

Jaro Mail

by Jaromil @ dyne.org

Sept 2013

Contents

1	Introduction	1
1.1	Features	2
1.2	Vision	3
2	Diagram	5
3	Setup	6
3.1	Build	6
3.2	Install	6
4	Configuration	6
4.1	Receive mail	6
4.2	Send mail	6
4.3	Filter mail	7
5	Organization	7
5.1	Folders	7
5.2	Whitelist	7
5.3	Blacklist	8
5.4	Addressbook	8
5.5	Organization In Brief	9
6	Workflow	9
6.1	Fetch and read your mail at home	9
6.2	Write a new mail	9
6.3	Reply messages	10
6.4	Peek without downloading anything	10
6.5	Send emails whenever possible	10
6.6	Save important emails for later	10
6.7	Workflow in brief	11
7	Searching	11
8	Security	12
8.1	Password storage	12
8.2	Temporary directory	12
8.3	A tip for GNU/Linux users	13

9	Storage and backup	13
9.1	Merge maildir	13
9.2	Remove duplicates from maildir	14
9.3	Backup mails older than	14
9.4	Filter a maildir	14
10	Usability tips	14
10.1	Quickly send a file via email on Apple/OSX	14
11	Acknowledgements	15
11.1	License	15
11.2	Jaro Mail credits	15
11.3	Mutt credits	16
11.4	Mairix credits	16
11.5	Fetchmail credits	16
11.6	Procmail credits	17
11.7	MSmtp credits	17
11.8	Statistics modules	17
12	Appendix	17
12.1	Configuration examples	17
12.1.1	Accounts/imap.default	17
12.1.2	Accounts/smtp.default	18
12.1.3	Filters.txt	19

1 Introduction

Jaro Mail is an integrated suite of interoperable tools to manage e-mail communication in a private and efficient way, without relying too much on on-line services, in fact encouraging users to store their email locally.

Rather than reinventing the wheel, this suite reuses existing free and open source tools and protocols and is mainly targeted for GNU/Linux/BSD desktop usage.

This manual illustrates the usage of Jaro Mail. The newest version of this manual is made available on <http://files.dyne.org/jaromail/jaromail-manual.pdf>

1.1 Features

```

0 1 Jun 02 Ian (0.3K) [Bricolabs] XXXO arts and technology festival
0 2 May 14 Jean-Noël Montagné (0.5K) [Bricolabs] Incredible Edible, Bricolabs-style initiative
N 3 May 14 Sonja van K (4.2K) '->
0 4 May 12 victoria sinclair (1.1K) [Bricolabs] technoshamanism chat room live now! sat 12th may
0 5 May 08 victoria sinclair (2.6K) [Bricolabs] technoshamanism chatroom - saturday
->[P 6 May 08 Gustaff Harriman Is (5.5K) '->
7 May 07 Felipe Fonseca (1.8K) [Bricolabs] all watched over
0 8 May 09 kiilo (3.8K) '->
0 9 May 09 James Wallbank (4.4K) '->
0 10 May 09 august (5.9K) | '->
0 11 May 09 John Hopkins (0.8K) | '->
0 12 May 09 victoria sinclair (5.1K) | | '->
0 13 May 09 F B (7.6K) | '->
14 May 08 e3jehron (4.3K) '->
0 15 May 08 Armin Medosch (2.8K) '->
0 16 May 04 venzha christ (0.7K) [Bricolabs] MICRONATION/MACRONATION 2012 - HONF project
0 17 May 04 stephen kovats (1.9K) '->
0 18 May 05 venzha christ (7.8K) '->
0 19 May 04 Paula Vélez (2.7K) [Bricolabs] Call for this summer. France. Festival hack&DIY A Pado Loup, 12 au 22 aout 2012, dans les Alpes
0 20 May 06 bronac@boundaryobje (0.2K) '->Re: [Bricolabs] water aqua l'eau
0 21 May 01 Jake Harries (10K) [Bricolabs] Open Call - Access Space Artist Residencies 2012 at Refab Space
22 May 01 Carsten Agger (1.8K) [Bricolabs] Alternatives to money
23 May 01 James Wallbank (1.6K) [Bricolabs] Research into Physical Computing
24 May 01 victoria sinclair (9.7K) '->
0 25 May 01 bronac@boundaryobje (0.9K) | '->Re: [Bricolabs] Event in Middle England
0 26 May 01 victoria sinclair (6.7K) | '->
0 27 May 01 bronac@boundaryobje (0.7K) | '->
28 May 01 Matt Ratto (2.7K) '->
29 Apr 30 yasir ياسر (1.6K) [Bricolabs] Sunflower Guerrillas: Plant Some Sunshine
30 Apr 30 yasir ياسر (9.7K) '->
+ 31 Apr 26 venzha christ (18K) Invitation: Mini Symposium at LAF
32 Apr 26 stephen kovats (9.2K) [Bricolabs] Fwd: [spectre] MICRONATION/MACRONATION - HONF project 2012
33 Apr 26 atteqa@gmail.com (9.6K) '->Re: [Bricolabs] Fwd: [spectre] MICRONATION/MACRONATION - HONF project 2012
34 Apr 25 victoria sinclair (11K) '->Re: [Bricolabs] Fwd: [spectre] MICRONATION/MACRONATION - HONF project 2012
35 Apr 26 venzha christ (16K) '->
36 Apr 11 pata de Perro (7.3K) [Bricolabs] Fwd: [La Siera en Peligro] Dear Friends, Activists, and Environmentalists,
r 37 Apr 08 Patrice Riems (2.8K) [Bricolabs] Mr Paw and the Power that Be (was: well, something like the power that Be ... ; -)
F 38 Apr 08 To Bricolabs (2.4K) '->
39 Apr 08 bronac@boundaryobje (0.3K) '->
r 40 Apr 08 Mr.Paw (29K) '->
O F 41 Apr 10 To Bricolabs (1.2K) | '->
42 Apr 08 Mr.Paw (33K) | '->Re: [Bricolabs] Mr Paw and the Power that Be
43 Apr 08 Patrice Riems (0.3K) '->
44 Apr 08 Mr.Paw (6.6K) '->
45 Mar 27 Rob van Kranenburg (22K) [Bricolabs] [Fwd: Re: [Ticket#2012032710000126] Gesuspende site]
46 Mar 28 atteqa@gmail.com (1.8K) |>
47 Mar 28 Rob van Kranenburg (1.4K) | '->
---Mutt: =dyne.bricolabs [Msgs:4137 New:1 Old:1468 Flag:11 Inc:23 41M]---(threads/reverse-date)-default------(1%)---

