

# Autofiler Documentation

Bart Vrijkorte

June 19, 2006



# Contents

<b>1 Quick HOWTO</b>	<b>5</b>
1.1 Prerequisites . . . . .	5
1.2 Installation . . . . .	5
<b>A The af_build_database program</b>	<b>7</b>
A.1 NAME . . . . .	7
A.2 SYNOPSIS . . . . .	7
A.3 OPTIONS . . . . .	7
A.4 DESCRIPTION . . . . .	7
A.5 SEE ALSO . . . . .	8
<b>B The af_deliver_message program</b>	<b>9</b>
B.1 NAME . . . . .	9
B.2 SYNOPSIS . . . . .	9
B.3 OPTIONS . . . . .	9
B.4 DESCRIPTION . . . . .	10
B.5 SEE ALSO . . . . .	10
<b>C The autofiler.conf file</b>	<b>11</b>
C.1 DESCRIPTION . . . . .	11
C.2 SEE ALSO . . . . .	12



# Chapter 1

## Quick HOWTO

In absence of detailed documentation we provide a quick howto for getting autofiler up and running.

### 1.1 Prerequisites

- A working perl installation.
- A working PostgreSQL installation.
- Several perl Modules. The Build script will complain if a certain module cannot be found. At least the Mail::Box, Set::Object and Tie::Hash cpan modules are required.

### 1.2 Installation

1. Extract the autofiler distribution tar.gz. file with `tar -zxf`.
2. Navigate to the extracted autofiler directory.
3. Execute `'perl Build.PL'`.
4. Execute `'./Build install'`.
5. Create a PostgreSQL database for each mailbox you want to manage with autofiler.
6. Use the psql client program to create the database schema. The `postgresql.sql` file contains the database schema.
7. Create a `/etc/autofiler.conf` file. An example file is provided in the distribution. In this file the database access is configured.

8. Configure your mail server so that deliveries go through the `af_deliver_message` program. This program reads a message from the standard input.

Caution: Autofiler e-mail deliveries can be resource intensive, you may want to make sure that not too many deliveries are started simultaneously.

9. (optional) create a cron job to automatically run the `af_build_database` program periodically for each mailbox.
10. run the `af_buid_database` program to train the software.

## Appendix A

# The af\_build\_database program

### A.1 NAME

af\_build\_database - Examines the specified mailbox and update the summary information in the configured database.

### A.2 SYNOPSIS

af\_build\_database [options] mailbox

Options:

`--help`                      brief help message.

### A.3 OPTIONS

`-help`

Prints a brief help message and exits.

### A.4 DESCRIPTION

This program will examine the specified mailbox and update the information in the configured database.

The mailbox path can be a filesystem path or a Mail::Box URL. An example of such an url could be:

maildir:/home/user/Maildir

## A.5 SEE ALSO

See also the documentation for the *Mail::Box* module found on CPAN.org for information about supported mail box formats.

See also the documentation entry for the *autofilter.conf* file format. In this file the database to be used is specified.



## Appendix B

# The af\_deliver\_message program

### B.1 NAME

af\_deliver\_message - Reads a message from standard input and delivers it into the appropriate folder in a mailbox.

### B.2 SYNOPSIS

af\_deliver\_message [options] mailbox

Options:

--help	brief help message.
--minscore %	minimum confidence percentage required for a decision.
--minlead %	minimum confidence lead in percent points for a decision.
--pretend	don't actually do anything.
--rationale n	add a header to the message with debug info.
--print-tokens	print each encountered token to standard output.
--print-results	print the probability of each folder to standard output.

### B.3 OPTIONS

**-help**

Prints a brief help message and exits.

**-minlead %**

Minimum lead in percent points the winning folder needs to have over the other folders for a decision to be made. Defaults to 10%. Higher values will result in less false positives and more false negatives.

**-pretend**

Don't actually deliver the message, only show where we would deliver it. Useful for debugging.

**-rationale n**

Autofiler can add a header to the mail message that details why a message was placed in a certain folder. The optional argument specifies the maximum number of lines of text. This defaults to 0 (no rationale).

**-print-tokens**

This option causes the program to print to standard output each token that was encountered in the message. This option could be combined with the `-pretend` option to omit actual delivery.

**-print-results**

This option causes the program to print to standard output the probability that the message belongs in each of the known folders in the database. This option could be combined with the `-pretend` option to omit actual delivery.

## B.4 DESCRIPTION

This program will read a message from standard input and analyse it. It will then be delivered to the appropriate folder in the mailbox.

The mailbox path can be a filesystem path or a Mail::Box URL. An example of such an url could be:

```
maildir:/home/user/Maildir
```

## B.5 SEE ALSO

See also the documentation for the *Mail::Box* module found on CPAN.org for information about supported mail box formats.

See also the documentation entry for the *autofiler.conf* file format. In this file the database to be used is specified.

# Appendix C

## The autofiler.conf file

autofiler.conf - Configuration file for autofiler

### C.1 DESCRIPTION

The autofiler.conf file is a plain text file that contains the configuration information for autofiler. The file is parsed by the different autofiler programs.

### LOCATION

The configuration file is searched in several standard locations. The search ends when the configuration file is found. The following locations are searched in order:

- `$HOME/.autofiler/config`
- `/etc/autofiler.conf`
- `/usr/local/etc/autofiler.conf`
- `/etc/autofiler/autofiler.conf`
- `/usr/local/etc/autofiler/autofiler.conf`

### FILE FORMAT

Options are specified in 'key=value' notation. Lines that start with a hash (#) mark are considered comments. The following environment variables are automatically substituted in the configuration file:

**\$HOME** The home directory for the user that currently executes an autofiler program.

**\$USER** The username of the user that currently executes an autofiler program.

In the autofiler distribution an example configuration file is provided.

## SUPPORTED OPTIONS

The following options are recognized:

**backend** This option specifies the database backend module to use. Currently the only supported backend is 'PostgreSQLBackend' for accessing PostgreSQL databases.

**dbi** Autofiler uses the perl DBI database interface module. With this option the DBI connection string is specified. The string should usually specify the database access module to use and the name of the database.

For example: 'dbi=dbi:Pg:dbname=\$USER' connects to the PostgreSQL database with the name of the current user.

**username** This specifies the username that is used for the database connection.

**password** This option specifies the password to be used for the database connection. Use of this option is discouraged as configuration files are usually world-readable. Use the `password_file` option instead if a database password needs to be specified.

**password\_file** This option can be used to point at a file that contains one line of text that specifies the password to be used for the database connection. The option's value can contain the `$USER` environment variable to specify different passwords for different users.

When neither this option nor the `password` option is specified, autofiler will prompt for a password if it is needed.

**maximum\_age** This option specifies the maximum age of a mail message for it to be considered by the `af_build_database` program. The value is specified as the number of days since the message's arrival.

## C.2 SEE ALSO

See also the documentation for the `Mail::Box` module for information about supported mail box formats.

See also the documentation for the autofiler programs `af_build_database` and `af_deliver_message`.

See also the documentation entry for the *autofiler.conf* file format. In this file the database to be used is specified.