SPAMHALTER

AntiSPAM Mercury/32 daemon

Version 4.3.0

Introduction

No Antispam system is perfect! This is because humans have trouble detecting SPAM, too.

SpamHalter is program that uses a Bayesian engine for detecting SPAM by its experience. This means you must teach this program what SPAM is. Without learning, SpamHalter will not work properly. SpamHalter works better with larger their databases. Larger databases - less mistakes.

SpamHalter is SPAM protection on the server level. It works for all local accounts automatically without any special software on the client side. You can use any post program on the client side.

Instructions for upgrade from Spamwall 3.x.x

1) Stop Mercury/32.
2) Run SpamHalter installer. (You will be asked whether to overwrite Spamwall.ini – say 'NO' to preserve your settings!)
3) Revise all ini file settings. (see this documentation and SpamHalter.ini sample file)
4) Run SpamHalterUpgrade.exe and upgrade all needed databases.
5) Run Mercury/32.

Instructions for upgrade from older Spamwall versions

It is not possible to upgrade directly from Spamwall 1.x.x or 2.x.x to version 4.x.x. Sorry! However you can use SpamwallTools program from older Spamwall versions for upgrade to version 3 and then you can do upgrade into version 4.

Fresh installation instructions:

1) Stop your Mercury/32.
2) Run SpamHalter installer.
3) Create two local mailboxes for SPAM corrections. Users can do corrections by forwarding to these two mailboxes. They are for messages that SpamHalter incorrectly marked as spam or incorrectly didn't mark as spam. Real local mailboxes must be used -- aliases are not allowed!
4) Edit SpamHalter.ini. (see this documentation!)
5) Create word database. (see later in this documentation)
6) Run Mercury/32

Uninstall

You can uninstall SpamHalter using your Control Panel | Add/Remove Programs. Do this the same way that you uninstall any other application. After uninstall ting you can delete all databases, too.

Building of word database

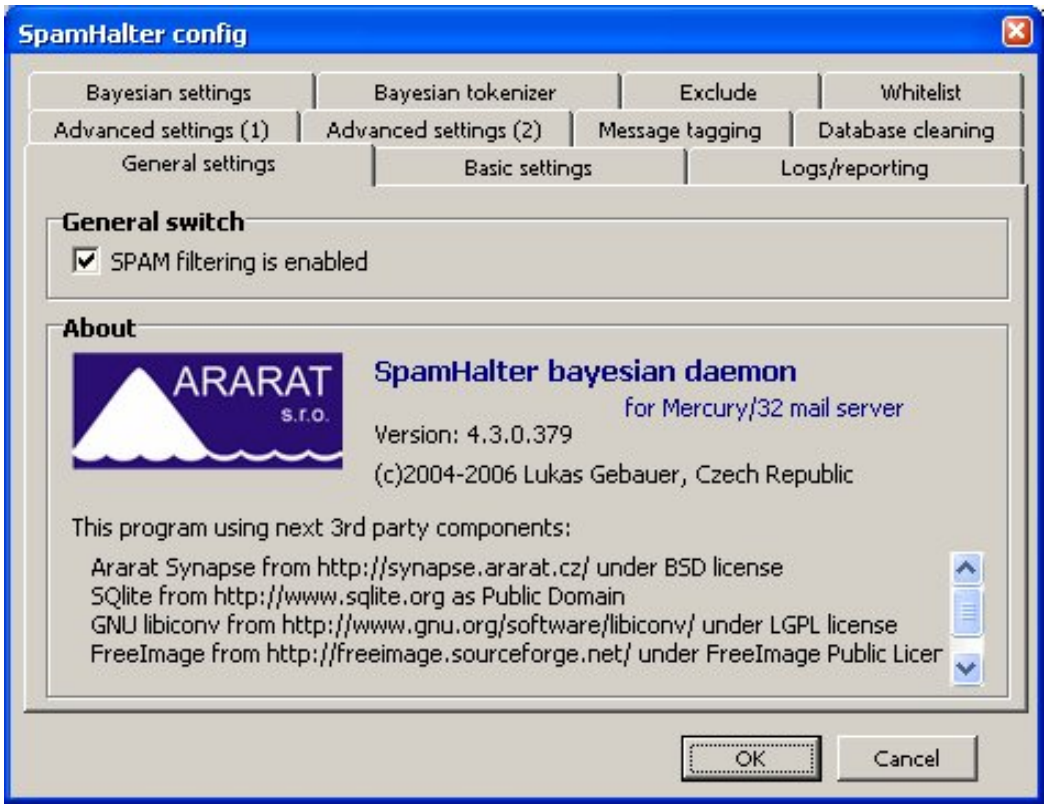
There are a few ways to build your database for Bayesian engine. You can use any combination of these methods:

- Upgrade the database from an older version of SpamHalter. You can use SpamHalterUpgrade.exe for this task. After the upgrade process the new database will be much smaller then the original database. This is because the new database format is much more effective and SpamHalterUpgrade.exe does basic database cleaning, too!
- You can merge your database (it may be empty!) with any other SpamHalter database. You can download this database from the SpamHalter website, or use one from a friend. For this task you can use SpamHalterTools.exe.
- You can create your own database. You must have prepared two directories with a lot of emails saved as text files. One directory for spam messages, the other for nospam messages. Each text file is one RAW message for database. To create these text files you can use the save message feature from Pegasus Mail. Then you must run SpamHalterTools.exe. In the window you can specify your directories with messages, and then add the contents of these directories to the database by hitting the corresponding button.
- You can import Pegasus Mail folders directly too. Use SpamHalterTools.exe.

After initial database is created, then database is updated automatically by each processed message and by user corrections. You must not run SpamHaltertools.exe for this!