```

- Minimalistic and efficient interface with message threading
- Targets intensive usage of e-mails and mailinglists
- Stores e-mails locally in a reliable format (maildir)
- Integrates whitelisting and blacklisting, local and remote
- Can do search and backup by advanced expressions
- Automatically generates filter rules (procmail, sieve)
- Imports and exports VCard contacts to its addressbook
- Computes and shows statistics on mail traffic
- Encrypts password storage (using keyrings)
- Provides advanced maildir management tools (rmdupes, backup)
- Defers connections for off-line operations
- Checks SSL certificates over (imap, smtp)
- Supports strong encryption messaging (GnuPG)
- Multi platform: GNU/Linux/BSD, Apple/OSX
- Old school, used by its author for the past 10 years

1.2 Vision

The internet offers plenty of free services, on the wave of the Web2.0 fuzz and the community boom, while all private informations are hosted on servers owned by global corporations and monopolies.

It is important to keep in mind that no-one else better than you can ensure the privacy of your personal data. Server hosted services and web integrated technologies gather all data into huge information pools that are made available to established economical and cultural regimes.

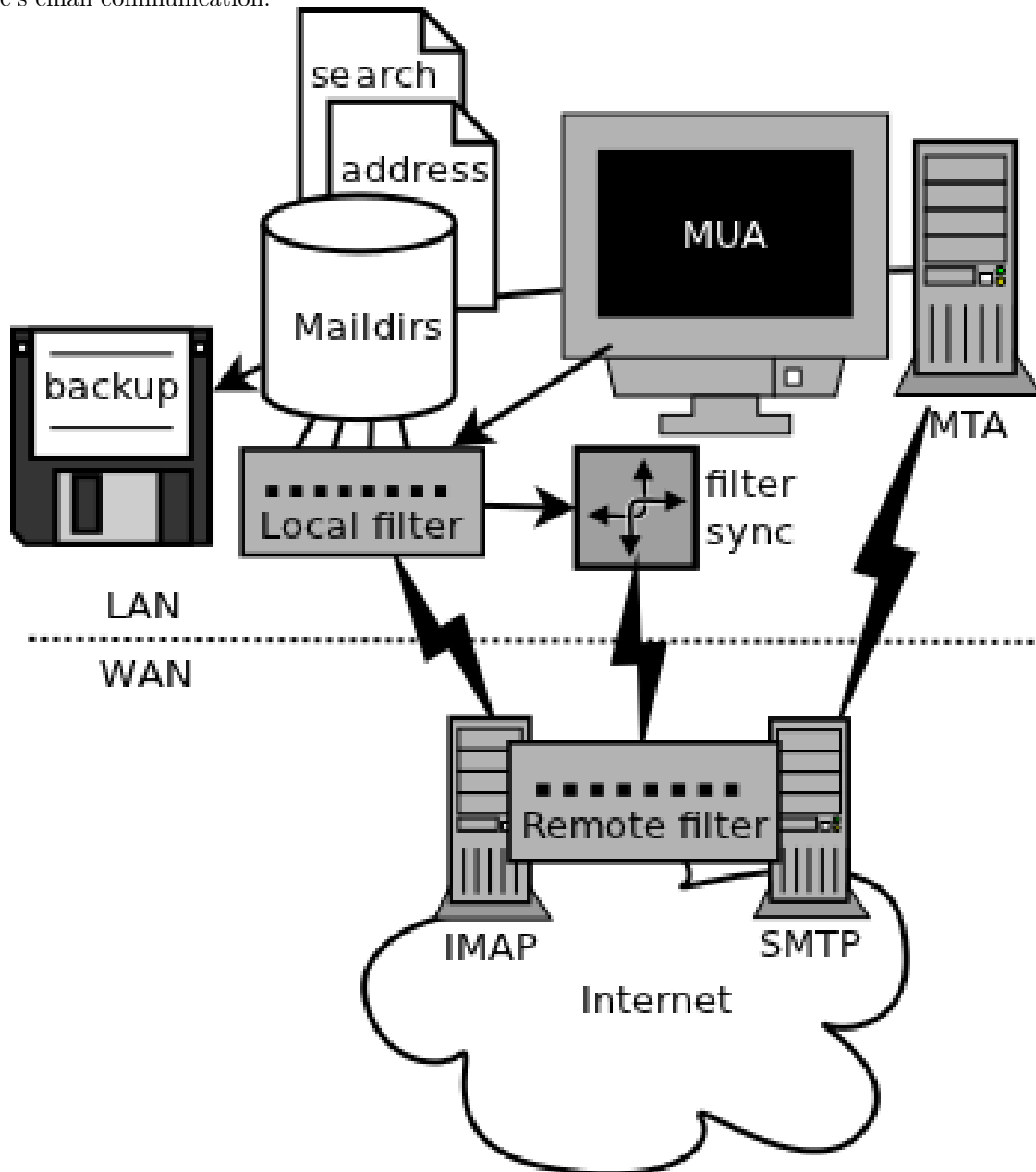
The vision behind this software is that of sharing a simple and consistent way to operate e-mail communication with tools that are available on most platforms and can be as well used remotely over a secure shell connection.

