessage**, **QuarantineMessage**, **NotifyAdmin**, **NotifyRecipient** and **NotifySender** parameters can be recorded in two equivalent forms: with a prefix and without one. The AV module assigns statuses preceded by the **av** prefix; the anti-spam module uses prefix **as**, for example: `BlockMessage=av/protected, as/spam, probable.`

The **as/all** status means all statuses which can be assigned by the anti-spam module. The **av/all** status means all statuses, which can be assigned by the AV module. The **all** status stands for all statuses which can be assigned both by the anti-spam module and the AV module.

- **QuarantineMessage=spam|as/spam**,**probable|as/probable**,**formal|as/formal**,**blacklisted|as/blacklisted**,**disinfected|av/disinfected**,**infected|av/infected**,**suspicious|av/suspicious**,**protected|av/protected**,**error|av/error**,**filtered|av/filtered**,**av/all|as/all|all|none** – list of statuses assigned to a message after its processing by the anti-spam module, and statuses assigned to message objects according to the anti-virus scan results. The application preserves a copy of the message in its quarantine directory.

  If an empty value is specified, the application will not quarantine messages.

  Required parameter.

  The default parameter value is empty.

**Attention!**

Messages identified as spam or probable spam, and messages containing infected, password-protected, damaged objects or objects which cannot be disinfected, are stored in separate quarantine directories. The path to the respective directories is defined by the **SpamQuarantinePath** and **AVQuarantinePath** parameters.

- **NotifyAdmin=disinfected|av/disinfected, infected|av/infected**,**suspicious|av/suspicious**,**protected|**

**av/protected**,**error|av/error**,**filtered|av/filtered**,**av/all|all|none** – list of statuses assigned to a message after its processing by the anti-spam module, and statuses assigned to message objects according to the anti-virus scan results. Messages with these statuses trigger notification delivery to the administrator.

Required parameter.

The default parameter value is **av/disinfected**, **av/infected**, **av/suspicious**, **av/protected**, **av/error**, **av/filtered**.

- **NotifyRecipient=disinfected|av/disinfected**, **infected|av/infected**,**suspicious|av/suspicious**,**protected|av/protected**,**error|av/error**,**filtered|av/filtered**,**av/all|all|none** – list of statuses assigned to a message after its processing by the anti-spam module, and statuses assigned to message objects according to the anti-virus scan results. Messages with these statuses trigger notification delivery to the original message recipient.

  Required parameter.

  The default parameter value is **av/disinfected**, **av/infected**, **av/suspicious**, **av/protected**, **av/error**, **av/filtered**.

- **NotifySender=disinfected|av/disinfected**, **infected|av/infected**,**suspicious|av/suspicious**,**protected|av/protected**,**error|av/error**,**filtered|av/filtered**,**av/all|all|none** – list of statuses assigned to a message after its processing by the anti-spam module, and statuses assigned to message objects according to the anti-virus scan results. Messages with these statuses trigger notification delivery to the original message sender.

  If an empty value is specified, no notifications will be delivered to the original message senders.

  Required parameter.

  The default parameter value is empty.

Group of parameters regulating preliminary filtering:

- **IncludeByName** – defines masks for filtering by the attachment name. The application will filter objects if their names match the specified masks, and do not match the masks used to define exclusions from scanning. Wildcards can be used for mask definition:

  * (asterisk) – an arbitrary string of characters.

  ? (question mark) – a single arbitrary character

  If the parameter is not defined, the application will not filter mail by attachment names.

You can specify several parameter values as a list:

```
IncludeByName=*exe
IncludeByName=*.bat
```

If an empty value is specified, the application will not filter mail by attachment names.

The default parameter value is empty.

- **ExcludeByName** – defines masks to exclude objects from filtering by the attachment name. Objects with names not matching these masks, and matching masks defining inclusions into scanning, will be filtered.

  The default parameter value is empty.

- **IncludeByMime** – defines masks for filtering by MIME type. The application will filter objects if their MIME type matches the specified masks, and does not match the masks defining exclusions from scanning. If this parameter is not defined, the application will not filter mail by MIME types.

  If an empty value is specified, the application will not filter mail by attachment types.

  The default parameter value is empty.

- **ExcludeByMime** – defines masks to exclude from filtering by MIME type. The application will filter objects with a MIME type not matching these masks, and matching the masks defining inclusions into scanning.

  The default parameter value is empty.

Parameters that control text addition to messages:

- **AddXHeaders=true|false** – the option defines whether the application will add the informational headers containing status assigned after anti-virus and anti-spam checks to processed messages (please refer to section B.18 on page 183 for details on the headers added to e-mail by the anti-spam module).

  If the parameter is set to **true**, the informational string will be added to message header.

  The default parameter value is **true**.

- **AddDisclaimer=true|false** – an option to add disclaimer text generated according to the template specified by the administrator in the **DisclaimerTemplateFilename** option.

  The default parameter value is **false**.

# A.15. Section *[path mailgw.group:group_name]*

The **[mailgw.group:group_name]** section contains settings for processing e-mail messages for particular groups of recipients/senders:

Parameters that define the lists of masks for addresses of mail senders and recipients:

- **Senders** – list of address masks (masks of IP and e-mail addresses) defining the senders of e-mail messages.

  Record format:

  Senders=ip <ip>

  Senders=ip <ip>/<netmask>

  Senders=ip <ip>/<netmask_cidr>

  Senders=host <hostname>

  Senders=network <network_name>

  Senders=<e-mailmask>

  where

  - **<ip>** corresponds to IP address (record format: x.x.x.x). E.g., 192.168.10.1;
  - **<hostname>** corresponds to the mask for host names. Wildcards can be used for mask definition:

    ***** (asterisk) – an arbitrary string of characters.

    ? (question mark) – a single arbitrary character.
  - **<network_name>** corresponds to a network with the appropriate name defined by the **NetworkName** option in the **[mailgw.network]** section.
  - **<e-mailmask>** - the list of masks for addresses of e-mail senders. Each mask must be specified in a separate line using the **Senders=<e-mailmask>** format. The «*» and «?» wildcards can be used in the masks (e.g., **Senders=*@example . com**). If the parameter is not defined, it is assumed to be set to «*@*» (all addresses).

The default value is empty.

- **Recipients** – list of address masks defining recipients of e-mail messages. Each mask is specified in a separate line in **Recipients=mask** format. You can use the "*" and "?" wildcards (e.g., **Recipients=*@example.com**). If this option is not defined, the value is assumed to be "***@***" (all addresses).

  The default value is empty.

> **Attention!**
>
> At least one of the **Senders** or **Recipients** parameters has to be specified.

Parameters regulating anti-virus scanning:

- **CheckAV=true|false** – defines the anti-virus scanning mode for all e-mail messages included in the particular group of recipients/senders. To disable the mode (i. e., configure the application to skip message scanning), set the option to **false**.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **AVCure=true|false** – the mode for disinfection of revealed objects. Set the parameter to **true** to enable disinfection.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **AVScanArchives=true|false** – the mode for scanning of archives. Set the parameter to **false** to disable the mode.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **AVScanMailBases=true|false** – the mode for scanning of e-mail bases. Set the parameter to **false** to disable the mode.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **AVUseBasesSet=standard|extended** – the set of AV databases in Kaspersky Mail Gateway. The **extended** set contains in addition to the records from the **standard** set also signatures of riskware, such as adware, remote administration utilities, etc.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

Parameters that define actions applied to objects scan (see section 4.4. on page 44) that have passed anti-virus scanning based on the statuses assigned after AV scan:

- **ActionDisinfected=cure|pass|remove|placeholder** – action to be applied to objects which have been disinfected successfully.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **ActionInfected=pass|remove|placeholder** – action to be applied to infected objects.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **ActionSuspicious=pass|remove|placeholder** – action to be applied to objects that are suspected of being infected with an unknown virus.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **ActionProtected=pass|remove|placeholder** – action to be applied to objects which the application failed to scan because they are password-protected.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **ActionError=pass|remove|placeholder** – action to be applied to objects which the application failed to scan because of a scan error.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **ActionFiltered=pass|remove|placeholder** – action to be applied to objects filtered by name or MIME type.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

Parameters that control the anti-spam module configuration:

- **CheckSpam=true|false** – defines the spam filtering mode for messages sent to members of the group.

- **SpamRateLimit=minimum|standard|high|maximum** – spam detection intensity. Four intensity levels are supported: **minimum**, **standard**, **high**, **maximum**. The **standard** level is recommended.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamMarkProbable=true|false** – the option enables addition of the **ProbableSpam** label to the headers of messages if scanning reveals that they belong to the corresponding category. If the parameter is set to **false**, the application will not recognize messages as probable spam.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamMarkObscene=true|false** – the option enables addition of the **Obscene** label to the headers of messages if scanning reveals that they belong to the corresponding category. If the parameter is set to **false**, the application will not recognize messages as Obscene.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

Parameters that define spam filtration criteria:

- **SpamUseDNS=true|false** – verification of information about message sender in DNS and DNS-based services: DNSBL, SPF, etc. The parameter controls the use of DNS services by the filtering module; individual services are enabled/disabled by their corresponding separate parameters.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamCheckDNSBL=true|false** – the option defines whether the application will check the sender’s IP address using the specified of DNSBL services.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamCheckHostInDNS=true|false** – the option defines whether the application will check the presence of the sender’s IP address in DNS.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamCheckSPF=true|false** – the option defines whether the application will check the existence of the sender’s IP address using SPF (Sender Policy Framework).

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamCheckSURBL=true|false** – the option defines whether the application will check the existence of the sender’s IP address using SURBL (Spam URL Realtime Blocklists).

If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamHeadersToUndiclosed=true|false** – the option defines whether the application will check message headers for the presence of undisclosed recipients lists.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamHeadersFromOrToDigits=true|false** – the option defines whether the application will check sender or recipient addresses for the presence of groups of digits. You are advised to set the parameter to **true**, if your e-mal addresses do not contain groups of digits.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamHeadersFromOrToNoDomain=true|false** – the option defines whether the application will check the address for the missing domain part. You are advised to set the parameter to **false** for recipients that allow delivery of e-mail with incomplete addresses.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamHeadersSubjectTooLong=true|false** – the option defines whether the application will check message subject for the presence of too long text strings. You are advised to set the parameter to **false** if such mail is allowed in your e-mail system.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamHeadersSubjectWSOrDots=true|false** – the option defines whether the application will check message header for multiple spaces and dots. You are advised to set the parameter to **false** if such mail is allowed in your e-mail system.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamHeadersSubjectDigitIDOrTimestamp=true|false** – the option defines whether the application will check message header for the presence of a digital identifier or time label.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamHeadersMarkAllChinese=true|false** – the option defines whether the application will check message header for the presence of words in the Chinese language.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamHeadersMarkAllKorean=true|false** – the option defines whether the application will check message header for the presence of words in the Korean language.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamHeadersMarkAllThai=true|false** – the option defines whether the application will check message header for the presence of words in the Thai language.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamHeadersMarkAllJapanese=true|false** – the option defines whether the application will check message header for the presence of words in the Japanese language.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **MarkSubject=spam|probable|formal|blacklisted** – the option adds to message header a prefix describing its status assigned to the letter by the anti-spam module after scanning (see section 4.2 on page 36).