Mercury/32 graphical interface screens

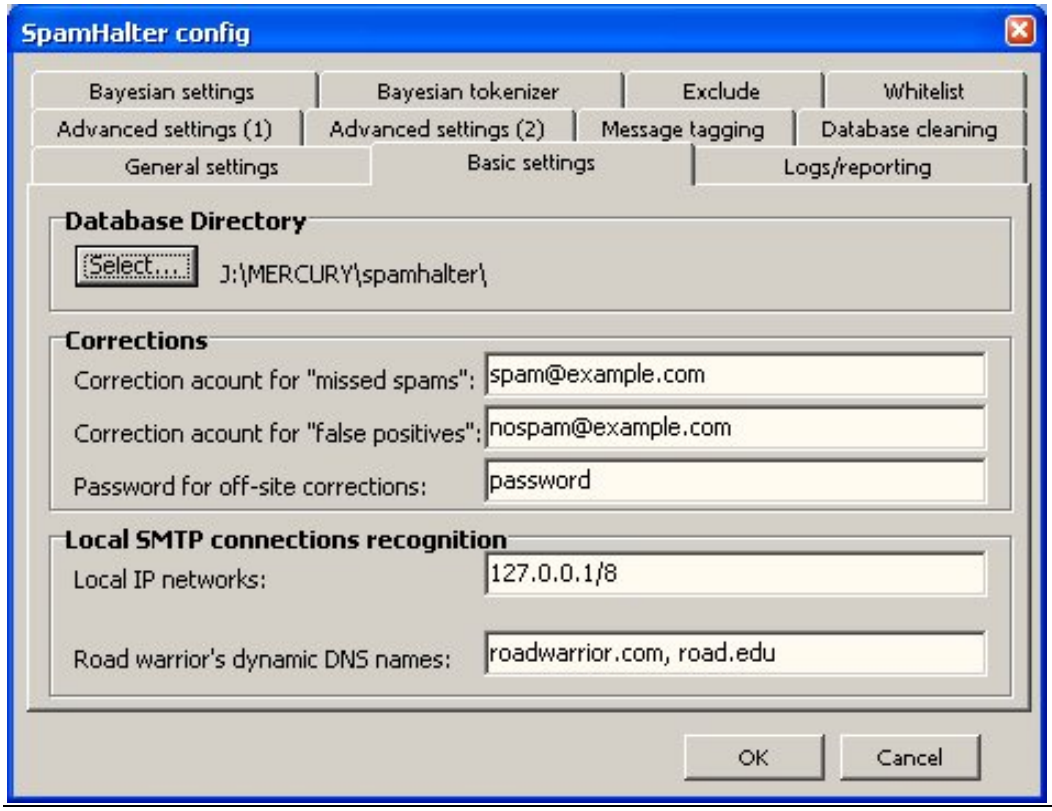
The section location and variable name in the spamhalter.ini file will follow the definitions. This is an example setup with defaults from the program and some basic information to allow the program to operate.



General switch

SpamHalter filter/classification is turned on by checking the box.

[SpamHalter] Enabled



Database directory

This is the directory with the SpamHalter databases. Directory names must end with the '\' character! Because SpamHalter uses databases intensively, you must place this directory on a fast local disk. Be careful with free space on this disk! You will not typically need more then 100 MB of space for SpamHalter. ***[SpamHalter] bayDataDir***

Corrections

Create two local mailboxes for SPAM corrections. Users can do corrections by forwarding to these two mailboxes. They are for messages that SpamHalter incorrectly marked as spam or incorrectly didn't mark as spam. Real local mailboxes must be used -- aliases are not allowed! These can be either simple local email accounts or full email addresses. When used with SpamHalter for Pegasus Mail they must be full addresses.

It is not normally possible send mail to correction addresses from the internet! This is for security reasons. But if you need to do this, then you must enable the '+' aliases feature in Mercury/32. Then you can send mail to the correction addresses followed by '+' and the password specified by this configuration directive.

Example: 'spam+password@domain.com'

[SpamHalter] SpamAddr, NoSpamAddr, Password

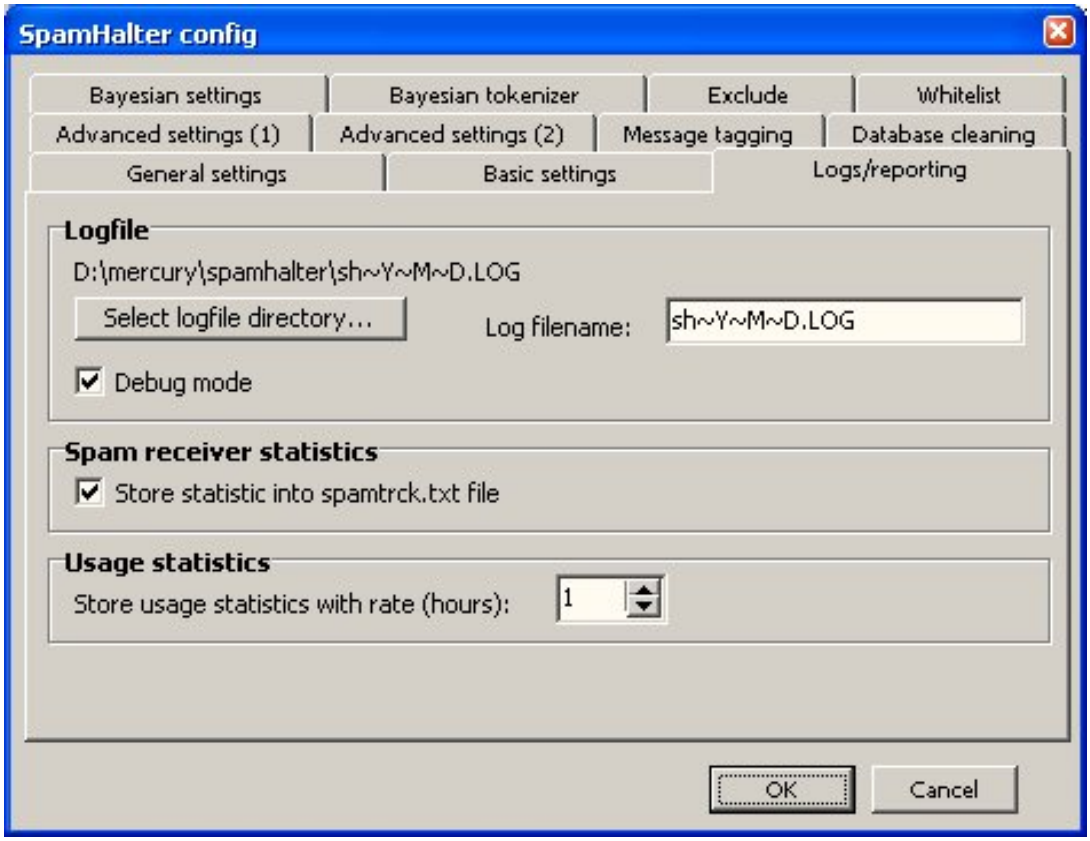
Local SMTP connections recognition

The local IP networks are defined by entering IP addresses with mask length for local IP addresses. Mail that arrives from this IP(s) is not classified by SpamHalter and is processed as local outgoing mail. Example: 127.0.0.1/8,192.168.0.0/16 ***[SpamHalter] LocalIP***

The road warriors dynamic DNS host names is a listing of host names (not IP addresses!) separated by commas of computers that can send messages by SMTP as a local sender. Mail that arrives by SMTP is checked by to see if it's on the local network first. If IP address of SMTP sender does not match any Local IP address, then SpamHalter will try to resolve host names in this listing to IP addresses and then compare sender's IP address with this resolved IPs.