Jaro Mail aims to facilitate the task of downloading and storing e-mail archives off-line in a way that they can be still accessible in more than 10 years time and independently of any software. Nowadays many users have the habit of keeping all their e-mails on servers, accessing them through an often clumsy web interface, while downloading them can free space and improve their privacy.



2 Diagram

A little diagram that clarifies a bit where do we place the components and actions involved in managing one's email communication:



Acronym	Function	Software
MUA	Mail User Agent	Mutt
MTA	Mail Transport Agent	Fetchmail
SMTP	Filtering Agent	Procmail
	Mail Delivery Agent	MSmtp
	Search engine	Mairix
	Addressbook	ABook

3 Setup

3.1 Build

There are two **build-** scripts coming with the sourcecode of Jaro Mail and they are both found in the **build/** subdirectory.

One is for building it on GNU/Linux systems, the other for Apple/OSX.

A compiler and various dependencies are mandatory in order to build this software. The version for GNU/Linux so far supports only Debian systems and will use apt-get (requiring sudo access) to install all needed packages.

Apple/OSX users that have no experience in building software can obtain a pre-built universal binary from our download zone on <http://files.dyne.org/jaromail/binary>

3.2 Install

Installing Jaro Mail once all dependencies are build is fairly easy: the **install.sh** script provided will create a working environment in your **\$HOME/Mail** directory or, if you like, you can specify a different one as an argument:

```
$ cd jaromail
$ ./install.sh $HOME/Mail
```

Every installation of Jaro Mail is fully reentrant, meaning the directory where it gets installed contains all maildirs, configurations, filters, whitelist and other necessary files.

Installing it in different directories a single user can have multiple Jaro Mail installations to permit the complete separation of E-Mail identities.

If called from outside the installation directory, the **jaro** command will use the default path **\$HOME/Mail**. If one is using a different installation path then should first change to that directory and then use the command from inside it.

4 Configuration

The place where Jaro Mail is installed (**\$HOME/Mail** by default) contains all configuration files.

For Apple/OSX users, this directory is inside their **\$HOME/Library** and then **Application Support/JaroMail**.

From now on, we will call this place the **Mail directory**.

4.1 Receive mail

Inside the Mail directory is found the folder **Accounts** with brief instructions and default templates to fill with Imap or Pop account configurations to fetch mail. A default template will be found in fresh installations: **Accounts/imap.default.txt**. The configuration can be edited with one's favourite text editor, the format of the file is pretty self-explanatory.

It is possible to have more than one account and in fact when retrieving e-mails using the **jaro fetch** command all accounts will be processed, unless one is explicitly selected using the **-a** commandline option.

4.2 Send mail

Also inside the **Accounts** directory is found the **smtp.default.txt** file that configures how to send mail.

When no special account is specified using the **-a** option, then the file named **smtp.default.txt** into the Accounts folder will be used. To fix a selection it is also possible to use a symbolic link.

Mutt users might want to customize settings by creating a **Mutt.txt** file inside the Mail directory: it will be included automatically if present and can contain any mutt configuration directives for the mail transport agent, such as custom headers appearing in composed e-mails and the default GPG¹ key to be used when signing and encrypting them.

4.3 Filter mail

In the mail directory a file named **Filters.txt** can be created and filled in with rules referencing the contents of the **From:** or **To:** fields of each e-mail that is fetched. The mails matching will be then saved in the indicated maildirs (created if not existing) to keep a bit of order, especially useful for mailinglist users.

The format of the filters configurarion is pretty easy and self explanatory, an example is found in the appendix of this manual.

5 Organization

One of the main goals for Jaro Mail is to organize the e-mail workflow so that one's attention is dedicated to important communications, rather than being constantly distracted by various degrees of spam and the need to weed it out of the mailbox. This ambitious task is pursued by realizing an integrated approach consisting of flexible whitelisting and the distinction between mails from known people and the rest.