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamCheckSizeLimit** – the option defines maximum size (Kb) of messages checked for the presence of spam. The anti-spam module will not scan larger messages.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

Parameters that define the addresses of notification sender and recipient:

- **AdminNotifyAddress** – e-mail address to which the application will send notifications for the administrator, regarding processing results for messages included in this group.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **NotifyFromAddress** – e-mail address from which the application will send notifications regarding the scan results for messages of this group.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

Parameters that define the paths to template files:

- **NotifyAdminTemplateFilename** – path to the file used as the template for notifications sent to the administrator.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **NotifyRecipientTemplateFilename** – path to the file used as the template for notifications sent to the recipient.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **NotifySenderTemplateFilename** – path to the file used as the template for notifications sent to the sender.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **PlaceholderTemplateFilename** – path to the file used as the template for replacement of attached infected objects.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **DisclaimerTemplateFilename** – path to the file used as the template for addition to appended messages.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

**Note**

Use of the template is defined by the **AddDisclaimer** option. You should modify the default template included in the distribution package to reflect the security policy of your company.

Parameters that define the paths to quarantine directories:

- **AVQuarantinePath** – path to the quarantine directory for messages containing objects that have been assigned the i**nfected**, **disinfected**, **suspicious**, **protected** or **error** status while scanning.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **SpamQuarantinePath** – path to the quarantine directory for messages identified by the Spamtest filter as spam or probable spam.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

Parameters regulating application actions after an object that has failed to pass anti-virus or anti-spam scanning has been assigned a certain status:

- **BlockDSN** – the list of statuses assigned to a message after its scanning by the anti-spam and AV components, for which the application will block generation and delivery of DSN notifications.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **BlockMessage=spam|as/spam**,**probable|as/probable**, **formal|as/formal**,**blacklisted|as/blacklisted**,**disinfected| av/disinfected**,**infected|av/infected**,**suspicious|av/suspicious**, **protected|av/protected**,**error|av/error**,**filtered|av/filtered**, **av/all|as/all|all|none** – list of statuses assigned to a message after its processing by the anti-spam module, and statuses assigned to message objects according to the anti-virus scan results. Delivery of original messages with these statuses is blocked.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **NotifyAdmin=disinfected|av/disinfected, infected|av/infected**,**suspicious|av/suspicious**,**protected| av/protected**,**error|av/error**,**filtered|av/filtered**,**av/all|all|none** – list of statuses assigned to a message after its processing by the anti-spam module, and statuses assigned to message objects according to the anti-virus scan results. Messages with these statuses trigger notification delivery to the administrator.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **NotifyRecipient=disinfected|av/disinfected**, **infected|av/infected**,**suspicious|av/suspicious**,**protected| av/protected**,**error|av/error**,**filtered|av/filtered**,**av/all|all|none** – list of statuses assigned to a message after its processing by the anti-spam module, and statuses assigned to message objects according to the anti-virus scan results. Messages with these statuses trigger notification delivery to the original message recipient.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **NotifySender=disinfected|av/disinfected**, **infected|av/infected**,**suspicious|av/suspicious**,**protected| av/protected**,**error|av/error**,**filtered|av/filtered**,**av/all|all|none** – list of statuses assigned to a message after its processing by the anti-spam module, and statuses assigned to message objects according to the anti-virus scan results. Messages with these statuses trigger notification delivery to the original message sender.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **QuarantineMessage=spam|as/spam**,**probable|as/probable**,**formal| as/formal**,**blacklisted|as/blacklisted**,**disinfected|av/disinfected**, **infected|av/infected**,**suspicious|av/suspicious**,**protected| av/protected**,**error|av/error**,**filtered|av/filtered**,**av/all|as/all|all|none** – list of statuses assigned to a message after its processing by the anti-spam module, and statuses assigned to message objects according to the anti-virus scan results. The application preserves a copy of the message in its quarantine directory.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

Group of parameters regulating preliminary filtering:

- **IncludeByName** – defines masks for filtering by attachment name. The application will filter objects if their name matches the specified masks, and does not match the masks defining exclusions from scanning. If this parameter is not defined, the application will not filter e-mail by attachment names. You can specify several parameter values as a list:

  ```
  IncludeByName=*exe
  IncludeByName=*.bat
  ```

- **IncludeByMime** – defines masks for filtering by the MIME type of attachments. The application will filter objects if their MIME types match the specified masks, and do not match the masks used to define exclusions from scanning. If the parameter is not defined, the application will not filter e-mail by the attachment types.

- **ExcludeByName** – defines masks to exclude objects from filtering by the attachment name. Objects with names not matching these masks, and matching masks defining inclusions into scanning, will be filtered. If an empty value is specified, the application will not filter mail by attachment names.

- **ExcludeByMime** – defines masks to exclude from filtering by MIME type. The application will filter objects with a MIME type not matching these masks, and matching the masks defining inclusions into scanning.

If an empty value is specified, the application will not filter mail by attachment types.

Parameters that control text addition to messages:

- **AddXHeaders=true|false** – the option defines whether the application will add the X-SpamTest-* informational header to processed messages (please refer to section B.18 on page 183 for details on the headers added to e-mail by the anti-spam module, see section 6.12 on page 93 for examples of headers assigned after anti-virus scanning).

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

- **AddDisclaimer=true|false** – an option to add disclaimer text generated according to the template specified by the administrator in the **DisclaimerTemplateFilename** option.

  If the parameter is not defined, the corresponding parameter value from the **[mailgw.policy]** section will be used.

# A.16. Section *[updater.path]*

The **[updater.path]** section contains the paths to directories and files required for the *keepup2date* component to work:

- **BackUpPath** – path to the directory that stores the archive of the anti-virus and anti-spam databases during updating. Required parameter.

  The default parameter value is:
  **/var/opt/kaspersky/mailgw/bases.backup** (in Linux) and
  **/var/db/kaspersky/mailgw/bases.backup** (in FreeBSD).

- **PidFile –** pid file path. The parameter is used to prevent simultaneous launch of several instances of the *keepup2date* component. If the parameter is missing, no pid file will be created and the application will not perform checks for the presence of other running instances of the component.

  The default parameter value is **/var/run/mailgw** (in Linux and FreeBSD).

- **AVBasesTestPath** – complete path to the *avbasestest* utility validating the anti-virus databases. The application uses the utility immediately after downloading updates. It will copy them from a temporary directory to the storage directory only if the retrieved updates are not corrupted. If the parameter is not specified, the application will display during update a console notification informing that the anti-virus databases could not be validated and the updates are installed without being checked. This message will also be appended to the log.

The default parameter value is **/opt/kaspersky/mailgw/lib/bin** (in Linux) and **/usr/local/libexec/kaspersky/mailgw** (in FreeBSD).

> **Attention!**
>
> The *avbasestest* utility will run automatically; it requires no user participation.

# A.17. Section *[updater.options]*

The **[updater.options]** section contains various parameters of the *keepup2date* component operation:

- **KeepSilent=true|false** – defines console display options for reports of the component work. If set to **true**, the reports are not output to console.

  The default parameter value is **false**.

- **PostUpdateCmd** – defines the command that will be executed immediately after updating of the anti-virus and anti-spam databases is successfully completed. The parameter's default value will automatically reload the updated anti-virus database and the anti-spam module. You are advised not to change the value of this parameter.

- **UseUpdateServerUrl=true|false** – an option making the application use the URL specified by the **UpdateServerUrl** parameter to update the database.

  The default parameter value is **false**.

- **UseUpdateServerUrlOnly=true|false** – an option making the application use **only** the URL specified by **UpdateServerUrl** to update the database. If this option is set to **false**, then whenever updating from the **UpdateServerUrl** address fails, the application will use a different address from the list of update servers.

  The default parameter value is **true**.

- **UpdateServerUrl=http://url/** | **ftp://url/** | **/local_path/** – defines the path to be used to update the anti-virus and anti-spam databases.

- **RegionSettings** – defines the customer region used to update the anti-virus and content filtration databases from the nearest Kaspersky Lab's update server.

  The default parameter value is **com**.

  To see a complete list of regions, run the *keepup2date* utility with the *–s* key (please see section B.11 on p. 168).

- **ConnectTimeout** – timeout (seconds) for network operations during an update of the anti-virus and anti-spam databases.

  The default parameter value is **30**.

- **UseProxy=true|false** – use a proxy-server to connect to one of Kaspersky Lab's update servers. If the parameter value is **false** a proxy server will not be used. If the parameter value is **true**, the proxy server address defined by the **ProxyAddress** parameter, will be used. If the value of the **ProxyAddress** parameter is not defined, the value of **http_proxy** environment variable will be used. If the value of environment variable is not defined, a proxy server will not be used.

- **ProxyAddress** – the address of the proxy server, used for connection. The parameter is specified as: **http://username:password@url:port**. **Username** and/or **password** can be omitted in proxy server address. If the address is not defined, its value will be taken from **http_proxy** environment variable.

- **PassiveFtp=true|false** – use passive FTP mode to download updates.

  The default parameter value is **true**.

- **UpdateComponentsList** – list of components, which will be updated. E.g.:

  - **UpdateComponentsList=KAS303, AVS, AVS_OLD, CORE, Updater, BLST** – download application databases.

  - **UpdateComponentsList=AVS, AVS_OLD, CORE, Updater, BLST** – download the anti-virus databases only (e.g., if your licensing policy implies just anti-virus mail scanning – see section 1.2 on page 11).

- **RetranslateComponentsList** – list of updates for which updating will be performed through copying of updates (see section 5.1 on page 46).

  If the value of the parameter is not defined, the value of **UpdateComponentsList** parameter will be used.

# A.18. Section *[updater.report]*

The **[updater.report]** section contains report output options for the *keepup2date* component:

- **ReportFilename** – name of the log file that will store the component's performance report. If the option is set to **syslog**, the report is saved to the system log.

  The default value is: **TEMP_PATH/mailgw-keepup2date.log**, where **TEMP_PATH** stands for the value of the **TMP** environment variable; if **TMP** is not defined, for the value of the **TEMP** variable, and if **TEMP** is not defined, for the **/tmp** directory.

- **ReportLevel=0|1|2|3|4|9** – the level of details in component performance report (**0** – Fatal, **1** – Error, **2** – Warning, **3** – Info, **4** – Activity, **9** – Debug).

  The default parameter value is **4**.

- **Append=true|false** – append a new component performance report to the end of the existing report file. If this option is set to **false**, the previous report will be overwritten by the new report.

  The default parameter value is **true**.

# APPENDIX B. SUPPLEMENTARY INFORMATION ABOUT THE PRODUCT

This annex gives detailed information about the default locations of the application files after installation; command line parameters for every component and their return codes, and generation of operational statistics.

## B.1. Distribution of the application files in directories

The default locations of the Kaspersky Mail Gateway files are as follows:

**Linux distribution kit:**

*/opt/kaspersky/mailgw/bin/* – directory containing executable files of the application components:

- *mailgw-keepup2date* – executable file of the component updating the anti-virus and anti-spam databases of Kaspersky Mail Gateway;
- *mailgw-licensemanager* – executable file of the component managing license keys;
- *mailgw-tlv* – utility for template syntax checks;
- *mailgw-mailq* – utility managing the application's working queue;
- *mailgw-maila* – utility for managing the message archives.

*/opt/kaspersky/mailgw/sbin/* – directory containing executable flies of daemons:

- *mailgwd* – main application component including its AV module;
- mailgw-*kas-license, mailgw-process-server, mailgw-spfd* – anti-spam module daemons.

*/opt/kaspersky/mailgw/lib/bin/* – directory containing executable files and scripts necessary for application functioning:

- *kas-filter/* – directory containing the files of the anti-spam module;
- setup/ – directory containing scripts and executable files used during the installation, post-install setup and removal of the application;

- *mailgw* – the script used to control the application. A link to the controlling script is also added to the */etc/init.d/* directory;
- *avbasestest* – utility validating downloaded updates to the updates for Kaspersky Mail Gateway databases prior to their installation.

*/opt/kaspersky/mailgw/share/doc/* – directory containing license files.

*/opt/kaspersky/mailgw/share/man/* – directory containing application manual pages.

*/etc/opt/kaspersky/* – directory containing the *mailgw.conf* default application configuration file.