This is good for situations when you have a notebook connected to the internet by various dial-ups. You have different IP addresses on each connection, but DNS computer name can be the same for all connections. Just configure your notebook to register its actual IP address to your dynamic DNS server and specify its DNS name in this directive. When your notebook sends a message to your mail server, SpamHalter will try to resolve its name to IP. If this IP matches the message sender's IP, the message is processed as a message from a local sender.

[SpamHalter] DynamicHost



Logfile

File for SpamHalter logging. You can use the same name macros as you can for log file names in Mercury. In this sample there is a daily log in the form sh060620.log for the day 20 June 2006. Checking the Debug mode will add debug lines into the log file.

[SpamHalter] logfile, Debug

Spam receiver statistics.

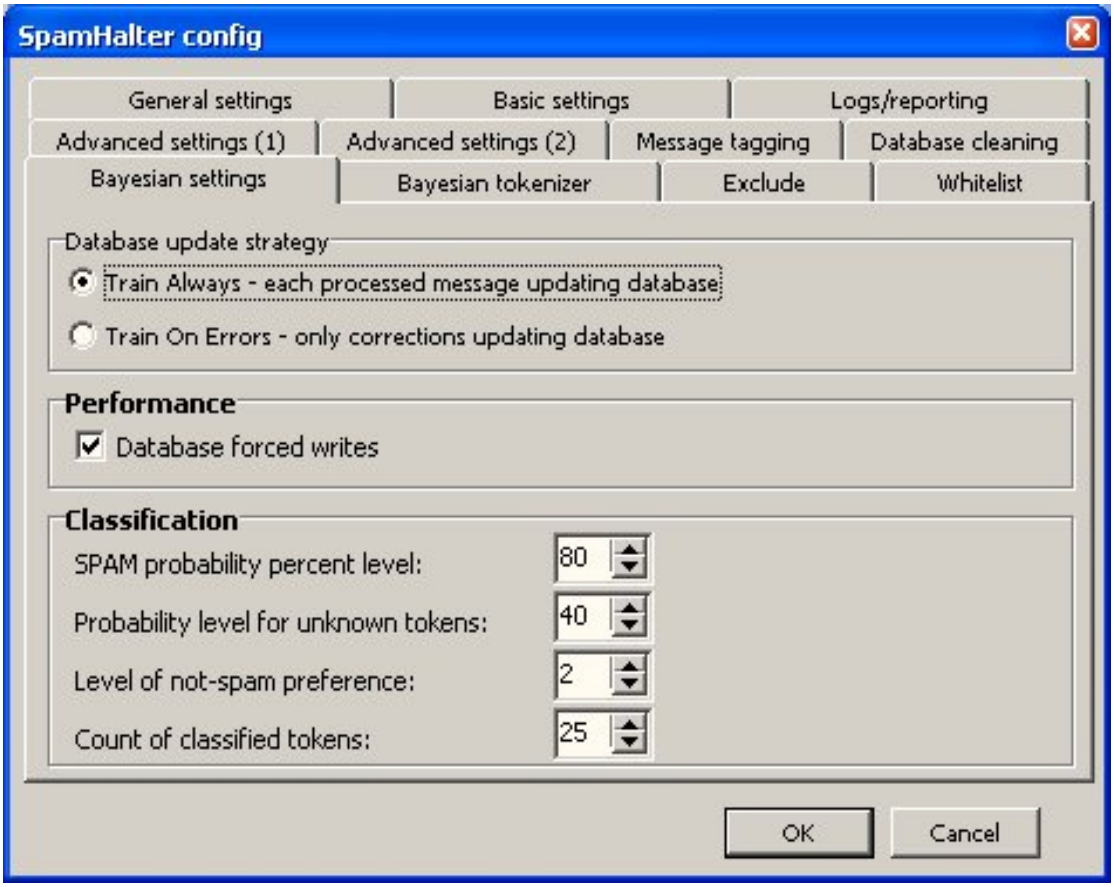
Checking this block will create a file with a listing of the users receiving spam and the number of spams received in the SpamHalter directory.

[SpamHalter] Spamtrack

Usage statistics

Checking this box will create the spamstat files in the SpamHalter directory.

[SpamHalter] StatRate



Database update strategy

There are two possible update strategies that can be used by SpamHalter, Train Always (TA) and Train on Errors (TOE). If you select Train on Errors only then you are effectively putting SpamHalter into a kind of "manual" mode, where it only updates its statistical tables when you explicitly tell it to do so. This results in a smaller database, but slightly less adaptive learning behavior. If you select Train always, then SpamHalter will automatically learn from all the messages it processes, as it processes them: this results in more resilient behavior that will adapt as the types of spam you receive change, but also requires a much larger database and may result in slightly slower classification.

To achieve the greatest spam detection accuracy, large mail systems with a large number of users and high volume of mail should always be using the TA method. The small system with low volume and small number of users should always be using the TOE system. The mix of spam to no spam is also a factor in the selection of this strategy, if there is a very high level of difference between the spam and good mail, or that the number of sources of good man is limited then the TOE strategy will probably be preferable.