5.1 Folders

First lets start with a categorization of the standard maildirs and a brief description for each. This information is **very important** to understand how Jaro Mail works: these maildirs are standard in Jaro Mail, here they are listed in order of priority

Folder	What goes in there
known	Mails whose sender is known (Whitelist)
priv	Unknown sender, we are the explicit destination
unsorted	Unknown sender, we are in cc: or somehow reached
unsorted.ml	From a mailinglist that we haven't filtered yet
zz.blacklist	Mails whose sender is not desired (Blacklist)

The advantage using such a folder organization is that every time we open up the mail reader we will be presented with something we are likely to be most interested in (known people replying our mails) and progressively, as we will have the time to scroll through, mails from “new people” or mass mailings of sort. Please note this organization does not includes spam, which is supposedly weeded out on the server via spamlists: White/Blacklisting has more to do with our own selection of content sources than with the generic protection from random pieces of information.

5.2 Whitelist

The way whitelisting works if quite crucial to this setup and, at the same time, is fairly simple since it does not include any automatic detection, learning filters, Markov chains or Bayesian A/I. We believe the user should be in full control of prioritizing communication channels and at the same time constantly able to tweak the setup in an easy way.

To whitelist an address is sufficient to send it an e-mail: at the moment the message is sent Jaro Mail will remember the destination address and prioritize all messages coming back from it. This we call implicit whitelisting.

¹GPG stands for GNU Privacy Guard, a system to securely encrypt and decrypt messages and files so that noone can read their content, even when intercepting the communication.

To explicitly whitelist an address from inside the mail reader index press [**a**] while selecting an email, this will add in the whitelist the sender address (From: header). If you want to add all addresses reached by the mail (From: To: and Cc: fields) use the same letter capitalized pressing shift [**A**].

All addresses selected this way will have the privilege of ending up in your **known/** folder, plus their name and e-mail will be completed automatically when composing a new email, pressing the **Tab** key while indicating them among the recipients.

5.3 Blacklist

To blacklist an address instead one can use the [**z**] key while an e-mail is selected on the index: the sender indicated in the From: field will be downgraded to the very bottom of your priorities, closes to spam than the rest, the most infamous **zz.blacklist/** folder.

5.4 Addressbook

What we call addressbook here basically consists of both the whitelist and the blacklist. We store both lists in a unique database file in the Mail dir called **Addressbook**². In future we may add similar support for other addressbook formats that people use (GnuPG keyring, Evolution etc.)³

Both the white and blacklist can be edited using a text interface, this way you can delete entries for instance (removing them from the whitelist or blacklist), or add entries by hand (for instance manually from visit cards), as well you can change details directly (name and email). To edit the addressbook in Jaro Mail the **abook** command is available

```
$ jaro abook
```

This will open the current whitelist for edit, but one can append “blacklist” to this commandline to open that one instead.

To quickly dump to the console all names and addresses in the Jaro Mail addressbook, one can use the **list** command

```
$ jaro list
```

To search a string across the addressbook, simply use the command search followed by a string, for instance:

```
$ jaro search autistici
```

will list all addresses @autistici in your addressbook.

Even more useful to interface with other addressbook software and mobile phones, you can use the **export** and **import** functions to transport your addressbook formatted as a series of VCards⁴.

```
$ jaro export
```

Will create or update the file in **Mail/jaro/addressbook.vcf**. On the other hand, the import command will include all entries found in a given VCard file that have at least one email address.

```
$ jaro import 00001.vcf
```

Imports into the whitelist all contacts found in the 00001.vcf file. The VCard format is fully compatible with import and export of contacts in Android mobile phones.

Apple Mac/OSX users can simply run the import command without any arguments

```
$ jaro import
```

Imports all the contacts found in the system addressbook used by Mail.app, hence making them whitelisted.

²Jaro Mail uses sqlite3 as its own database storage

³On Apple/OSX systems Jaro Mail has access to the Mail.app addressbook, so all contacts known in your Mac will be automatically whitelisted

⁴A file format standard for electronic business cards, see: <http://en.wikipedia.org/wiki/VCard>

5.5 Organization In Brief

Below a recapitulation of keys related to the white and blacklisting functionality, to be used in the e-mail index or when an e-mail is open inside the mail user agent:

List	Key	Function	Fields
White	a	Add the sender address	From:
White	A (shift)	Add all addresses	From: To: Cc:
Black	z	Blacklist the sender	From:

And here the addressbook commands that are available from the commandline:

Command	Function
abook	edit the addressbook
list	list the addressbook
search	search a name or address
export	export to a VCard file
import	import from a VCard file

6 Workflow

This section goes through a scenario of simple usage for Jaro Mail

6.1 Fetch and read your mail at home

As you acces your computer where Jaro Mail has been configured, you can open a Terminal and type:

```
$ jaro fetch
```

This will download all new mails.

If you have configured **fetchall** among the imap account options, then will delete them from the server, freeing online space.

If you have configured the **keep** option, which is the default, Jaro Mail will only download the email that you have not yet read and in any case it won't delete anything from the server.

```
$ jaro
```

This will open the first folder containing unread emails, starting from the **known** folder, then **priv**, then all the rest.