*/etc/opt/kaspersky/mailgw/templates/* – directory containing the default application template files:

- *notify.tmpl* – template for notification messages;
- *placeholder.tmpl* – template used for replacing an infected object with a message;
- *dsn.tmpl* – template used for DSN messages generated by the application;
- *disclaimer.tmpl* – template used to generate the disclaimer text appended to e-mail messages.

*/var/opt/kaspersky/mailgw/* – directory containing application data modified during operation and including the following subdirectories and files:

- *bases/* – directory containing the databases of Kaspersky Mail Gateway and the *updcfg.xml* configuration file of the *keepup2date* component;
- *bases.backup/* – directory where the *keepup2date* component saves backup copies of Kaspersky Mail Gateway databases and anti-spam databases;
- *licenses/* – directory containing license key files;
- *quarantine/av/* – directory used by the application to store messages that have been assigned after anti-virus scanning the statuses specified in the **QuarantineMessage** parameter of the application's configuration file;
- *quarantine/spam/* – directory used by the application to store messages that have been assigned after anti-spam checks the statuses specified in the **QuarantineMessage** parameter of the application's configuration file;
- *arch_in/* – directory for the archive of all received e-mail messages;
- *arch_out/* – directory for the archive of all sent e-mail messages;

- *arch_spool/* – directory for the archive of e-mail messages, which cannot be processed;
- *stats/* – directory for the statistics files;
- *stats/webmin/* – directory for statistics files required for the operation of Webmin plug-in;
- *kas-filter /* – directory containing the files required for operation of the anti-spam filter.

*/var/spool/kaspersky/mailgw/* – directory used by the application to store the working queue of messages.

*/var/log/kaspersky/mailgw/* – directory for log files.

*/var/run/mailgw/* – directory for storing PID files and sockets.

**FreeBSD distribution kit:**

*/usr/local/bin/* – directory containing executable files of the application components:

- *mailgw-keepup2date* – executable file of the component updating the anti-virus and anti-spam databases of Kaspersky Mail Gateway;
- *mailgw-licensemanager* – executable file of the component managing license keys;
- *mailgw-tlv* – utility for template syntax checks;
- *mailgw-mailq* – utility managing the application's working queue;
- *mailgw-maila* – utility for managing the message archives.

*/usr/local/sbin/* – directory containing executable flies of daemons:

- *mailgwd* – main application component including its AV module;
- *mailgw-kas-license, mailgw-process-server, mailgw-spfd* – anti-spam module daemons.

*/usr/local/libexec/kaspersky/mailgw/* – directory containing executable files and scripts necessary for application functioning:

- *kas-filter/* – directory containing the files of the anti-spam module;
- setup/ – directory containing scripts and executable files used during the installation, post-install setup and removal of the application;
- *avbasestest* – utility validating downloaded updates to the updates for Kaspersky Mail Gateway databases prior to their installation.

*/usr/local/share/doc/mailgw/* – directory containing license files.

/usr/local/etc/rc.d/mailgw.sh – the script used to control the application.

*/usr/local/man/* – directory containing application manual pages.

*/usr/local/etc/kaspersky/* – directory containing the *mailgw.conf* default application configuration file.

*/usr/local/etc/kaspersky/templates/* – directory containing the default application template files:

- *notify.tmpl* – template for notification messages;
- *placeholder.tmpl* – template used for replacing an infected object with a message;
- *dsn.tmpl* – template used for DSN messages generated by the application;
- *disclaimer.tmpl* – template used to generate the disclaimer text appended to e-mail messages.

*/var/db/kaspersky/mailgw/* – directory containing application data modified during operation and including the following subdirectories and files:

- *bases/* – directory containing the databases of Kaspersky Mail Gateway and the *updcfg.xml* configuration file of the *keepup2date* component;
- *bases.backup/* – directory where the *keepup2date* component saves backup copies of Kaspersky Mail Gateway anti-virus and anti-spam databases;
- *licenses/* – directory containing license key files;
- *quarantine/av/* – directory used by the application to store messages that have been assigned after anti-virus scanning the statuses specified in the **QuarantineMessage** parameter of the application's configuration file;
- *quarantine/spam/* – directory used by the application to store messages that have been assigned after anti-spam checks the statuses specified in the **QuarantineMessage** parameter of the application's configuration file;
- *arch_in/* – directory for the archive of all received e-mail messages;
- *arch_out/* – directory for the archive of all sent e-mail messages;
- *arch_spool/* – directory for the archive of e-mail that cannot be processed;
- *stats/* – directory for the statistics files;

- *stats/webmin/* – directory for statistics files required for the operation of Webmin plug-in;
- *kas-filter /* – directory containing the files required for operation of the anti-spam filter.

*/var/spool/kaspersky/mailgw/* – directory used by the application to store the working queue of messages.

*/var/log/kaspersky/mailgw/* – directory for log files.

*/var/run/mailgw/* – directory for PID files and sockets.

# B.2. Use of external configuration files

You may connect external configuration files to the main configuration file, using any of the following methods:

- Using the **include** directive.
- Using a record of the form: **file:file_name** as the parameter value.

External configuration files are connected using the **include** directive added into an arbitrary location in the configuration file as a line that looks like:

```
!include <file_name>
```

where **<file_name>** is the absolute path to the specified external configuration file; the file must exist and be available for reading.

Include files are useful, for example, to specify all the parameters for a certain group of users in a separate file. In this case, modifying the settings for that group would require modification of that file only. You will not have to change the main configuration file.

External configuration files are connected using the **file:file_name** record added as a parameter that looks like:

```
Senders=file:<file_name>
RelayRule=deny for file:<file_name> to *
```

where **<file_name>** is the absolute path to the file; the file must exist and be available for reading.

In this case, the application will substitute the contents of the external file, line by line, for the **file:** construction; the result of the substitution will be identical to assigning the same number of values to that parameter.

**Attention!**

The application also substitutes empty strings from the external file. Therefore such strings are unwanted, since their presence can cause errors in the syntax of the application configuration file.

Example:

Ensure that the application uses parameters specified in an external file to control client access.

To perform the task, you should do the following:

1. Create a *my-recipients.list* text file containing a list of addresses for users who should receive the e-mail, using the following format:

```
localpart1@my.domain
localpart2@my.domain
...
localpartN@my.domain
```

2. Assign the following value for the **RelayRule** parameter in the configuration file:

```
[mailgw.access]
RelayRule=allow for any to file:<absolute file
path>/my-recipients.list
RelayRule=deny from any to *
```

or, to enable transfer of both incoming and outgoing e-mail for those addresses:

```
[mailgw.access]
RelayRule=allow for any to file:<absolute file
path>/my-recipients.list
RelayRule=allow for file:<absolute file path>/my-
recipients.list to *
RelayRule=deny for any to *
```

**Attention!**

External files cannot be used to define parameters in the **[updater.*]** sections or the **[path]** or **[locale]** sections containing common parameters.

**Attention!**

If the application is being remotely controlled via Webmin module, the use of external configuration files is not supported and may cause incorrect application behaviour.

## B.3. Control signals for the main application daemon

You can manipulate the application using the **TERM (15), QUIT (3), INT (2)** special control signals, which terminate application activity.

## B.4. Command line application management

You can manage the application from the command line. Command syntax:

In Linux:

```
# /opt/kaspersky/mailgw/sbin/mailgwd -x <command>
```

In FreeBSD:

```
# /usr/local/sbin/mailgwd -x <command>
```

Used commands are listed in the table below.

| | |
|---|---|
| **stats** | Display application status statistics. |
| **recv-on** | Start the Receiver module. |
| **recv-off** | Stop the Receiver module. |
| **check-on** | Start the scanning module. |
| **check-off** | Stop the scanning module. |
| **send-on** | Start the Sender module. |
| **send-off** | Stop the Sender module. |
| **reload-db** | Application restart with anti-virus database reloading. |

To initiate an action, specify the corresponding command as a command line option. The entered command will be passed on through the application socket **ControlSocket** (**[mailgw.path]** section) to the *mailgw* component, which will perform it.

# B.5. Application statistics

Following the **stats** command of the control script (see section 6.8 on p. 88) the application writes its performance statistics (from application startup to the current moment) to the text file specified by the **StatFilename** option in the **[mailgw.options]** section.

This txt file contains a set of lines in the following format:

```
parameter_name=parameter_value
```

The table below lists the names and values of the application status parameters.

| **Parameter name**<br>Parameter value |
|---|
| **time_initialized**<br>Time when the server was initialized (in unix time format). |
| **time_initialized_iso**<br>Time when the server was initialized (in ISO8601 format). |
| **time_processing**<br>Server operation time (seconds). |
| **mta_received_messages**<br>Number of incoming messages successfully received by the server since its initialization. |
| **mta_received_bytes**<br>Number of bytes successfully received by the server since its initialization. |
| **mta_received_recipients**<br>Number of different recipients of incoming messages successfully received by the server since its initialization. |

| **Parameter name** | Parameter value |
|---|---|
| **mta_sent_messages** | Number of outgoing messages successfully sent by the server since its initialization. |
| **mta_sent_bytes** | Number of bytes successfully sent by the server since its initialization. |
| **mta_sent_recipients** | Number of different recipients of outgoing messages successfully sent by the server since its initialization. |
| **mta_stored_messages_current** | Number of queued messages at the time the report was generated. |
| **mta_incoming_connections_total** | Number of established incoming connections to the server since its initialization. |
| **mta_incoming_connections_current** | Number of established incoming connections to the server at the time the report was generated. |
| **mta_incoming_connections_maximum** | Maximum number of incoming connections to the server since its initialization. |
| **mta_incoming_connections_errors** | Number of incoming connection errors since server initialization. |
| **mta_incoming_connections_refused_total** | Total number of rejected incoming connections to the server since its initialization. |
| **mta_incoming_connections_refused_for_relaying** | Total number of incoming connections rejected by the server because of the relaying rules, since server initialization. |

| **Parameter name**<br>Parameter value |
|---|
| **mta_incoming_connections_refused_for_connections_limit**<br>Number of incoming connections rejected by the server because of the limit on the number of simultaneous incoming connections, since server initialization. |
| **mta_incoming_connections_refused_for_connections_per_ip_limit**<br>Number of incoming connections rejected by the server due to the limit on the number of simultaneous incoming connections from a single IP address, since server initialization. |
| **mta_outgoing_connections_total**<br>Number of outgoing connections from the server since its initialization. |
| **mta_outgoing_connections_current**<br>Number of simultaneous outgoing connections at the time the report was generated. |
| **mta_outgoing_connections_maximum**<br>Maximum number of outgoing connections from the server, since server initialization. |
| **mta_outgoing_connections_errors**<br>Number of outgoing connection errors, since server initialization. |
| **mta_outgoing_connections_failed_total**<br>Total number of rejected outgoing connections from the server since its initialization. |
| **mta_outgoing_connections_failed_through_cache**<br>Total number of rejected outgoing connections to inaccessible servers. |
| **mta_routing_queries_total**<br>Total number of routing queries since server initialization. |
| **mta_dns_queries_total**<br>Total number of DNS queries after server initialization. |