[SpamHalter] TrainAlways

Performance

The forced writes is by default is enabled – database engine waits for real disk writes on each database write operation. It is the safest method for protecting the database from corruption by

computer crashes, but it slows down the processing speed. If you have a stable operating system, on stable hardware, protected from power loss by a UPS, you can disable this option. SpamHalter will be much faster in this mode! ***[bayDynamic] bayForcedWrites***

Classification

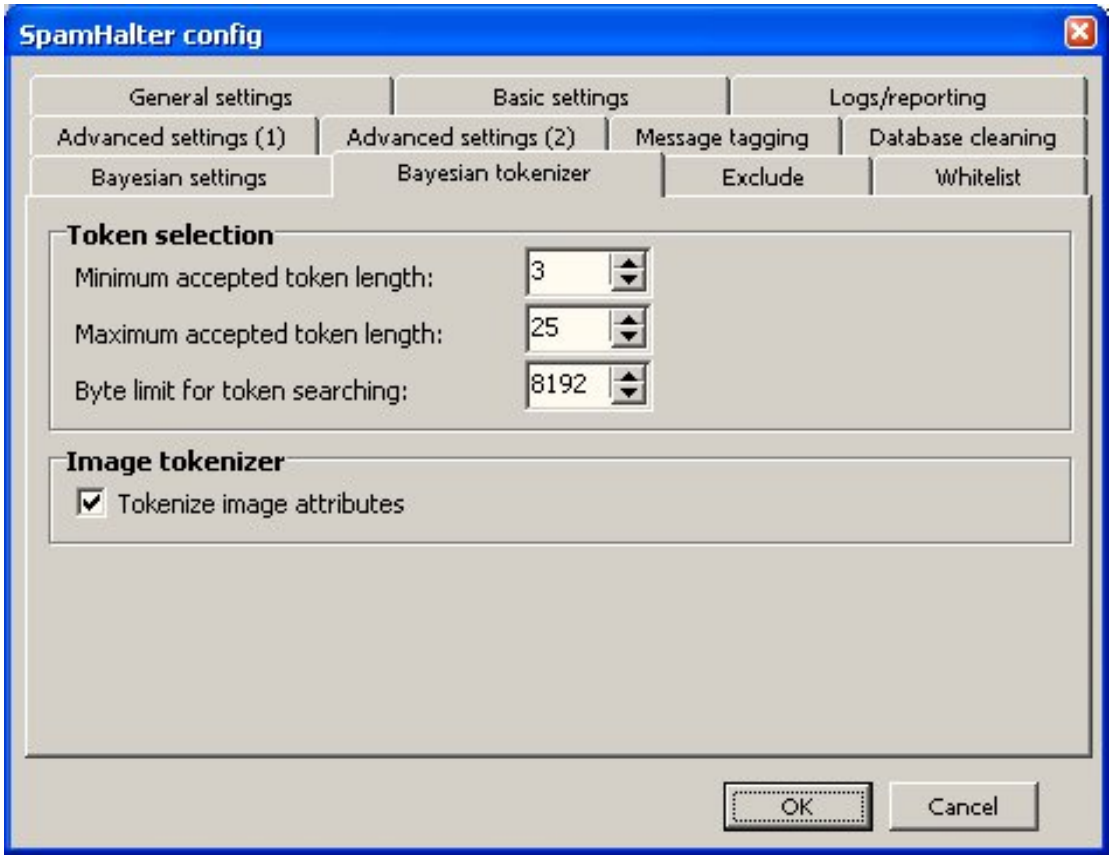
Spam probability percent level: When the computed SPAM probability of the whole message is larger than this value, the message is marked as spam. (Value 80 is 80% probability.) ***[bayDynamic] baySpamProb***

Probability level for unknown tokens: When a word does not exist in the word database, use this probability for this word. (Value 40 is 40% probability.)

[bayDynamic] bayUnknownProb

Level of not-spam preference: While computing, the nospam word count is multiplied by this value. It sets the preference of nospam over spam for reduction of false positives. When starting, you can use a value of 4. When your databases are full, you can use a lower value, for example 2. ***[bayDynamic] bayNoSpamBoost***

Count of classified tokens: Maximum count of most important words in message that are used for Bayesian testing. ***[bayDynamic] bayClassifyMaxTokens***



Token selection

Minimum accepted token length: This defines the minimum word length that will be processed by the Bayesian engine. If a word is shorter, it is ignored by the Bayesian engine. Default is 3 characters. ***[bayStatic] MinTokenLength***

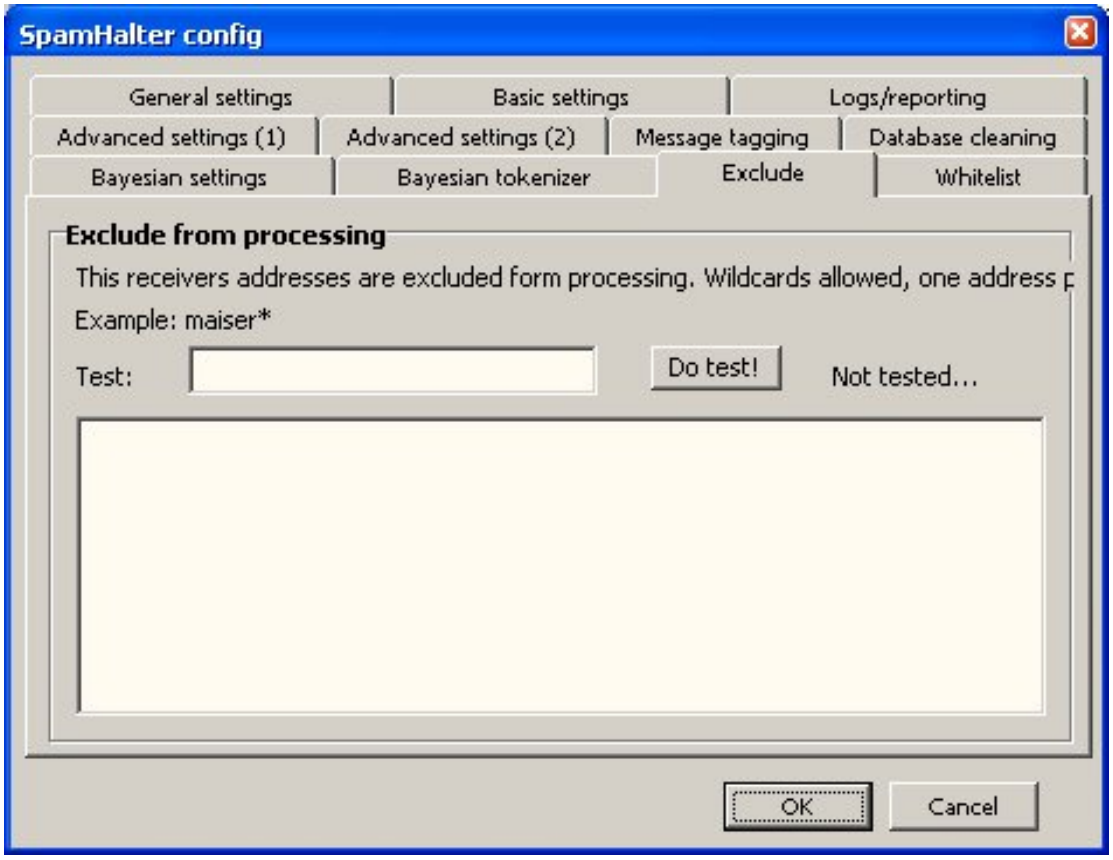
Maximum accepted token length: This defines the maximum word length that will be processed by the Bayesian engine. If a word is longer, it is ignored by the Bayesian engine. Default is 15 characters. ***[bayStatic] MaxTokenLength***

Byte limit for token searching: This defines the maximum message length for Bayesian processing. If message is longer, it is truncated to this length. However before processing, each message is cleaned of binary attachments and unnecessary message headers.