From there on, pressing **=** or **c** you can change the folder and explore your **priv** folder, the mailinglist ones as configured by your Filters.txt, as well your **unsorted** mails.

6.2 Write a new mail

If you like to write a mail to someone, just write his/her own address as an argument to Jaro Mail

```
$ jaro compose friend@home.net
```

But if you don't remember the email of your friend, then you can just start **jaro compose** without options, then start typing the name or whatever you remember of it: pressing the **Tab** key a completion will help to remind what you are looking for, offering a list of options to choose from.

If you are writing an email with attachments (and you are sure their size is reasonably small to be circulated via email) you can launch Jaro Mail with files as arguments, or even wildcards, and they will be automatically set as attachments, you can then specify its recipients

```
$ jaro picture01.jpg jingle02.mp3 ~/myicons/*
```

Will send a mail with various separate attachments (using MIME encapsulation): a picture, an hopefully small audio file and a list of icons which are all the files contained into the `myicons/` directory.

After composing the email you will be able to review all those attachments and eventually remove some of them by hand: move up and down across them and then click [**D**] to remove the selected one.

6.3 Reply messages

While browsing through the index of emails in various folders, one can reply any of them just by pressing the [**r**] key, which will ask if the original message should be quoted and then open your favorite editor to compose your text.

If the email you are replying has been sent to multiple recipients (for instance using multiple addresses in the Cc: or From: fields) they will all be included, but you will have the possibility to exclude them by hand editing those fields.

It is also possible to forward a message to someone else than the sender, for instance to submit it to his or her attention, or that of a mailinglist. To do that, you can use the [**f**] key which will present you with the full message and the possibility to write something on top of it, to describe its contents to its new recipients.

6.4 Peek without downloading anything

If you are around and like to see your new mails without downloading them, then you can use the **peek** function:

```
$ jaro peek
```

This will open the default configured IMAP account and folder over SSL protocol (securing the data transfer) and show your emails.

From peek you can reply and even delete emails, but be careful since what you delete here will be removed from the server and won't be there when you download it from home.

This functionality can be also very useful if you are from a slow connection and need to delete some email that is clogging it and that you are not able to download because of its size.

6.5 Send emails whenever possible

All the time you write an E-mail, Jaro Mail will save it into your outbox folder, waiting for the right moment to send it. In fact you will have to tell it the right moment by running the **send** command:

```
$ jaro send
```

This will authenticate with your SMTP and send all your emails to destination. This way even if you are off-line you will always be able to write emails and eventually bring them around for sending them whenever possible.

6.6 Save important emails for later

Sometimes one can be on the rush while reading emails (local or via imap) and flagging them as important can be useful to keep focus on priorities. In some cases it is very useful to save such important messages locally for later reference, for instance in a folder keeping messages that need to be remembered and that will constitute a kind of TODO list (a'la GTD).

Jaro Mail implements such functionalities: by pressing the [**F**] key (shift-f) one can flag an email, which will turn bright-green in the index. In addition to that there is a folder called **remember/** where one can copy emails on the fly using the [**R**] key (shift-r) any time. Messages will be duplicated into the remember folder (which of course can be opened with the command **jaro remember**) so they can also be edited with annotations on the task they refer to, for instance using the [**e**] key, without affecting the original message.

6.7 Workflow in brief

Below a recapitulation of keys commonly used in our workflow

Key	Function
m	Compose a new message
Tab	Complete addresses and folders input
r	Reply to the sender of a message
g	Group reply to all recipients
f	Forward a message to new recipients
=	List all filtered maildir folders
c	Change to another folder
F	Flag a message as important
R	Copy a message to remember
s	Move a message to another folder
C	Copy a message to another folder

7 Searching

Searching across all your emails it is as important as demanding of a task. Jaro Mail implements it using Mairix, a portable program written by a bunch of talented programmers in portable C language.

After the indexing is done, you can use the command **jaro search** followed by any number of arguments to run the search.

If one of the arguments given to the search command is the name of an existing email directory folder in ~/Mail, then the search will be on contents of the folder.

More than one word is aloud to refine the match (they are all AND'ed together), plus a number of tricks can be done: every single word following the command can be a particular expression that indicates in which header to search and for what.

Here below a short reference of possible expressions:⁵

word	match word in message body and major headers
t:word	match word in To: header
c:word	match word in Cc: header
f:word	match word in From: header
a:word	match word in To:, Cc: or From: headers (address)
s:word	match word in Subject: header
b:word	match word in message body
m:word	match word in Message-ID: header
n:word	match name of attachment within message
F:flags	match on message flags (s=seen,r=replied,f=flagged,-=negate)
p:substring	match substring of path
d:start-end	match date range
z:low-high	match messages in size range
bs:word	match word in Subject: header or body (or any other group of prefixes)
s:word1,word2	match both words in Subject:
s:word1/word2	match either word or both words in Subject:
s:~word	match messages not containing word in Subject:
s:substring=	match substring in any word in Subject:
s: <i>substring</i> =	match left-anchored substring in any word in Subject:
s:substring=2	match substring with <=2 errors in any word in Subject:

⁵For a reference on how the date range works in search expressions, you can look into the **Backup** section in this manual.