| **Parameter name**<br>Parameter value |
|---|
| **mta_dns_queries_failed**<br>Total number of rejected DNS queries after server initialization. |
| **mta_ incoming_sessions_refused_total**<br>Total number of incoming connections rejected by the server since its initialization. |
| **mta_ incoming_sessions_refused_for_message_size_limit**<br>Total number of incoming connections rejected by the server because of the message size limit, since server initialization. |
| **mta_ incoming_sessions_refused_for_session_size_limit**<br>Number of incoming messages rejected by the server because of the session size limit. since server initialization. |
| **mta_ incoming_sessions_refused_for_hops_limit**<br>Number of incoming messages rejected by the server because of the limit on the number of hops, since server initialization. |
| **mta_ incoming_sessions_refused_for_messages_per_session_limit**<br>Number of incoming messages rejected by the server because of the limited number of messages per session, since server initialization. |
| **mta_ outgoing_sessions_failed_total**<br>Total number of rejected outgoing messages, since server initialization. |
| **mta_ outgoing_sessions_failed_for_message_size_limit**<br>Number of outgoing messages rejected because of the size limit, since server initialization. |
| **mta_ outgoing_sessions_failed_for_8bitmime**<br>Number of outgoing messages rejected because the remote server does not support 8BITMIME SMTP protocol extension, since server initialization. |
| **mta_malformed_messages**<br>Number of malformed incoming messages received since server initialization. |

| **Parameter name** | Parameter value |
|---|---|
| **mta_dsn_generated** | Number of generated DSN messages since server initialization. |
| **antispam_ sessions_current** | Number of anti-spam sessions at the time of report creation. |
| **antispam_ sessions_maximum** | Maximum number of anti-spam sessions since server initialization. |
| **antispam_checked_messages_total** | Total number of messages processed by the spam filter since server initialization. |
| **antispam_checked_messages_spam** | Total number of messages identified as spam since server initialization. |
| **antispam_checked_messages_probable_spam** | Total number of messages identified as probable spam since server initialization. |
| **antispam_checked_messages_formal** | Total number of messages recognized as formal since server initialization. |
| **antispam_checked_messages_blacklisted** | Total number of messages recognized as mail from blacklisted senders since server initialization. |
| **antispam_checked_messages_blocked** | Total number of messages blocked as a result of spam filtration since server initialization. |
| **antivirus_ sessions_current** | Number of anti-virus sessions at the time the report was generated. |
| **antivirus_sessions_maximum** | Maximum number of simultaneous anti-virus scanning sessions since server initialization. |

| **Parameter name**<br>Parameter value |
|---|
| **antivirus_checked_objects_total**<br>Total number of objects processed by the anti-virus scanner since server initialization. |
| **antivirus_checked_objects_infected**<br>Number of infected objects which the application failed to cure since server initialization. |
| **antivirus_checked_objects_disinfected**<br>Number of objects disinfected since server initialization. |
| **antivirus_checked_objects_suspicious**<br>Number of suspicious objects detected since server initialization. |
| **antivirus_checked_objects_protected**<br>Number of protected objects not subject to anti-virus scanning since server initialization. |
| **antivirus_checked_objects_filtered**<br>Number of filtered objects since server initialization. |
| **antivirus_checked_objects_error**<br>Number of object scanning errors that occurred since server initialization. |
| **antivirus_checked_messages_total**<br>Total number of e-mail messages processed by anti-virus scanner since server initialization. |
| **antivirus_checked_messages_blocked**<br>Total number of messages blocked after anti-virus scanning, since server initialization. |
| **antivirus_checked_messages_modified**<br>Number of messages modified following anti-virus scanning, since server initializa- |

| **Parameter name**<br>Parameter value |
|---|
| tion. |
| **antivirus_notifications_generated_for_sender**<br>Number of sender notifications generated since server initialization. |
| **antivirus_notifications_generated_for_recipients**<br>Number of recipient notifications generated since server initialization. |
| **antivirus_notifications_generated_for_admin**<br>Number of administrator notifications generated since server initialization. |
| **task_sender_module_running**<br>Status of the Sender module:**0** - stopped, **1** - running. |
| **task_receiver_ module_running**<br>Status of the Receiver module: **0** - stopped, **1** - running. |
| **task_scanning_module_running**<br>Status of the scanning module: **0** - stopped, **1** - running. |
| **antivirus_bases_loaded**<br>Date of the last update of Kaspersky Mail Gateway databases in unix time format. |
| **antivirus_bases_loaded_iso**<br>Date and time of the last update of Kaspersky Mail Gateway databases in ISO8601 format. |
| **antivirus_bases_record**<br>Number of records in Kaspersky Mail Gateway databases. |
| **antivirus_bases_released_date**<br>Release date and time of Kaspersky Mail Gateway databases (in ISO8601 format). |

# B.6. SNMP traps for interaction with the application via SNMP

Administrator can configure the application to send SNMP traps when certain events occur. Generation of SNMP traps is regulated by the **SNMPTraps** option in the **[mailgw.snmp]** section of the configuration file. The list below contains SNMP traps and events triggering their generation.

| **Parameter name**<br>Parameter meaning |
|---|
| **trap_starting**<br>Send SNMP trap if application is starting. |
| **trap_started**<br>Send SNMP trap if the application has been started. |
| **trap_stopping**<br>Send SNMP trap if the application is stopping. |
| **trap_stopped**<br>Send SNMP trap if the application has been stopped. |
| **trap_config_reloading**<br>Send SNMP trap if the application configuration is being reloaded at the moment. |
| **trap_config_reloaded**<br>Send SNMP trap if application configuration has been reloaded. |
| **trap_bases_reloading**<br>Send SNMP trap if Kaspersky Mail Gateway databases are being reloaded at the moment. |
| **trap_bases_reloaded**<br>Send SNMP trap if Kaspersky Mail Gateway databases have been reloaded. |
| **trap_error**<br>Send SNMP trap if a critical error has occurred in the application activity. |

# B.7. *Mailgwd* command line options

The configuration file's parameter values can be redefined using command line options, if the application is launched from the command line. This section lists the command line parameters.

| | |
|---|---|
| **Help options** | |
| **-h**<br>or **--help** | Display on the console a summary of the command line options supported by the component, and then exit. |
| **-v**<br>or **--version** | Display the application version on the console, and exit. |
| **Configuration options** | |
| **-c <path_to _file>**<br>or<br>**--conf-file=<path_to _file>** | Use the alternative configuration file **<path_to _file>**. |
| **-d**<br>or **–no-daemon** | Do not run the component as a daemon process. |
| **-p <path_to _file>**<br>or<br>**--pid-file=<path_to_ file>** | Use the alternative PID file **<path_to _file>**. |
| **-n**<br>or **-no-pid-file** | Do not use the PID file. |
| **-o**<br>or<br>**--no-change-owner** | Do not change the user-owner of the process. |
| **-w**<br>or | Do not launch the **watchdog** process. |

| | |
|---|---|
| **--no-watchdog** | |
| **-i <time>**<br>or<br>**--wd-init-timeout=<time>** | Timeout for the **watchdog** process to wait for successful application launch (seconds); the range of supported values is 60 – 2400, default value is 600). |
| **-b <time>**<br>or<br>**--wd-headrtbeat-timeout=<time>** | Timeout for the **watchdog** process to wait for a signal indicating a successful operation by an application component (seconds); the range of supported values is 60 – 1200, default value is 600). |
| **-y <time>**<br>or<br>**--wd-heartbeat-delay=<time>** | Interval between sending application messages to the **watchdog** process, indicating a successful operation by an application components (seconds); the range of supported values is 30 – 600, default value is 180). |
| **-k**<br>or<br>**--check-config** | Check configuration file syntax. |
| **-x**<br>or<br>**--exec** | Change current running tasks. |
| **-s**<br>or<br>**--state** | Define task launch at application start. |

# B.8. *Mailgwd* return codes

The *mailgwd* component may return any of the following codes on exiting:

| | |
|---|---|
| **0** | The component started successfully. |
| **1** | Error in command line options. |

| | |
|---|---|
| **30** | A critical system error occurred during the application operation. |
| **42** | The PID file cannot be created. |
| **43** | Unable to run the daemon process for the application. |
| **44** | The UID and GID of the owner of the process cannot be changed. |
| **45** | The signal handler cannot be identified. |
| **48** | Error while initializing the anti-virus kernel. |
| **49** | Error initializing the debug information display (trace) module. |
| **50** | Error loading the anti-virus databases. |
| **51** | The anti-virus database date stamp is not within the key validity period. |
| **55** | Error matching the network name with the socket (bind). |
| **64** | License key data is missing or no license key was found using the path specified in the configuration file. |
| **65** | The configuration file cannot be loaded. |
| **66** | Error in the configuration file. |
| **67** | Error while initializing the log file. |
| **70** | Component executable file is corrupted. |
| **71** | Error during operations with the application management socket. |

# B.9. *Licensemanager* command line options

| **Help options** | |
|---|---|
| **-h** | Display on the console a summary of the component's command line options, and exit. |
| **Options for managing the keys** | |
| **-s** | Output information about all installed license keys to the console. |
| **-c (C) <path_to_file>** | Use the alternative configuration file **<path_to_file>**. |
| **-k<keyfile>** | Output to the console information about the key. |
| **-a <path_to_file>** | Install a key. |
| **-d <a\|r>** | Delete both the current and the additional key \| additional key. |

# B.10. *Licensemanager* return codes

The *licensemanager* component may return any of the following codes on exiting:

| | |
|---|---|
| **0** | The component has successfully completed its operation. |
| **30** | Critical system error. |
| **64** | Licensing error. |
| **65** | Error reading the configuration file. |
| **66** | Error in command line options. |
| **70** | The component executable file is corrupted. |

# B.11. *Keepup2date* command line options

| | |
|---|---|
| **Help options** | |
| **-h** | Display on the console a summary of the component’s command line options, and exit. |
| **-v** | Display the application version on the console and exit. |
| **-s** | Display a list of update servers with information about their respective regions. |
| **Update options** | |
| **-c <path_to_file>** | Use the alternative configuration file **<path_to_file>**. |
| **-u <directory>** | Copy the application update to the local **<directory>**. Within the specified directory, the utility will reproduce the internal structure of an update server, enabling local computers to update from that directory. |
| **-x <directory>** | Copy updates for all products of Kaspersky Lab to the local **<directory>**. Within the specified directory, the utility will reproduce the internal structure of an update server, enabling local computers to update from that directory.. |
| **-b <path>** | When updating, create a backup copy of the current anti-virus and anti-spam databases in the **<path>** directory. |
| **-t <path>** | Use the **<path>** directory to store temporary files. |
| **-r** | Rollback the last update. Updated databases will be replaced by their previous versions. |
| **-k** | Disable execution of the command defined by the **PostUpdateCmd** parameter. |

| | |
|---|---|
| **-d <path_to_file>** | Use the specified PID file. |
| **-g <url>** | Use the server with the specified URL as the source of updates. |
| **Report generation options** | |
| **-l <path_to_file>** | Log work results in the file **<path_to_file>**. |
| **-q** | Disable the output of runtime messages produced by the utility. |
| **-e** | Display critical error messages only. |

# B.12. *Keepup2date* return codes

The *keepup2date* component may return any of these codes on exiting:

| | |
|---|---|
| **0** | The anti-virus and content filtration databases do not need an update. |
| **1** | The anti-virus and content filtration databases were updated successfully. |
| **10** | A critical error occurred; updating was interrupted. |
| **12** | An error occurred while rolling back to the previous version of the anti-virus databases. Rollback has been interrupted. |
| **30** | The **PostUpdaterCmd** command could not be executed after the databases were updated or the command was completed with errors. |
| **60** | License key information is missing, or no key was found using the path specified in the configuration file. |
| **75** | The configuration file either cannot be loaded or contains errors. |
| **128+signal code** | The application has exited upon receiving a signal with the corresponding code. |

# B.13. Templates

Kaspersky Mail Gateway provides an opportunity for creation of custom notification templates for administrators, recipients and senders using a special language for notifications.

Further we examine in more detail all components of the language, its syntax and some examples.

## B.13.1. Templates language

### B.13.1.1. Iteration constructions

*An iteration construction is a basic element* of the notifications language used for generation of notification templates.