[bayStatic] MaxLength

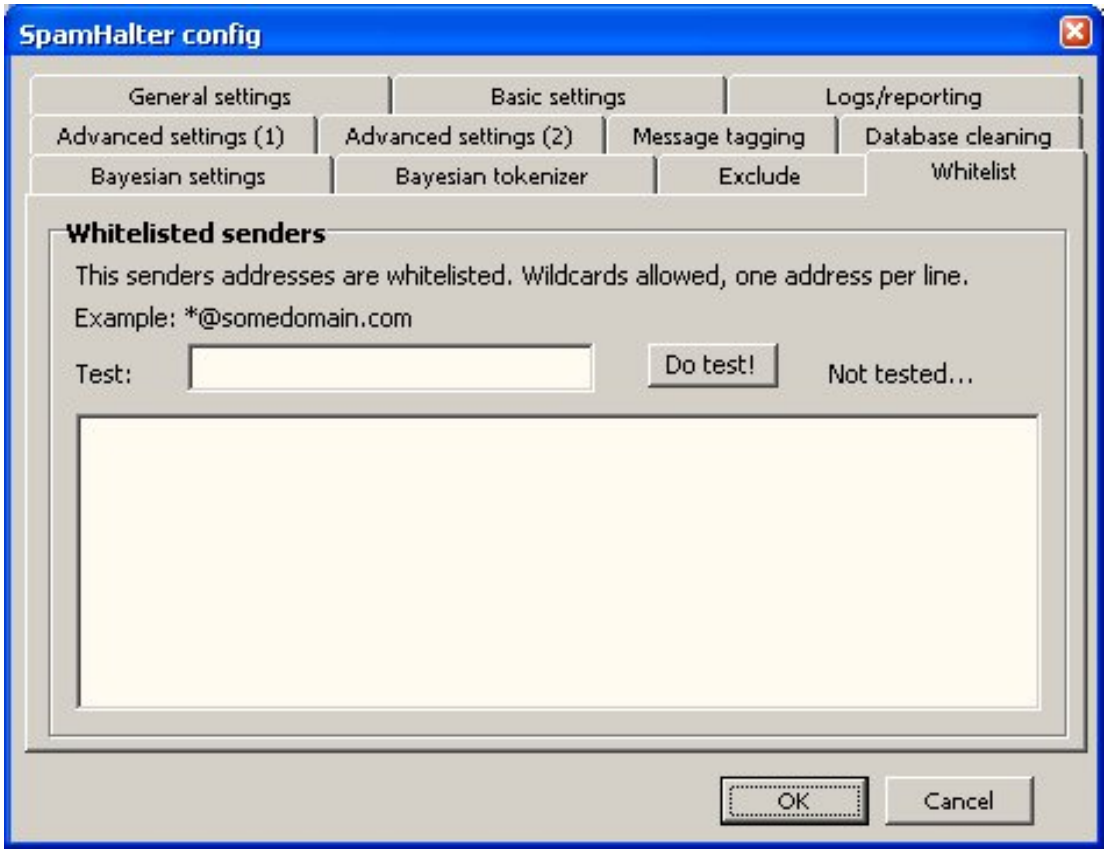
Image tokenizer

Check this box to use the image tokenizer (hammer for image-only spams).
[SpamHalter] ImageParser



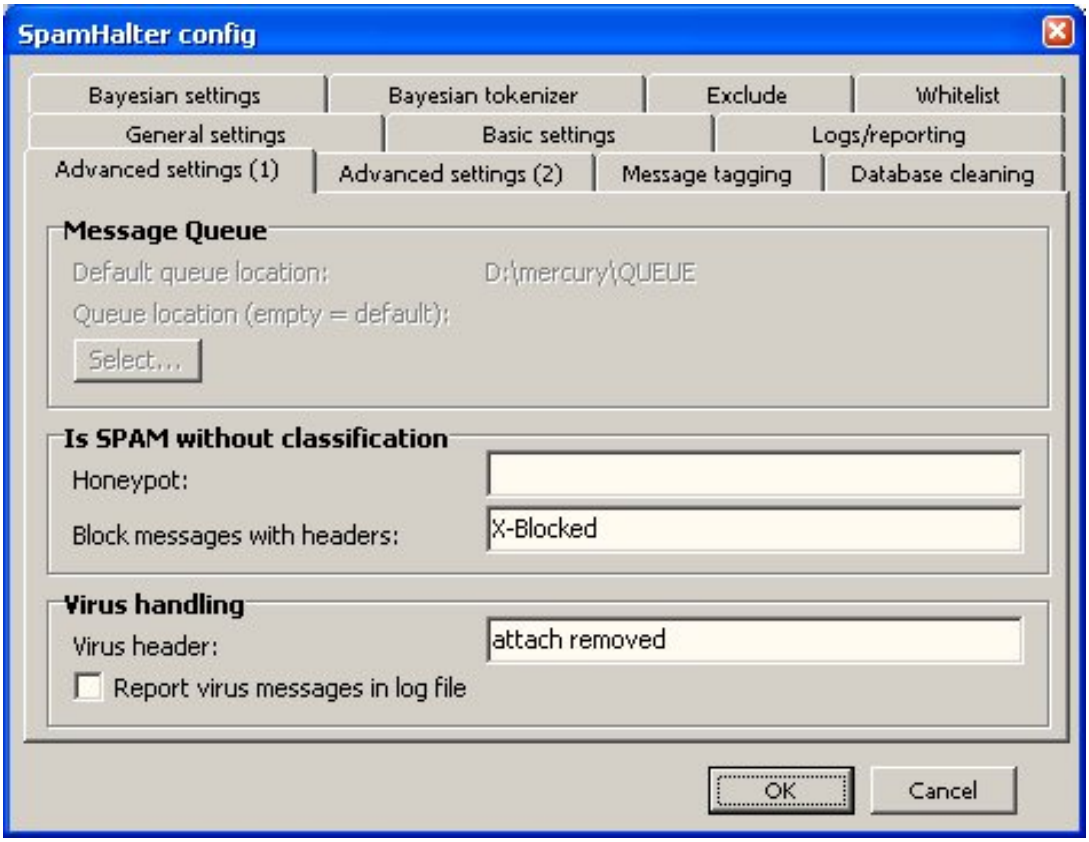
Exclude from processing

A listing of email addresses and domains that will be excluded from processing. Wildcards are allowed, one address per line. The entry *@domain.com will exclude this domain from processing; a*@domain.com will exclude all the users with the name starting with "a" from processing. The address can be tested for validity.