If none of the arguments is an email folder existing in `~/Mail` then the search will be run over addressbook whitelist entries, returning addresses of found contacts.

8 Security

8.1 Password storage

Our MUA (Mutt) and our MTA (Fetchmail) normally required the user to input the email account password every time or write it clear inside a plain text file, jeopardizing the secrecy of it.

But most desktops nowadays have a keyring that stores passwords that are often used during a session, saving the user from retyping them every time.

Jaro Mail provides an interesting (and long awaited) feature even for those who are already using Mutt for their email: **it stores passwords securely**. This is done in different ways depending from the operating system is being running on.

Jaro Mail will use the default keyring whenever present to store all new passwords for each account used: the first time will prompt for a password and, while using it, will save it in relation to the particular account. This way the user can simply authenticate into the keyring at login and, while managing such sensitive informations using OS specific tools, Jaro Mail can be launched without the tedious task of a password input every time e-mails are being checked.

On **Apple/OSX** the default internal keyring is being used.

On **GNU/Linux** gnome-keyring is supported if found, else JaroMail will revert to use its own encrypted⁶ database called **keyring**. Every time a password will be retrieved or saved, the keyring password will be asked.

To explicitly change a password one can operate the default keyring manager or use the command **jaro passwd -a imap.default** (or other accounts) which will prompt to set for a new password even if an old one is known.

8.2 Temporary directory

For its password management system to work, Jaro Mail often requires to write down passwords in clear text, at least until software like Fetchmail and Mutt is updated to avoid such a stupid need.

The way we overcome this limitation is by using a temporary directory, making sure that all sensitive files created in it are deleted as soon as possible, as well that no other user on the system has access to them, but still we can't deny that an administrator access them.

If a ramdisk is present on the system it will be used by Jaro Mail: that is a “volatile” directory in RAM whose contents are never written on the disk. This prevents an intruder to seize the machine and recover deleted data from unused sectors on the hard-disk, because all data saved in RAM gets irremediably lost after approximately 2 minutes the machine is switched off for such an operation.

On **Apple/OSX** systems to enable this feature one must explicitly activate the ramdisk using the command

```
$ jaro ramdisk open
```

This will create and mount `/Volume/JaroTmp` which is 10MB large and will be used for our delicate security transactions.

On **GNU/Linux** systems this is done automatically if the shared memory volume is available and writable (`/dev/shm`) without the need to use the ramdisk command.

For the aforementioned reasons of writing passwords in clear text, Jaro Mail also requires the use of safe deletion techniques as those provided by **srnm** (on Apple/OSX) and **wipe** (on GNU/Linux) every time a file is deleted. So even if a ramdisk is not activated it will be very hard if not impossible for an attacker to retrieve information from hard-disk sectors or using a cold-boot attack on RAM.

⁶The keyring is encrypted using weak symmetric encryption via GnuPG, the only protection for the data inside then is the password memorized by the user.

8.3 A tip for GNU/Linux users

Those using a GNU/Linux system might want to have a look at our other software **Tomb, the Crypto Undertaker**⁷ which takes care of quick mount and umount of an encrypted volume when desired and includes a **hook** mechanism to automatize the execution of commands to make a directory inside the encrypted volume immediately available in the user's home.

Using a light combination of scripts between Jaro Mail and Tomb is possible to achieve a strong level of personal security, definitely above the average.

For more information about Tomb please refer to its own documentation.

9 Storage and backup

Most existing e-mail systems have their own storage format which is often over-complicated and forces us to convert our archives to it.

Jaro Mail stores emails using the well documented format **Maildir** which is common to many other free and open source e-mail software and was developed and well documented by D.J. Bernstein.

We can safely say that the Maildir format to store e-mails will stay the same and well compatible in 10 years from now, if not more, mostly because of its simplicity: that's what we need the most from a storage format after all.

Quoting him about the wonders of this format:

Why should I use maildir?

Two words: no locks. An MUA can read and delete messages while new mail is being delivered: each message is stored in a separate file with a unique name, so it isn't affected by operations on other messages. An MUA doesn't have to worry about partially delivered mail: each message is safely written to disk in the tmp subdirectory before it is moved to new. The maildir format is reliable even over NFS.⁸

What this virtuous, sometimes very cryptical man is trying to say here is that the Maildir format in its simplicity of implementation represents an extremely reliable way to retrieve and store emails without the risk of losing any if the Internet connection goes down.

While skipping over the internal details of this storage system, which basically consists in plain text files saved into sub-directories, we will have a look at some very interesting features that Jaro Mail can offer to its users and to the even larger audience of Maildir format users.

9.1 Merge maildir

Jaro Mail can safely merge two different maildirs basically gathering all e-mails stored in them into a unique place. This is done using two arguments, both maildir folders: the first is the source and the second is the destination e-mails from both will be gathered:

```
$ jaro merge ml.saved-mails ml.global-archive
```

The above command will move all emails stored inside the maildir folder "ml.saved-mails" to the other maildir folder "ml.global-archive". Upon success the first argument "ml.saved-mails" will be deleted and all its contents will be found in "ml.global-archive".