Construction syntax:

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

where:

- `<FOR` – beginning of construction definition The `<` character, which does not belong to the definition, must be isolated (please refer to section B.13.1.4 on page 174 for details).
- `INAME` – construction name in **1*(nchar)*(nchar)** format; maximum name length is 64 bytes.
- `IOP` – format comparison operation **==**, **|**, **!=**; length^ 2 bytes.
- `IVALUE` – construction value in **1*(vchar)*(vchar)** format, maximum length is 4096 bytes. Iteration construction value must be specified in quotes. If construction value is compared with a value containing a quotation mark, an isolating (escape) character has to be used (see section B.13.1.4 on page 174). E.g.:

  **<FOR** *_macro_name_parent_* == "\"_value_1\""**>**
- `>` – the end of definition of an iteration construction, beginning of definition for an iterator body. The `>` character, which does not belong to the definition, must be isolated (see section B.13.1.4 on page 174).
- `BODY` – iterator body in ***(char)** format.

- `</FOR>` – end of definition for the iterator body. The < character, which does not belong to the end of iterator body definition, must be isolated (see section B.13.1.4 on page 174).
- … – delimiter in *( )*(\t) format.
- `nchar` – characters belonging to the a-z, A-Z, 0-9, -, _ range.
- `vchar` – characters belonging to the nchar, *, ? group of symbols.
- `char` – characters belonging to the 32 – 255 range of values.

Sample iteration construction:

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

During execution of the construction preprocessor parses it into the following conventional constructions:

```
<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 constructions are executed in sequence.

Thus, iteration constructions allow detection of a specific macro value and a group of values.

E.g., if the `%FILTERNAME%` macro has values `KAVFilter1`, `KAVFilter2`, `KAVFilter3`, `SimpleFilter`, then:

the following construction:

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

will be converted into the following text:

```
KAVFilter1
```

construction:

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

will be converted into text:

```
KAVFilter1, KAVFilter2, KAVFilter3
```

construction:

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

will be converted into text:

```
KAVFilter1, KAVFilter3, SimpleFilter
```

construction:

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

will be converted into text:

```
SimpleFilter
```

## B.13.1.2. Visibility borders for iteration constructions

Any iteration construction can include embedded macros defined only within the visibility borders of that construction. Iteration constructions can be used to output specific macro values and also for indication of the visibility borders of the embedded macros.

Visibility borders of an embedded macro are defined by the opening and closing tags of the conditional construction:

```
<FOR _macro_name_parent_ ==
"_value_1">%_macro_name_child_%</FOR>
```

The scope of the *%_macro_*name*_parent_%* macro will apply then to all nested levels (i.e. those within the specified tags), if the macro value is not redefined.

## B.13.1.3. Variables

Variables are used to allow greater flexibility in definitions for template creation.

To define a variable in specified visibility area, the following construction is supported:

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

Further the variable can be used as a regular macro without limitations.

Syntax of variable definition:

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

where:

- `<DEF` – beginning of the variable definition construction. The < character not belonging to the definition must be isolated (see section B.13.1.4 on page 174).

- `VNAME` – name of the variable in **1*(nchar)*(nchar)** format; maximum length is 64 bytes.
- `VOP` – operation of format assignment **=**, length is 1 byte.
- `VVALUE` – variable value in **1*(vchar)*(vchar)** format; maximum length is 4096 bytes. Variable value must be specified in quotes. If the value is compared with a value containing a quotation mark, an isolating (escape) character has to be used (see section B.13.1.4 on page 174). Sample construction for variable definition:

  ```
  <DEF _value_name_ = "\"_value_1\""/>
  ```

- `>` – end of construction for variable definition. The > character not belonging to the end of variable definition must be isolated (see section B.13.1.4 on page 174). DEF construction has no body like the FOR construction, so the closing bracket of its tag must notify the parser about absence of the closing tag.
- `…` – delimiter in ***( )*(\t)** format.
- `nchar` – characters belonging to the a-z, A-Z, 0-9, -, _ range.
- `vchar` – characters belonging to the nchar, *, ? group of symbols.

In case of variable redefining within the borders of its visibility, the new value will be substituted after every new definition. Thus, the following construction:

```
<DEF __NAME__= "NAME_1"/>Here we shall see the first
value: %__NAME__%.
<DEF __NAME__= "NAME_2"/>Here we shall see the second
value: %__NAME__%.
```

will be converted into the following text:

```
Here we shall see the first value: NAME_1.
Here we shall see the second value: NAME_2.
```

A variable can use a macro as its value.

```
<DEF _var_name_ = “%_macro_name_%”/>
```

In that case preprocessor will first replace the variable with macro and then with its value.

## B.13.1.4. Language syntax

### Service characters

| | |
|---|---|
| **%** | Macro indicator. A macro is recorded between two "%" characters.<br>Example: %VIRUSNAME% |
| **<** | Opening tag bracket.<br>Example: `<FOR FILTERNAME == "KAVFilter1">` |
| **>** | Closing tag bracket.<br>Example: `<FOR FILTERNAME == "KAVFilter1">` |
| **</** | Opening bracket of the closing tag.<br>Example: `</FOR>` |
| **/>** | Closing bracket of a bodyless construction tag.<br>Example: `<DEF __NAME __= "ИМЯ_1"/>` |
| **\** | escape character. Cancels the token following it. Example: `\%VIRUSNAME\%` |
| **==** | Comparison: mask or value match.<br>Example: `<FOR FILTERNAME == "KAVFilter1">`<br>Example: `<FOR FILTERNAME == "KAVFilter*">` |
| **!=** | Comparison: non-match of mask or value.<br>Example: `<FOR FILTERNAME != "KAVFilter1">`<br>Example: `<FOR FILTERNAME != "KAVFilter*">` |
| ***** | All possible values of unlimited size. The character is only used within tags during comparison with templates.<br>Example: `<FOR FILTERNAME == "KAV*">` |
| **?** | All possible single character values. The character is only used within tags during comparison with templates.<br>Example: `<FOR FILTERNAME == "KAVFilter?">` |
| **#** | Comment mark, parser ignores all characters beginning with `#` until the end of line. |

**Service words**

| | |
|---|---|
| FOR | Definition of an iteration construction.<br>Example: `<FOR FILTERNAME = "KAVFilter1">` |
| DEF | Variable definition (construction without closing tag). Example: `<DEF __NAME__= "NAME_1"/>` |

## B.13.1.5. Predefined macros

| | |
|---|---|
| %CRLF% | Line feed macro |
| %TAB% | Tabulation macro |

All processing occurs within a global section not defined in any construction or within a conventional construction
`<FOR KAV_LANGUAGE == "5.0"> ... </FOR>`

## B.13.1.6. Escape sequences

Notifications language supports the following sequences:

- Use the ‘\\’ sequence to output the ‘\’ character in template text.
- Line ending in the ‘\’ escape character continues to the next line. The escape character is displayed on-screen as a line feed character. During processing such line will be concatenated with the next line before the parser performs any steps for template processing. Such escape character works irrespectively of its position inside or outside a tag.

  To place the ‘\’ character in the line end so that it is not interpreted as a line feed, use the ‘\\’ sequence.
- To output in template text the ‘%’ character, use the ‘\%’ sequence.
- To output in template text the ‘/’ character, use the ‘\/’ sequence.
- To output in template text the ‘<’ character, use the ‘\<’ sequence.
- To output in template text the ‘>’ character, use the ‘\>’ sequence.
- To output in template text the ‘#’ character, use the ‘\#’ sequence.

## B.13.2. Macros

| Name | Disclamer | DSN | Notify | Place-holder | Value |
|---|---|---|---|---|---|
| APPLICATION_VERSION | X |  | X | X | String of the «5.6.X/RELEASE» type |
| APPLICATION_NAME | X |  | X | X | «Kaspersky Mail Gateway» string |
| DATE |  | X | X | X | Current date in RFC 822 format |
| DESTINATIONFILE |  |  | X |  | Processed message |
| DSN_STATUS |  | X |  |  | Depending upon DSN type:<br>DSN_FAILED \| DSN_DELAYED \| DSN_DELIVERED\| DSN_RELAYED |
| ENVELOPE_ID | X | X | X |  | EnvelopeID of the current message |
| ENVID |  | X |  |  | Original message EnvelopeID |
| INITIAL_MSG_TYPE |  | X |  |  | Depending upon the value of the **DSNEntireMessage** option in the **[mailfw.options]** section that determines whether just headers or entire message should be included into DSN |
| HOSTNAME | X | X | X | X | Host name from the configuration file |

| **Name** | **Disclamer** | **DSN** | **Notify** | **Place-holder** | **Value** |
|---|---|---|---|---|---|
| MESSAGE | | X | | | Headers or entire message depending upon the value of the **DSNEntireMessage** option in the **[mailfw.options]** |
| MESSAGE_BODY | X | | | | Message body |
| MESSAGE_HEADERS | X | | | | Headers in FIELD array (i.e. all headers of the original message) |
| NOTIFY_TYPE | | | X | | Notification recipient SENDER \| RECIPIENT \| ADMIN |
| ORCPT | | X | | | From information about the recipient |
| ORIG_MESSAGE_ID | | | X | | **MessageID** field from the outgoing message |
| ORIG_RECIPIENTS | | | X | | Recipients from the outgoing message |
| ORIG_SENDER | | | X | | Sender from the outgoing message |
| POSTMASTER | | X | | | Value of the **Postmaster** option in the **[mailgw.network]** section of the configuration file |
| QUEUEDAYS | | X | | | Duration of storage in queue |

| Name | Disclamer | DSN | Notify | Place-holder | Value |
|---|---|---|---|---|---|
| QUARANTINE | | | X | | Indicates whether a message has been quarantined |
| QUARANTINE_ID | | | X | | If a message has been quarantined, ID of the message. |
| RCPT_ADDR | | X | X | | Recipient address |
| REMOTEHOST | | X | | | Hostname of the server where message delivery has been attempted last time |
| RECIPIENT | | | X | | Notification recipient |
| SENDER | | X | | | Sender of the message for which a DSN notification has been formed. |
| SENDER | | | X | | From configuration file (notification sender) |
| STATUS | | X | | | Code returned by the SMTP server to which the last message delivery has been attempted |
| SOURCEFILE | | | X | | Original message |
| SPAMTEST_STATUS | | | X | | Result of checks by the anti-spam module |

| **Name** | **Disclamer** | **DSN** | **Notify** | **Place-holder** | **Value** |
|---|---|---|---|---|---|
| TRACE_INFO | | X | X | | Value of the **PrependReceived** option from the **[mailgw.options]** section of the configuration file |

Some macros contain results for PART portions, one such part is present in **placeholder** and several others in **notify**. The **notify** value can be empty, **placeholder** – INLINE or PART, depending on whether the part being replaced is a separate portion or it is contained in another part (e.g., in UUE format).

It contains:

| | |
|---|---|
| PARTNAME | File name |
| PARTMIMETYPE | MIME type from header |
| PARTCHARSET | **charset** field from header |
| PARTSTATUS | Status after AV scan |
| PARTACTION | Action applied to the part |
| SCAN_RESULT | Status after AV scan |

Depending upon the status assigned after anti-virus scanning, SCAN_RESULT contains one of the following values:

| | |
|---|---|
| VIRUS_NAME | Virus name if the status is **infected** |
| SUSPICIOUS | Name if the status is **suspicious** |
| FILTERED_MIME | String matching the mask for filtration by MIME type |
| FILTERED_NAME | String matching the mask for filtration by attachment name |
| DISINFECTED_OBJECT | Virus name if the status is **cure** |

# B.14. *Mailgw-tlv* utility return codes

The *mailgw-tlv* utility may return the following codes on exit:

| | |
|---|---|
| **0** | Template has correct syntax. |
| **1** | Template name for examination has not been specified. |
| **2** | Template file cannot be opened. |
| **3** | Template has incorrect syntax. |
| **4** | System error in template parser operation. |

# B.15. *Mailgw-mailq* utility command line options

| | |
|---|---|
| **Help options** | |
| **-h**<br>**--help** | Output to the console a summary of command line options for the utility, and exit. |
| **-v**<br>**--version** | Output to the console the utility's version number, and exit. |
| **-s**<br>**--show-all** | Output to the console information about all messages in the application's working queue. |
| **-i QueueID**<br>**--show-id= QueueID** | Output to the console, information about the message identified by the number **QueueID**. |
| **Options for work with messages in working queue** | |
| **-q**<br>**--queue-path=** | Specify a custom directory for the application's working queue in the parameter **path_to_queue**. |

| | |
|---|---|
| **path_to_queue** | |
| **-r**<br>**--remove-all** | Remove all messages from the application’s working queue. |
| **-d QueueID**<br>**--remove-id=QueueID** | Remove the message identified by the number **QueueID** from the application’s working queue. |
| **-a**<br>**--send-all** | Send all messages in the application’s working queue to recipients. |
| **-o QueueID**<br>**--send-id=QueueID** | Send the message identified by the number **QueueID** from the application’s working queue to recipients. |

## B.16. *Mailgw-maila* utility command line options

| | |
|---|---|
| **Help options** | |
| **-h**<br>**--help** | Output to the console a summary of command line options, and exit. |
| **-v**<br>**--version** | Output to the console the utility’s version number, and exit. |
| **-s**<br>**--show-all** | Output to the console information about all messages in storage. |
| **-i QueueID**<br>**--show-id= QueueID** | Output to the console information about the message identified by number **QueueID**. |
| **Options for work with messages in storage** | |
| **-q** | Specify a custom path for the application’s working |

| | |
|---|---|
| **--queue-path=**<br>**path_to_queue** | queue in the parameter **path_to_queue**. This option may be necessary when you have to send an archived message to a specific queue of e-mail messages. |
| **-p**<br>**--archive-path=**<br>**path_to_archive** | Specify a custom directory for the archive of e-mail messages in the parameter **path_to_archive**. |
| **-r**<br>**--remove-all** | Remove all the archived messages. |
| **-d QueueID**<br>**--remove-id=QueueID** | Remove the message identified by the number **QueueID** from the message archive. |
| **-a**<br>**--send-all** | Send all messages from the archive to their original recipients. |
| **-o QueueID**<br>**--send-id=QueueID** | Send the message identified by the number **QueueID** from the archive to its original recipients. |
| **-n QueueID**<br>**--send-id-without-check=QueueID** | Send the message identified by the number **QueueID** from the archive to its original recipients without any anti-virus scanning or spam filtering operations. |

## B.17. *Mailgw-maila* and *mailgw-mailq* return codes

The *mailgw-maila* and *mailgw-mailq* utilities may return the following codes on exit:

| | |
|---|---|
| **0** | The utility has finished its operation successfully. |
| **1** | Error in the command line parameters. |
| **2** | Directory cannot be read. |

| | |
|---|---|
| **3** | System error. |
| **4** | Requested action has not been performed. |

# B.18. Special headers added by the anti-spam module

The anti-spam module may add the following headers to messages while processing them:

- **X-SpamTest-Version** – header containing the information about the version of Kaspersky Mail Gateway.
- **X-SpamTest-Status** and **X-SpamTest-Status-Extended** are headers containing message evaluation results reflected in its status. **X-SpamTest-Status** header was used in earlier versions of Kaspersky Anti-Spam (2.0). Now it is used to maintain compatibility with Kaspersky Anti-Spam. Possible header values are listed in the table below:

| Header | Value | Description |
|---|---|---|
| **X-SpamTest-Status** | **SPAM** | Message identified as spam. |
| | **Probable Spam** | Message identified as probable spam. |
| | **Not detected** | Message not identified as spam or probable spam. |
| **X-SpamTest-Status-Extended** | **blacklisted** | Message sender is in the black list of senders. |
| | **Spam** | Message identified as spam. |
| | **probable_spam** | Message identified as probable spam. |
| | **formal** | Message identified as formal automatic reply from e-mail server. |

| | | |
|---|---|---|
| | **not_detected** | Message not identified as spam or probable spam. |

- **X-SpamTest-Obscene** – header added to messages containing obscene words or phrases.
- **X-SpamTest-Formal** – header added to messages identified as **Formal**.
- **X-SpamTest-Rate** – header containing the rating assigned to a message during its filtration. Kaspersky Mail Gateway uses the value to assign a certain status to messages.
- **X-SpamTest-Group-ID** – header containing the identifier of the group associated with the rules used for message processing.
- **X-SpamTest-Categories** – header containing the category assigned to a message after its filtration.
- **X-SpamTest-Info** – header containing informational messages.
- **X-SpamTest-Envelope-From** – header containing sender address from SMTP envelope. It is used to monitor the application of local black and white lists.
- **X-SpamTest-Method** – header containing the methods, which provided the results for assignment of the message status. Possible values of the header are listed in the table below.

| Value | Method |
|---|---|
| **white ip list** | Check against white list of sender IP addresses. |
| **white e-mail list** | Check against white list of sender e-mail addresses. |
| **black ip list** | Check against black list of sender IP addresses. |
| **black e-mail list** | Check against black list of sender e-mail addresses. |
| **GSG** | Graphical signature analysis. |

| | |
|---|---|
| **headers**<br>and<br>**headers plus** | Header analysis. |
| **DNSBL** | Check involving DNSBL services. |
| **UDS** | Check involving UDS services. |
| **UDS BL** | Check involving UDS services. The method is a combined check using heuristic rules and black lists. |
| **SURBL** | Check using SURBL service. |
| **Content** | Check of e-mail message content. |
| **probable** | Probable spam method. |
| **detection disabled** | Anti-spam e-mail filtration is disabled for the recipient in the group policy. |
| **Multiple** | Results of several methods have been used to assign message status. |
| **None** | None of the methods allows message classification. Such messages receive the **Not detected** status. |

## B.19. Format of messages about anti-virus scanning and spam filtration

The application allows statistical data of the anti-virus and anti-spam modules to be viewed separately.

To create a file containing statistical data of the AV module and the anti-spam module, specify the file name as for value for the option **MessageStatis-**

**tics=<file_name>** in the **[mailgw.options]** section of the configuration file. This statistics file will store information about each processed object.

Each line in the statistics file will contain data about one processed object, using the following format ("\t" indicates the tab character):