Whitelist senders

A listing of email addresses and domains that will be automatically whitelisted. Wildcards are allowed, one address per line. The entry *@domain.com will exclude this domain from processing; a*@domain.com will exclude all the users with the name starting with "a" from processing. The address can be tested for validity.



Message Queue

This is directory with Mercury/32 queue. If you don't specify this, then automatic detection is used. If automatic detection fails, then you can specify your queue directory here to override automatic detection process.

When you are using Mercury/32 version 4.1 or higher, this directive is ignored because SpamHalter uses a new job accessing code! ***[SpamHalter] queue***

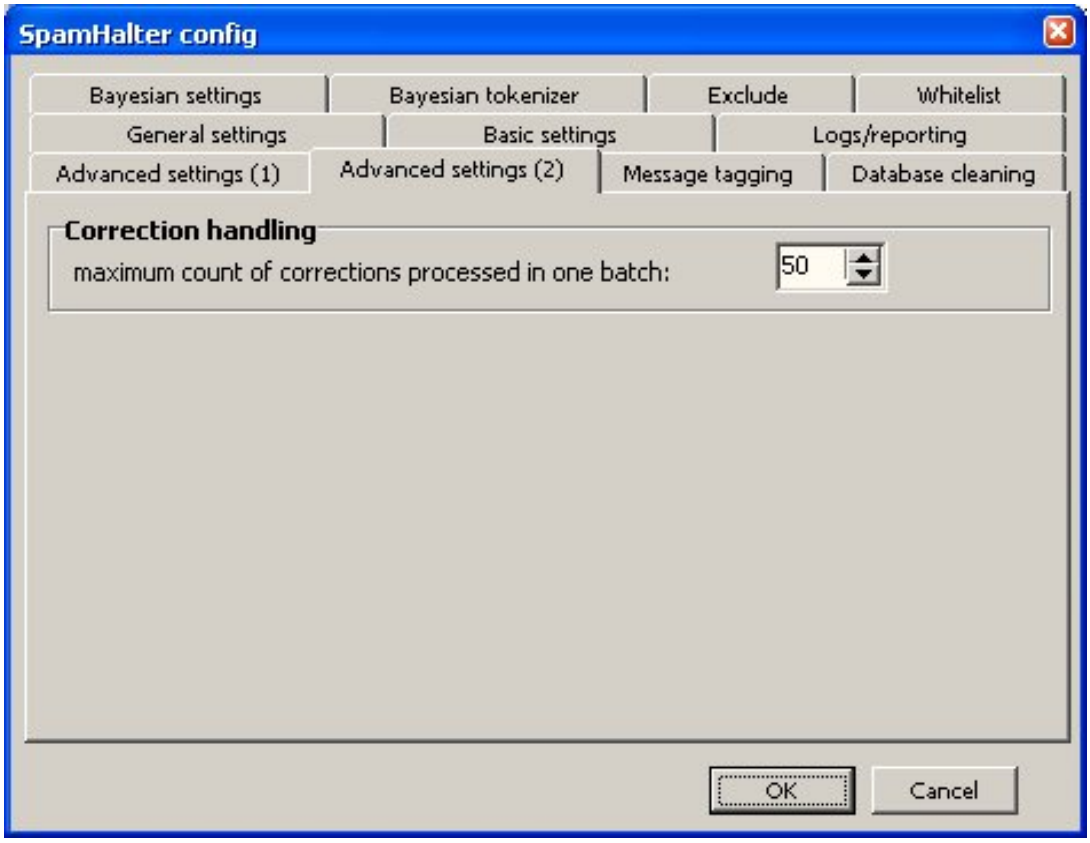
Is SPAM without classification

Honeypot: Set of honeypot e-mail addresses separated by commas. When at least one of the mail message recipients matches at least one address from this list of addresses, the message is automatically processed as spam. ***[SpamHalter] honeypot***

Block message with headers: When SpamHalter finds this named header as it processes a message, it marks this message as spam. This is good when you are using DNSBL or RBL in your SMTP. You can configure mercury/32 to add headers when the message is not passed by SMTP checking, typically by using 'x-blocked' header. ***[SpamHalter] BlockTag***