⁷<http://tomb.dyne.org>

⁸<http://cr.yp.to/proto/maildir.html>

9.2 Remove duplicates from maildir

As a result of a merge or a multiple fetch of e-mails, it can often occur that a maildir contains duplicates which are also highlighted in red in the e-mail index and, if many, can be tedious to eliminate by hand. Jaro Mail offers the automatic functionality of removing all duplicate emails from a maildir folder using the **rmdupes** command:

```
$ jaro rmdupes ml.overflow
```

Will look for all duplicates emails in the “ml.overflow” maildir, matching them by their unique **Message-Id:** header and a SHA1 hash of their content⁹, and delete all duplicates for mails that are present more than once.

9.3 Backup mails older than

To facilitate the separate storage of e-mails that are too old to be of any interest, but still might be useful to be retrieved just in case, Jaro Mail implements a function that will move all messages older than a certain date out of a maildir folder into another.

```
$ jaro backup ml.recent ml.yearsago d:5y-1y
```

The above command will move out of the “ml.recent” maildir all e-mails that are older than 1 year (up to 5 years before, can be more) and stores them into the “ml.yearsago” maildir which for instance could be present on an external usb hard-disk or any other backup device, helping us to save space on the desktop in use.

```
jaro backup unsorted d:may98-may99 unsorted.week.old
```

Will move all emails found in any folder that are dated between May 1998 and May 1999. Here below more examples of date range expressions: d:2002-2003 d:may2002-2003 d:2002may-2003 d:feb98-15may99 d:feb98-15may1999 d:2feb98-1y d:02feb98-1y d:970617-20010618

9.4 Filter a maildir

If filters are updated or one desires to import a maildir into Jaro Mail processing it through its filters, the **filter** command is provided to (re)filter a maildir.

```
$ jaro filter my-old-maildir
```

Beware that filtering is a lengthy operation, especially on big maildirs: it will first pass all messages found through your filters, refiling them to folders (even creating duplicates) and then prune all the affected folders to remove the duplicates.

10 Usability tips

10.1 Quickly send a file via email on Apple/OSX

To right-click on a file and send it via email attach using Jaro Mail you must create a “Service” using the application “Automator”. It is fairly simple:

1. Start Automator
2. Choose the Service template

⁹The standard utility ‘formail -D’ is used for this operation

3. In the dropdown boxes that appear choose “files or folders” and “Finder”
4. Look for “Run Applescript” in the Library tree
5. Drag “Run Applescript” in the workflow area and paste this script into it:

```
on run {input, parameters}
    tell application "Terminal"
        activate
        tell window 1
            do script "~/Mail/jaro/bin/jaro " & POSIX path of input
        end tell
    end tell
end run
```

Now Save the new service (you can name it “Send file via Jaro Mail”) and when you will right click on a file, in the submenu “Services” you will find the option you just scripted, which will open a Terminal asking you the email address, while the file will be already configured as attach.

11 Acknowledgements

Jaro Mail would have never been possible without the incredible amount of Love shared by the free and open source community, since it is relying on the development of software like Procmail, Mutt, Fetchmail and even more code which is included and used by this program.

Heartfelt thanks go to all those contributing code and sharing it to make the world a better place by not letting down all users in the hands of corporate non-sense and proprietary technologies and protocols.

This manual is written and maintained by Jaromil who is also the one who wrote the Jaro Mail scripts. Still he is far from being the person that wrote most of the code running here, just the one who organized it in an hopefully intuitive way for users.

In the following chapters the best is done in order to credit most people that contributed to free and open source software that Jaro Mail makes use of.

11.1 License

The following copyright notice applies to this manual, the software included is licensed under the same or different GNU GPL or BSD licenses, or available in the public domain.

Copyright (C) 2010-2012 Denis Roio <jaromil@dyne.org>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; Permission is granted to make and distribute verbatim copies of this manual page provided the above copyright notice and this permission notice are preserved on all copies.

11.2 Jaro Mail credits

Jaro Mail is written and maintained by Denis Roio (aka Jaromil) it started from the intention to share his own 10 years old e-mail setup, encouraged by the geek tradition of exchanging configuration files between friends.

The RFC 822 address parser (fetchaddr) is originally written by Michael Elkins for the Mutt MUA.

The gateway to Apple/OSX addressbook (ABQuery) was written by Brendan Cully and just slightly updated for our distribution.

Special thanks go to Alvise Gottieri, Anatole Shaw, Francesco Politi and Fabio Pietrosanti for early testing and debugging.

11.3 Mutt credits

Please note that this is by no means an exhaustive list of all the persons who have been contributing to Mutt. Please see the manual for a (probably still non complete) list of the persons who have been helpful with the development of Mutt. Our special thanks go to Antonio Radici, the Mutt maintainer in Debian, for his suggestions and encouragement.