```
Time \t Size \t Sender \t Recipients \t Status \t VirusList
\t IP \t Message-id \t GroupName \t QueueID
```

Example:

```
1215489931      1022    <user1@example.com>
<user2@example.com>     as/spam_not_detected,av/infected
EICAR-Test-File 10.12.10.145
<200611231502.15109.user1@example.com>       group1
m85STOEp13057
```

Table 5 contains descriptions of each parameter. If the parameter is optional, the corresponding field in the report line may remain blank.

Table 5. Statistics-related parameters

| Symbolic name | Value | Required parameter |
|---|---|---|
| Time | Record creation time | Yes |
| Size | Record size | Yes |
| Sender | Sender's e-mail address | Yes |
| Recipients | E-mail addresses of recipients. Several addresses can be listed. | Yes |
| Status | List of statuses assigned after the anti-virus scan and anti-spam processing. | Yes |
| VirusList | List of viruses. | No |
| IP | IP-address of the host from which the message was received. | No |
| Message-id | Message identification number. | No |

| | | |
|---|---|---|
| GroupName | Name of the group associated with the rules used to process the message. | No |
| QueueID | Identification number of the message in queue. | No |

The information in the statistics file is logged after the anti-virus scan and spam filtering of e-mail message has been performed.

If, for some reason, the report on the object's processing cannot be output (for example, the statistics server is not available), information about the object will not be logged.

## B.20. Notifications about actions applied to the message

Messages added to the log file may be different depending on the action performed.

When a message is delivered, the following line is added to the report file:

```
envelope-id: RECEIVED, from=<...>, nrcpt=...,
size=..., client=[...], helo=<...>,
message_id=<message id>, flags=...
```

where:

- `envelope-id` – message identifier in the application working queue;
- `from` – value received from the MAIL FROM command;
- `nrcpt` – the number of e-mail message recipients (transmitted with the RCPT TO command(s));
- `size` – message size;
- `client` – IP address of the client's host;
- `helo` – client's domain name, received from HELLO/EHLO command;
- `message_id` – message ID;
- `flags` – flag(s) that have the following meanings:
    - `E` – used ESMTP;

- `D` – client requested DSN-confirmations.

When message processing by the anti-virus engine completes, the following line is added to the log file:

```
envelope-id: AV-SCANNED, group=<...>, nrcpt=...,
srcid=...,
status="...", names="..."
```

where:

- `envelope-id` – message identifier in the application working queue;
- `group` – the name of the group of recipients (or **policy** group) to which the message belongs;
- `nrcpt` – the number of recipients of this e-mail message (out of the recipients that belong to this group);
- `srcid` – the original message's ID;
- `status` – status assigned to the message based on the anti-virus scan results;
- `names` – names of viruses, if any, separated by ", ".

When message processing by the anti-spam module is over, the following line is added to the log file:

```
envelope-id: AS-SCANNED, group=<...>, nrcpt=...,
as-status="...", as-category="..."
```

where:

- `envelope-id` – message identifier in the application working queue;
- `group` – the name of the group of recipients (or **policy** group) to which the message belongs;
- `nrcpt` – the number of recipients of the e-mail message (out of the recipients that belong to this group);
- `as-status` – status assigned as a result of its processing by the anti-spam module;
- `as-category` – category assigned to the message based on the content analysis.

When generating system notifications, the following line will be added to the log file:

```
envelope-id: CREATED,
notify=<admin|recipient|sender>, nrcpt=..., size=...,
srcid=...
```

where:

- `envelope-id` – the message's identifier in the application's working queue;
- `notify` – account where the notification will be delivered (possible values are `admin, recipient, sender`);
- `nrcpt` – the number of recipients of the e-mail message;
- `size` – message size;
- `srcid` – original message ID.

When a copy of an e-mail message is created (for delivery of that message to different groups of recipients) the following line will be added to the log file:

```
envelope-id: SPLITTED, domain=<...>, nrcpt=...,
srcid=...
```

where:

- `envelope-id` – message identifier in the application working queue;
- `domain` – name of the domain for which a copy of the original message was created;
- `nrcpt` – the number of recipients of the e-mail message (out of the recipients that belong to this group);
- `srcid` – the original message's ID.

When an e-mail message is delivered, the following line will be added to the log file:

```
envelope-id: DELIVERED, rcpt=<...>, server=...,
size=..., status=sent|failed
```

where:

- `envelope-id` – message identifier in the application's working queue;
- `rcpt` – address of the message's recipient(s);
- `server` – IP address and name of the server where the message is delivered;

- `size` – message size;
- `status` – delivery status, possible values are:
  - `sent` – message was successfully delivered;
  - `failed` – message was not delivered.

When an e-mail message is blocked, the following line will be added to the log file:

```
envelope-id: BLOCKED, rcpt=..., size=...
```

where:

- `envelope-id` – message identifier in the application's working queue;
- `rcpt` – address of the message recipient;
- `size` – message size.

# APPENDIX C. SENDING SPAM TO THE GROUP OF SPAM ANALYSTS

Kaspersky Lab is grateful to all its users providing new samples of spam to the group of spam analysis. Received samples help us react in a timely manner to new methods of spam delivery preventing them during initial distribution stages.

You can also send to us samples of mail erroneously recognized as spam. The messages will be carefully examined by the experts at linguistic laboratory who will be able to increase the quality of spam recognition and make the number of false alerts lower.

Sending spam samples in accordance with the instruction below maximally automates mail processing and shortens the response time of Kaspersky Mail Gateway to new methods used in spam mail.

| | |
|---|---|
| Address for spam samples: | spam@kaspersky.com |
| Address for messages erroneously recognized as spam: | notspam@kaspersky.com |

**Attention!**

Spam samples should be sent as message attachments**.**

Different e-mail programs use different methods to ensure minimum loss of message headers in transit. We describe the procedure for users of most popular e-mail clients.

1. To forward spam using the e-mail client of Microsoft Office Outlook, perform the following steps:

   - If you wish to forward a single message, create a new letter using the **New** button or the **New Mail Message** command and drag the spam message to the new letter with the mouse.

   - If you wish to forward several messages, highlight them and press the **Forward** button. E-mail client will automatically forward the selected messages as attachments to the new letter.

2. To forward spa using The Bat! e-mail client, perform the following steps:
   - If you wish to forward mail manually, highlight one or several messages and use the **Alternative Forward** command accessible from the **Specials** toolbar menu.
   - If you wish to configure automatic spam forwarding, configure the sorting rules in mail manager as follows:
     - Disable the **Do not send attachments** checkbox.
     - Disable the **Use MIME** checkbox.
3. To forward spam using Microsoft Outlook Express e-mail client, select one or several messages and perform **Message** → **Forward as Attachment** command.

# APPENDIX D. 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.

# D.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.
- Disinfect infected detected objects.
- 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, and Linux 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 withal 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 three 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.

# D.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 http://www.kaspersky.com/supportinter.html<br>Helpdesk: www.kaspersky.com/helpdesk.html |
| General information | WWW: http://www.kaspersky.com<br>http://www.viruslist.com<br>E-mail: info@kaspersky.com |

# APPENDIX E. LICENSE AGREEMENT

End User License Agreement

NOTICE TO ALL USERS: CAREFULLY READ THE FOLLOWING LEGAL AGREEMENT ("AGREEMENT") FOR THE LICENSE OF SPECIFIED SOFTWARE ("SOFTWARE") PRODUCED BY KASPERSKY LAB ("KASPERSKY LAB").

IF YOU HAVE PURCHASED THIS SOFTWARE VIA THE INTERNET BY CLICKING THE ACCEPT BUTTON, YOU (EITHER AN INDIVIDUAL OR A SINGLE LEGAL ENTITY) CONSENT TO BE BOUND BY AND BECOME PARTY TO THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, CLICK THE BUTTON THAT INDICATES THAT YOU DO NOT ACCEPT THE TERMS OF THIS AGREEMENT, AND DO NOT INSTALL THE SOFTWARE.

IF YOU HAVE PURCHASED THIS SOFTWARE ON A PHYSICAL MEDIUM, HAVING BROKEN THE CD'S SLEEVE YOU (EITHER AN INDIVIDUAL OR A SINGLE ENTITY) ARE CONSENTING TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT DO NOT BREAK THE CD's SLEEVE, DOWNLOAD, INSTALL OR USE THIS SOFTWARE.

IN ACCORDANCE WITH THE LEGISLATION, REGARDING KASPERSKY SOFTWARE INTENDED FOR INDIVIDUAL CONSUMERS (KASPERSKY ANTI-VIRUS PERSONAL, KASPERSKY ANTI-VIRUS PERSONAL PRO, KASPERSKY ANTI-HACKER, KASPERSKY SECURITY FOR PDA) PURCHASED ON LINE FROM THE KASPERSKY LAB INTERNET WEB SITE, CUSTOMER SHALL HAVE A PERIOD OF 7 WORKING DAYS AS FROM THE DELIVERY OF PRODUCT TO MAKE RETURN OF IT TO THE MERCHANT FOR EXCHANGE OR REFUND, PROVIDED THE SOFTWARE IS NOT UNSEALED.

REGARDING THE KASPERSKY SOFTWARE INTENDED FOR INDIVIDUAL CONSUMERS (KASPERSKY ANTI-VIRUS PERSONAL, KASPERSKY ANTI-VIRUS PERSONAL PRO, KASPERSKY ANTI-HACKER, KASPERSKY SECURITY FOR PDA) NOT PURCHASED ONLINE VIA INTERNET, THIS SOFTWARE NEITHER WILL BE RETURNED NOR EXCHANGED EXCEPT FOR CONTRARY PROVISIONS FROM THE PARTNER WHO SELLS THE PRODUCT. IN THIS CASE, KASPERSKY LAB WILL NOT BE HELD BY THE PARTNER'S CLAUSES.

THE RIGHT TO RETURN AND REFUND EXTENDS ONLY TO THE ORIGINAL PURCHASER.

All references to "Software" herein shall be deemed to include the software activation key ("Key Identification File") with which you will be provided by Kaspersky Lab as part of the Software.

1. License Grant. Subject to the payment of the applicable license fees, and subject to the terms and conditions of this Agreement, Kaspersky Lab hereby grants you the non-exclusive, non-transferable right to use one copy of the specified version of the Software and the accompanying documentation (the "Documentation") for the term of this Agreement solely for your own internal business purposes. You may install one copy of the Software on one computer, workstation, personal digital assistant, or other electronic device for which the Software was designed (each a "Client Device"). If the Software is licensed as a suite or bundle with more than one specified Software product, this license applies to all such specified Software products, subject to any restrictions or usage terms specified on the applicable price list or product packaging that apply to any such Software products individually.

1.1 Use. The Software is licensed as a single product; it may not be used on more than one Client Device or by more than one user at a time, except as set forth in this Section.

1.1.1 The Software is "in use" on a Client Device when it is loaded into the temporary memory (i.e., random-access memory or RAM) or installed into the permanent memory (e.g., hard disk, CD-ROM, or other storage device) of that Client Device. This license authorizes you to make only as many back-up copies of the Software as are necessary for its lawful use and solely for back-up purposes, provided that all such copies contain all of the Software's proprietary notices. You shall maintain records of the number and location of all copies of the Software and Documentation and will take all reasonable precautions to protect the Software from unauthorized copying or use.

1.1.2 If you sell the Client Device on which the Software is installed, you will ensure that all copies of the Software have been previously deleted.

1.1.3 You shall not decompile, reverse engineer, disassemble or otherwise reduce any part of this Software to a humanly readable form nor permit any third party to do so. The interface information necessary to achieve interoperability of the Software with independently created computer programs will be provided by Kaspersky Lab by request on payment of its reasonable costs and expenses for procuring and supplying such information. In the event that Kaspersky Lab notifies you that it does not intend to make such information available for any reason, including (without limitation) costs, you shall be permitted to take such steps to achieve interoperability, provided that you only reverse engineer or decompile the Software to the extent permitted by law.

1.1.4 You shall not make error corrections to, or otherwise modify, adapt, or translate the Software, nor create derivative works of the Software, nor permit any third party to copy the Software (other than as expressly permitted herein).

1.1.5 You shall not rent, lease or lend the Software to any other person, nor transfer or sub-license your license rights to any other person.

1.1.6 You shall not use this Software in automatic, semi-automatic or manual tools designed to create virus signatures, virus detection routines, any other data or code for detecting malicious code or data.

1.2 Server-Mode Use. You may use the Software on a Client Device or on a server ("Server") within a multi-user or networked environment ("Server-Mode") only if such use is permitted in the applicable price list or product packaging for the Software. A separate license is required for each Client Device or "seat" that may connect to the Server at any time, regardless of whether such licensed Client Devices or seats are concurrently connected to or actually accessing or using the Software. Use of software or hardware that reduces the number of Client Devices or seats directly accessing or utilizing the Software (e.g., "multiplexing" or "pooling" software or hardware) does not reduce the number of licenses required (i.e., the required number of licenses would equal the number of distinct inputs to the multiplexing or pooling software or hardware "front end"). If the number of Client Devices or seats that can connect to the Software exceeds the number of licenses you have obtained, then you must have a reasonable mechanism in place to ensure that your use of the Software does not exceed the use limits specified for the license you have obtained. This license authorizes you to make or download such copies of the Documentation for each Client Device or seat that is licensed as are necessary for its lawful use, provided that each such copy contains all of the Documentation’s proprietary notices.

1.3 Volume Licenses. If the Software is licensed with volume license terms specified in the applicable product invoicing or packaging for the Software, you may make, use or install as many additional copies of the Software on the number of Client Devices as the volume license terms specify. You must have reasonable mechanisms in place to ensure that the number of Client Devices on which the Software has been installed does not exceed the number of licenses you have obtained. This license authorizes you to make or download one copy of the Documentation for each additional copy authorized by the volume license, provided that each such copy contains all of the Document's proprietary notices.

2. Duration. This Agreement is effective for the period specified in the Key File (the unique file which is required to fully enable the Software, please see Help/ about Software or Software about, for Unix/Linux version of the Software see the notification about expiration date of the Key File) unless and until earlier terminated as set forth herein. This Agreement will terminate automatically if you fail to comply with any of the conditions, limitations or other requirements described herein. Upon any termination or expiration of this Agreement, you must immediately destroy all copies of the Software and the Documentation. You may terminate this Agreement at any point by destroying all copies of the Software and the Documentation.

3. Support.

(i) Kaspersky Lab will provide you with the support services ("Support Services") as defined below for a period of one year following:

(a) Payment of its then current support charge, and:

(b) Successful completion of the Support Services Subscription Form as provided to you with this Agreement or as available on the Kaspersky Lab website, which will require you to produce the Key Identification File which will have been provided to you by Kaspersky Lab with this Agreement. It shall be at the absolute discretion of Kaspersky Lab whether or not you have satisfied this condition for the provision of Support Services.

(ii) Support Services will terminate unless renewed annually by payment of the then-current annual support charge and by successful completion of the Support Services Subscription Form again.

(iii) By completion of the Support Services Subscription Form you consent to the terms of the Kaspersky Lab Privacy Policy, which is deposited on www.kaspersky.com/privacy, and you explicitly consent to the transfer of data to other countries outside your own as set out in the Privacy Policy.

(iv) "Support Services" means:

(a) Daily updates of the anti-virus database;

(b) Free software updates, including version upgrades;

(c) Extended technical support via e-mail and phone hotline provided by Vendor and/or Reseller;

(d) Virus detection and disinfection updates 24 hours per day.

4. Ownership Rights. The Software is protected by copyright laws. Kaspersky Lab and its suppliers own and retain all rights, titles and interests in and to the Software, including all copyrights, patents, trademarks and other intellectual property rights therein. Your possession, installation, or use of the Software does not transfer any title to the intellectual property in the Software to you, and you will not acquire any rights to the Software except as expressly set forth in this Agreement.

5. Confidentiality. You agree that the Software and the Documentation, including the specific design and structure of individual programs and the Key Identification File, constitute confidential proprietary information of Kaspersky Lab. You shall not disclose, provide, or otherwise make available such confidential information in any form to any third party without the prior written consent of Kaspersky Lab. You shall implement reasonable security measures to protect such confidential information, but without limitation to the foregoing shall use best endeavours to maintain the security of the Key Identification File.

6. Limited Warranty.

(i) Kaspersky Lab warrants that for six (6) months from first download or installation the Software purchased on a physical medium will perform substantially in accordance with the functionality described in the Documentation when operated properly and in the manner specified in the Documentation.

(ii) You accept all responsibility for the selection of this Software to meet your requirements. Kaspersky Lab does not warrant that the Software and/or the Documentation will be suitable for such requirements nor that any use will be uninterrupted or error free.

(iii) Kaspersky Lab does not warrant that this Software identifies all known viruses, nor that the Software will not occasionally erroneously report a virus in a title not infected by that virus.

(iv) Your sole remedy and the entire liability of Kaspersky Lab for breach of the warranty at paragraph (i) will be at Kaspersky Lab option, to repair, replace or refund of the Software if reported to Kaspersky Lab or its designee during the warranty period. You shall provide all information as may be reasonably necessary to assist the Supplier in resolving the defective item.

(v) The warranty in (i) shall not apply if you (a) make or cause to be made any modifications to this Software without the consent of Kaspersky Lab, (b) use the Software in a manner for which it was not intended, or (c) use the Software other than as permitted under this Agreement.

(vi) The warranties and conditions stated in this Agreement are in lieu of all other conditions, warranties or other terms concerning the supply or purported supply of, failure to supply or delay in supplying the Software or the Documentation which might but for this paragraph (vi) have effect between the Kaspersky Lab and you or would otherwise be implied into or incorporated into this Agreement or any collateral contract, whether by statute, common law or otherwise, all of which are hereby excluded (including, without limitation, the implied conditions, warranties or other terms as to satisfactory quality, fitness for purpose or as to the use of reasonable skill and care).