Virus handling

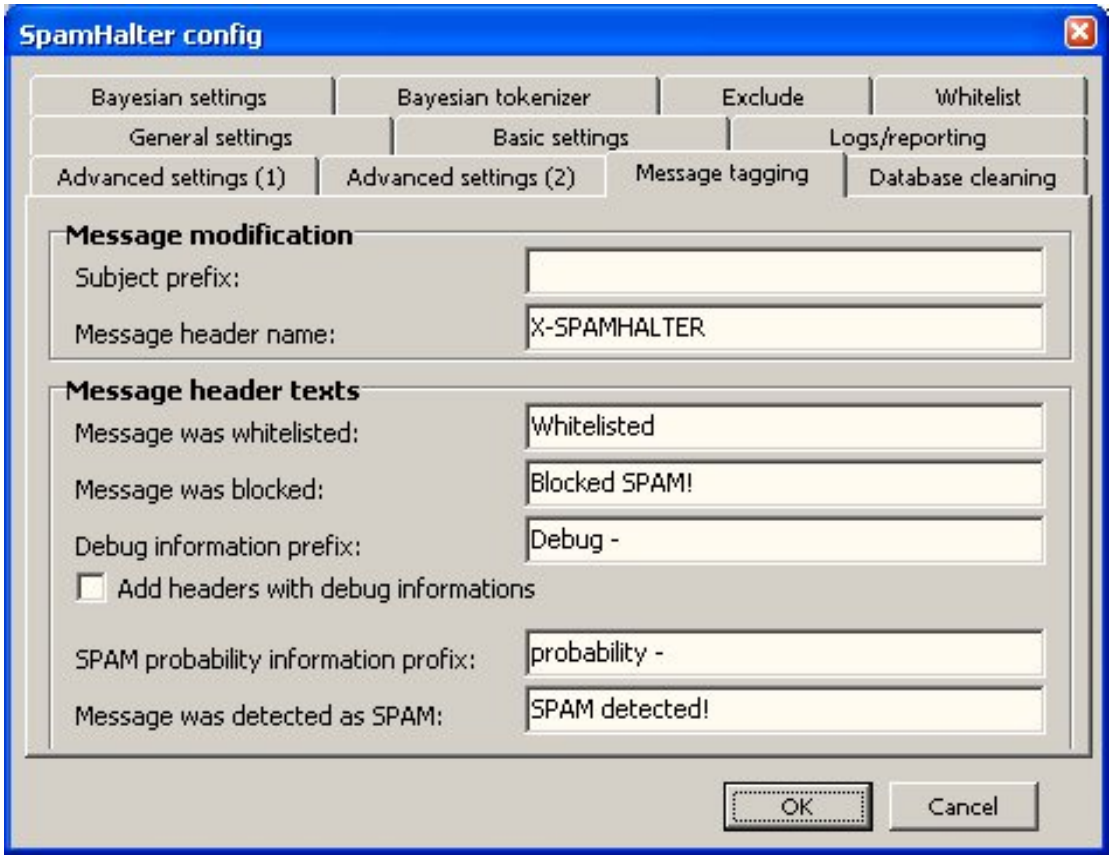
Virus header: When the string defined by this directive is found in the message headers, the message will be marked as SPAM and is not processed by SpamHalter. This changes statistical information only. Checking the block allows you to enable logging of virus messages to SpamHalter's log file. ***[SpamHalter] VirusTag, LogVirWall***



Correction handling

Maximum number of correction messages to process before processing another incoming email. Default is 50. Example: If you have 1000 messages in the correction account, (with default settings) SpamHalter will process the maximum of 50 correction messages. Now 950 messages remain in the correction account. It then processes the next message from Mercury/32 before processing the next 50 corrections, etc...

[bayDynamic] bayMaxCorrCnt



Message modification

Subject prefix: This defines the string that will be added to the beginning of the subject for SPAM messages. This string will be inserted into the subject within chars '[' and ']' automatically. When this string is empty (default), then the subject is not modified!

[SpamHalter] subject

Message header name: Message header name for SpamHalter information written to processed mails. Default is 'X-SPAMHALTER'.

[SpamHalter] tagname

Message header texts

Message was whitelisted: Here you can modify text that will be added to SpamHalter headers when the message is whitelisted. When this string is empty, SpamHalter will not generate this line to message headers. Default value is: 'Whitelisted'

[SpamHalter] WhitelistText

Message was blocked: Here you can modify text that will be added to SpamHalter headers when the message is blocked by SMTP DNSBL. When this string is empty, SpamHalter will not generate this line to message headers. Default value is: 'Blocked SPAM!'

[SpamHalter]Blocktext

Debug information prefix: Here you can modify text that will be added to SpamHalter headers with debug information. Default value is: 'Debug -' Check the block **Add headers with debug information** to enable writing these headers.

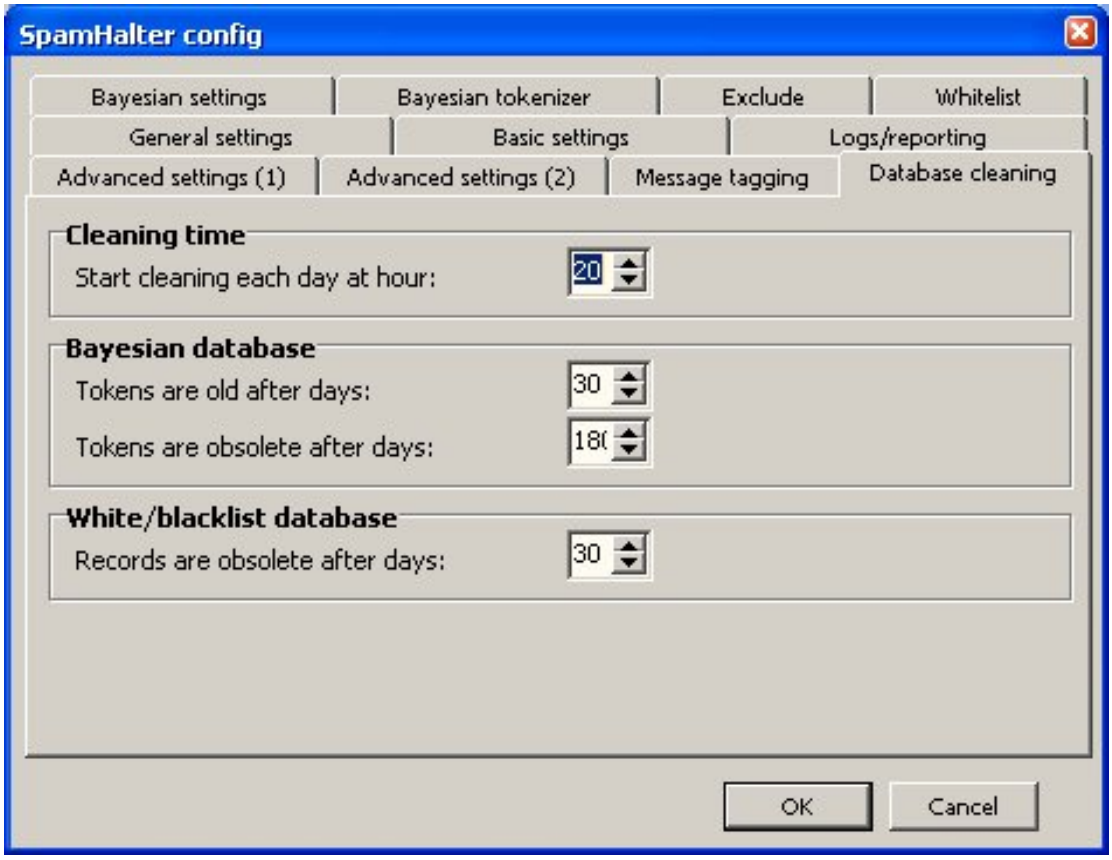
[SpamHalter] DebugText, BayDebug

Spam probability information prefix: Here you can modify text that will be added to SpamHalter headers with SPAM probability information. When this string is empty, SpamHalter will not generate this line to message headers. Default value is: 'probability -'

[SpamHalter] ProbText

Message was detected as SPAM: Here you can modify text that will be added to SpamHalter headers when the message is detected as SPAM. When this string is empty, SpamHalter not generate this line to message headers. Default value is: 'SPAM detected!'

[SpamHalter] SpamText



Cleaning time

The database cleaning task is run once a day. It runs automatically at the hour of day you specify. The cleaning task causes a delay of message processing by a few minutes, depending on your computer speed! Default is 20 – the first message from the internet after time 20:00 starts the cleaning task. ***[SpamHalter] CleanTime***

Bayesian database

Tokens are old after days: Words in database will be skipped over during the cleaning process until they have been there this many days. Default is 30 days.

[bayDynamic] bayOldDays

Tokens are obsolete after days: If a word in the database is unseen in messages longer than this number of days, it is deleted from the database. Default is 180 days. Be careful to not set this value too low, because it can purge your database of words!

[bayDynamic] bayExpire

White/blacklist database

When no outgoing mail has been sent to an e-mail address in the whitelist for longer than this many days, the address is deleted from the whitelist database. Default is 60 days.

[bayDynamic] bayWhiteOldDays

Corrections

No antispam protection is perfect! This means that sometimes it misses something. Because you are teaching SpamHalter, you must inform SpamHalter about its mistake. You can do this by forwarding the missed mail (forward it without edit!) to one of the correction addresses specified in the INI file. One address is for messages that are missed spams. The other address is for messages that are incorrectly marked as spam. If you get a missed spam message, forward it to the spam correction address. If you have a false positive message, forward it to the nospam correction address.

You cannot normally forward messages to corrections addresses from the internet. This is for your protection, because otherwise anybody could confuse your database! If your roaming users need to send corrections, they must use password protected corrections. See 'password' configuration directive above.

Whitelist/Blacklist

The basic rule for SpamHalter's spam battle is: "Do not talk with spammers!" When someone replies to an email, SpamHalter remembers this email address in the whitelist (unless it is in the blacklist). Any subsequent email coming from that address will not be considered spam because it will be whitelisted.

It works the other way, too. When you mark a message as spam, the sender is considered a spammer, and is added to the blacklist (and taken out of the whitelist).

SpamHalter's whitelist and blacklist are different than the whitelist and blacklist in Mercury/32! They are for internal SpamHalter use only and are temporary lists that expire after x days defined by the bayWhiteOldDays configuration directive.

Additions to the whitelist are managed automatically by each outgoing mail to internet. Only non-local e-mail addresses are added to the whitelist. Also, addresses are only added to the whitelist in this way (outgoing mail) when they are not in the blacklist.

The whitelist and blacklist are maintained by corrections, too. A correction to spam removes the address from the whitelist and adds it to the blacklist. A correction to nospam removes the address from the blacklist and adds it to the whitelist. All e-mail addresses in the message are processed (except the To field), not only in the headers, but in the message body, as well. You can use this feature for manually modifying the whitelist and blacklist. Just create a new message, write the desired e-mail addresses to the message body and send it to the spam or nospam correction account, depending on your desired operation.

If sender of incoming message exists in whitelist, message is processed as NoSpam. Blacklist just disabling automatic addition of blacklisted address into whitelist.

Testing

When you add '+spamtest' to the destination address (see '+' aliases in mercury), SpamHalter is turned on test mode. It does the spam classification of the message without database modifications. (Normally the database is updated after each test.) This is good for testing SpamHalter's processing. You can forward any messages to your self address and see SpamHalter result. (Example: send the email to youraddress+spamtest@yourdomain.com.)

SpamHalterTools

This is the database helper program. You can build databases, rebuild the word database, merge databases and run statistics. Use SpamHalterTools to create any needed database, if it does not yet exist.

You cannot use this program while Mercury/32 with SpamHalter is running!

SpamHalterUpgrade

This is the helper program to upgrade databases from SpamHalter 3.x.x only. For upgrading from older versions you must use SpamHalterTools from SpamHalter 3.x.x and upgrade database to SpamHalter 3 format first. Then you can upgrade this database to version 4 format with SpamHalterUpgrade.

You cannot use this program while Mercury/32 with SpamHalter is running!

Statistics

SpamHalter puts statistical information into the same directory where you have SpamHalter's databases.

Main statistical file is spamstat.txt. This is a continuous statistical file. All statistics are counted from the beginning of this file. When you delete this file, the statistics start over.

The second helper statistical file is spamstt2.txt. It is the same as spamstat.txt, but this file is automatically reset after StatRate (ini file configuration option) period. Before this file is deleted its content is written to spamstat.csv as the next line.

This spamstat.csv file is good for various analyses. It is a standard CSV file and you can import it to your popular spreadsheet and draw various graphs, etc.

SPAM detection

When SpamHalter detects spam, a special header specified in INI file (default is 'X-SPAMHALTER') with value 'SPAM detected!' is added to the message. (You can modify this text by the SpamHalter configuration file!). You can create a new mail filtering rule for automatically moving spam messages to a special folder. You can use regular expression filtering rule with search string like 'X- SPAMHALTER: SPAM*'.

Some programs do not have these filtering rules. In this case you can use the 'Subject' configuration option for enabling modifying of the subject line when SPAM is detected.

Performance tuning

Here are some tips for getting the best SpamHalter performance:

- Bayesian classification uses the database intensively. Keep your databases on a fast local disk!
- You get the best database performance when your operating system has a lot of free memory. Then your operating system can create a large file cache and fit all of SpamHalter's databases in memory!

- Install Mercury/32 on a stable computer and protect it with a UPS. Then you can disable database forced writes. It really boosts performance! (See bayForcedWrites ini option.)

Fixing of corrupted database

All database integrity errors can be repaired by a total rebuild of the word database. SpamHalter must not be running, of course!

1. Backup your words4.db3 file.
2. Create a dump by typing at command line: sqlite3.exe words4.db3 .dump > words4.txt
3. Delete or rename your old words4.db3 database.
4. Restore database from dump by typing: sqlite3.exe words4.db3 < words4.txt