7. Limitation of Liability.

(i) Nothing in this Agreement shall exclude or limit Kaspersky Lab's liability for (a) the tort of deceit, (b) death or personal injury caused by its breach of a common law duty of care or any negligent breach of a term of this Agreement, or (c) any other liability which cannot be excluded by law.

(ii) Subject to paragraph (i) above, the Supplier shall bear no liability (whether in contract, tort, restitution or otherwise) for any of the following losses or damage (whether such losses or damage were foreseen, foreseeable, known or otherwise):

(a) Loss of revenue;

(b) Loss of actual or anticipated profits (including for loss of profits on contracts);

(c) Loss of the use of money;

(d) Loss of anticipated savings;

(e) Loss of business;

(f) Loss of opportunity;

(g) Loss of goodwill;

(h) Loss of reputation;

(i) Loss of, damage to or corruption of data, or:

(j) Any indirect or consequential loss or damage howsoever caused (including, for the avoidance of doubt, where such loss or damage is of the type specified in paragraphs (ii), (a) to (ii), (i).

(iii) Subject to paragraph (i), the liability of Kaspersky Lab (whether in contract, tort, restitution or otherwise) arising out of or in connection with the supply of the Software shall in no circumstances exceed a sum equal to the amount equally paid by you for the Software.

8. (i) This Agreement contains the entire understanding between the parties with respect to the subject matter hereof and supersedes all and any prior understandings, undertakings and promises between you and Kaspersky Lab, whether oral or in writing, which have been given or may be implied from anything written or said in negotiations between us or our representatives prior to this Agreement and all prior agreements between the parties relating to the matters aforesaid shall cease to have effect as from the Effective Date. Save as provided in paragraphs (ii) - (iii) below, you shall not have any remedy in respect of an untrue statement made to you upon which you relied in entering into this Agreement ("Misrepresentation") and Kaspersky Lab shall not have any liability to the other than pursuant to the express terms of this Agreement.

(ii) Nothing in this Agreement shall exclude or limit Kaspersky Lab's liability for any Misrepresentation made thereby if aware that it was untrue.

(iii) The liability of Kaspersky Lab for Misrepresentation as a fundamental matter, including a matter fundamental to the maker's ability to perform its obligations under this Agreement, shall be subject to the limitation of liability set out in paragraph 7(iii).

# APPENDIX F. SOFTWARE COMPONENTS FROM THIRD-PARTY VENDORS

This section contains a list of third-party software used in development of Kaspersky Mail Gateway 5.6 and descriptions of terms and conditions regulating the use thereof.

## F.1. Berkeley DB 1.85 library

**Berkeley DB 1.85 library is used subject to the following conditions:**

Copyright (c) 1990, 1993, 1994 The Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by Margo Seltzer.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- 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.
- All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Berkeley and its contributors.
- 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.

THIS SOFTWARE IS PROVIDED BY THE REGENTS 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 REGENTS 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.

# F.2. Libjpeg 6b library

**Libjpeg 6b library is used subject to the following conditions:**

LEGAL ISSUES

============

In plain English:

1. We don't promise that this software works. (But if you find any bugs, please let us know!)

2. You can use this software for whatever you want. You don't have to pay us.

3. You may not pretend that you wrote this software. If you use it in a program, you must acknowledge somewhere in your documentation that you've used the IJG code.

In legalese:

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane.

All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

(1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.

(2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".

(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

ansi2knr.c is included in this distribution by permission of L. Peter Deutsch, sole proprietor of its copyright holder, Aladdin Enterprises of Menlo Park, CA. ansi2knr.c is NOT covered by the above copyright and conditions, but instead by the usual distribution terms of the Free Software Foundation; principally, that you must include source code if you redistribute it. (See the file ansi2knr.c for full details.). However, since ansi2knr.c is not needed as part of any program generated from the IJG code, this does not limit you more than the foregoing paragraphs do.

The Unix configuration script "configure" was produced with GNU Autoconf. It is copyright by the Free Software Foundation but is freely distributable. The same holds for its supporting scripts (config.guess, config.sub, ltconfig, ltmain.sh). Another support script, install-sh, is copyright by M.I.T. but is also freely distributable.

It appears that the arithmetic coding option of the JPEG spec is covered by patents owned by IBM, AT&T, and Mitsubishi. Hence arithmetic coding cannot legally be used without obtaining one or more licenses. For this reason, support for arithmetic coding has been removed from the free JPEG software.

(Since arithmetic coding provides only a marginal gain over the unpatented Huffman mode, it is unlikely that very many implementations will support it.) So far as we are aware, there are no patent restrictions on the remaining code.

The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that

"The Graphics Interchange Format(c) is the Copyright property of

CompuServe Incorporated. GIF(sm) is a Service Mark property of

CompuServe Incorporated."

# F.3. Libungif library

**Libungif library is used subject to the following conditions**

The GIFLIB distribution is Copyright (c) 1997 Eric S. Raymond

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.

# F.4. Libevent library

**Libevent library is used subject to the following conditions:**

Copyright (c) 2000-2004 Niels Provos <provos@citi.umich.edu>

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. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``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 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.

# F.5. Libspf2 library

**Libspf2 library is used subject to the following conditions:**

The code in the libspf-alt distribution is Copyright 2004 by Wayne Schlitt, all rights reserved. Copyright retained for the purpose of protecting free software redistribution.

This program is free software; you can redistribute it and/or modify it under the terms of either:

a) the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1, or (at your option) any later version,

OR

b) The two-clause BSD license.

Some code in the 'replace' subdirectory was obtained form other sources and have different, but compatible, licenses. These routines are used only when the native libraries for the OS do not contain these functions. You should review the licenses and copyright statments in these functions if you are using an OS that needs these functions.

The two-clause BSD license:

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.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "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 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.

# F.6. Libpatricia library

**Libpatricia library is used subject to the following conditions:**

Copyright (c) 1997, 1998, 1999

The Regents of the University of Michigan ("The Regents") and Merit Network, Inc. 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. All advertising materials mentioning features or use of this software must display the following acknowledgement:

This product includes software developed by the University of Michigan, Merit

Network, Inc., and their contributors.

4. Neither the name of the University, Merit Network, nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS 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 REGENTS 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.

# F.7. Pcre library

**Pcre library is used subject to the following conditions:**

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,

# F.8. Zlib library

**Zlib library is used subject to the following conditions:**

Zlib.h -- interface of the 'zlib' general purpose compression library version 1.1.3, July 9th, 1998

Jean-loup Gailly Mark Adler

jloup@gzip.org madler@alumni.caltech.edu

The data format used by the zlib library is described by RFCs (Request for Comments) 1950 to 1952 in the files ftp://ds.internic.net/rfc/rfc1950.txt (zlib format), rfc1951.txt (deflate format) and rfc1952.txt (gzip format).

# F.9. Expat library

**Expat library is used subject to the following conditions:**

# F.10. STLport library

**STLport library is used subject to the following conditions:**

Permission to use or copy this software for any purpose is hereby granted without fee, provided the above notices are retained on all copies. Permission to modify the code and to distribute modified code is granted, provided the above notices are retained, and a notice that the code was modified is included with the above copyright notice.

# F.11. OpenSSL library

**OpenSSL library is used subject to the following conditions:**

LICENSE ISSUES

==============

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License

==============================================================

Copyright (c) 1998-2004 The OpenSSL Project. 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. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED 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 OpenSSL PROJECT OR ITS 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.

====================================================================

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)

All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).

The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

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 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. All advertising materials mentioning features or use of this software must display the following acknowledgement:

"This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)"

The word 'cryptographic' can be left out if the rouines from the library being used are not cryptographic related:-).

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:

"This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "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.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed, i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

# F.12. FreeBSD libc library

**FreeBSD libc library is used subject to the following conditions:**

Copyright (C) 1992-2005 The FreeBSD Project. 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.

THIS SOFTWARE IS PROVIDED BY AUTHOR 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 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.

# F.13. Mcpp preprocessor program

**Mcpp preprocessor program is used under the following conditions:**

Copyright (c) 1998, 2002-2004 Kiyoshi Matsui <kmatsui@t3.rim.or.jp>

All rights reserved.

Some parts of this code are derived from the public domain software DECUS cpp (1984,1985) written by Martin Minow.

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.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "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 BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

# F.14. Libbind library

**Libbind library is used subject to the following conditions:**

# F.15. Snmp++v3.2.22 library

**Snmp++v3.2.22 library is used subject to the following conditions:**

Copyright (c) 2001-2007 Jochen Katz, Frank Fock

This software is based on SNMP++2.6 from Hewlett Packard:

Copyright (c) 1996

Hewlett-Packard Company

ATTENTION: USE OF THIS SOFTWARE IS SUBJECT TO THE FOLLOWING TERMS.

Permission to use, copy, modify, distribute and/or sell this software and/or its documentation is hereby granted without fee. User agrees to display the above copyright notice and this license notice in all copies of the software and any documentation of the software. User agrees to assume all liability for the use of the software; Hewlett-Packard and Jochen Katz make no representations about the suitability of this software for any purpose. It is provided "AS-IS" without warranty of any kind, either express or implied. User hereby grants a royalty-free license to any and all derivatives based upon this software code base.

# F.16. Libdes-l-4.01a library

**Libdes-l-4.01a library is used subject to the following conditions:**

**Copyright notice**

“Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of that the SSL library. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.”

**Conditions**

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 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. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by Eric Young (eay@mincom.oz.au)

**Disclaimer**

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``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.

# F.17. Crypt-1.02 library

**Author of the crypt-1.02 library provided it for public use without any limitations.**

# F.18. AgentX++v1.4.16 library

**AgentX++v1.4.16 library is used subject to the following conditions:**

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-porting 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 Excecutable 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 if 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 LINCENSEE 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

## F.19. Agent++v3.5.28a library

**Agent++v3.5.28a library is used subject to the following conditions:**

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 authorr'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 authorr'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.

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.

## F.20. Universal Charset Detector (Mozilla) library

**Universal Charset Detector (Mozilla)**

Version: MPL 1.1/GPL 2.0/LGPL 2.1

The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Mozilla Communicator client code.

The Initial Developer of the Original Code is Netscape Communications Corporation.

Portions created by the Initial Developer are Copyright (C) 1998 the Initial Developer. All Rights Reserved.

Contributor(s):

Alternatively, the contents of this file may be used under the terms of either the GNU General Public License Version 2 or later (the "GPL"), or the GNU Lesser General Public License Version 2.1 or later (the "LGPL"), in which case the provisions of the GPL or the LGPL are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of either the GPL or the LGPL, and not to allow others to use your version of this file under the terms of the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the GPL or the LGPL. If you do not delete the provisions above, a recipient may use your version of this file under the terms of any one of the MPL, the GPL or the LGPL.