Copyright (C) 1996-2007 Michael R. Elkins <me@cs.hmc.edu>
Copyright (C) 1996-2002 Brandon Long <blong@fiction.net>
Copyright (C) 1997-2008 Thomas Roessler <roessler@does-not-exist.org>
Copyright (C) 1998-2005 Werner Koch <wk@isil.d.shuttle.de>
Copyright (C) 1999-2009 Brendan Cully <brendan@kublai.com>
Copyright (C) 1999-2002 Tommi Komulainen <Tommi.Komulainen@iki.fi>
Copyright (C) 2000-2004 Edmund Grimley Evans <edmund@rano.org>
Copyright (C) 2006-2008 Rocco Rutte <pdmef@gmx.net>

11.4 Mairix credits

Jaro Mail includes a search engine for e-mails that is also licensed GNU GPL v2. Here below the names of the copyright holders and all those who have written it:

Copyright (C) Richard P. Curnow 2002,2003,2004,2005,2006,2007,2008
Copyright (C) Sanjoy Mahajan 2005
Copyright (C) James Cameron 2005
Copyright (C) Paul Fox 2006

Mairix received contributions from: Anand Kumria André Costa, Andreas Amann, Andre Costa, Aredridel, Balázs Szabó, Bardur Arantsson, Benj. Mako Hill, Chris Mason, Christoph Dworzak, Christopher Rosado, Chung-chieh Shan, Claus Alboege, Corrin Lakeland, Dan Egnor, Daniel Jacobowitz, Dirk Huebner, Ed Blackman, Emil Sit, Felipe Gustavo de Almeida, Ico Doornekamp, Jaime Velasco Juan, James Leifer, Jerry Jorgenson, Joerg Desch, Johannes Schindelin, Johannes Weißl, John Arthur Kane, John Keener, Jonathan Kamens, Josh Purinton, Karsten Petersen, Kevin Rosenberg, Mark Hills, Martin Danielsson, Matthias Teege, Mikael Ylikoski, Mika Fischer, Oliver Braun, Paramjit Oberoi, Paul Fox, Peter Chines, Peter Jeremy, Robert Hofer, Roberto Boati, Samuel Tardieu, Sanjoy Mahajan, Satyaki Das, Steven Lumos, Tim Harder, Tom Doherty, Vincent Lefevre, Vladimir V. Kisil, Will Yardley, Wolfgang Weisselberg.

11.5 Fetchmail credits

Fetchmail is also licensed GNU GPL v2

Copyright (C) 2002, 2003 Eric S. Raymond
Copyright (C) 2004 Matthias Andree, Eric S. Raymond, Robert M. Funk, Graham Wilson
Copyright (C) 2005 - 2006, 2010 Sunil Shetye
Copyright (C) 2005 - 2010 Matthias Andree

11.6 Procmail credits

Procmail was originally designed and developed by Stephen R. van den Berg.

In the fall of 1998, recognizing that he didn't have the time to maintain procmail on his own, Stephen created a mailing list for discussion of future development and deputized Philip Guenther as a maintainer.

The included Procmail library collection is developed and maintained by Jari Aalto.

11.7 MSmtplib credits

MSmtplib is developed and maintained by Martin Lambers.

You can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

11.8 Statistics modules

We are including some (experimental, still) modules for statistical visualization using JQuery libraries. The first module inspiring us to implement such a functionality is Timecloud, then other modules followed.

Timecloud is Copyright (C) 2008-2009 by Stefan Marsiske
Dual licensed under the MIT and GPLv3 licenses.

TagCloud version 1.1.2
(c) 2006 Lyo Kato <lyo.kato@gmail.com>
TagCloud is freely distributable under the terms of an MIT-style license.

ExCanvas is Copyright 2006 Google Inc.
Licensed under the Apache License, Version 2.0 (the "License");

jQuery project is distributed by the JQuery Foundation under the terms of either the GNU General Public License (GPL) Version 2.

The Sizzle selector engine (which is included inside the jQuery library) is held by the Dojo Foundation and is licensed under the MIT, GPL, and BSD licenses.

jQuery.sparkline 2.0 is licensed under the New BSD License

Visualize.JQuery is written by Scott Jehl
Copyright (c) 2009 Filament Group
licensed under MIT (filamentgroup.com/examples/mit-license.txt)

12 Appendix

12.1 Configuration examples

12.1.1 Accounts/imap.default

```
# Name and values are separated by spaces or tabs
# comments start the line with a hash

# Name appearing in From: field
name To Be Configured
```

```

# Email address (default is same as login)
email unknown@gmail.com

# Aliases also received on this mail
# alias mimesis@gmail.com
# alias nemesis@gmail.com

# Internet address
host imap.gmail.com

# Username
login USERNAME@gmail.com

# Authentication type
auth plain # or kerberos, etc

# Identity certificate: check or ignore
cert ignore

# Transport protocol
transport ssl

# Service port
port 993

# Options when fetching
# to empty your mailbox you can also use: fetchall
# by default this is 'keep' which will not delete mails from server
options keep
# we encourage you to store emails locally, hence using a fetchall
# configuration from a machine that you keep at home and secured.

# Imap folders
# uncommment to provide a list of folders to be fetched
# folders INBOX, known, priv, lists, unsorted, unsorted.ml

```

12.1.2 Accounts/sntp.default

```

# Name and values are separated by spaces or tabs
# comments start the line with a hash

# Name for this account
name To Be Configured

# Internet address
host smtp.gmail.com

# Username
login USERNAME@gmail.com

# Transport protocol
transport ssl # or "tls" or "plain"

```

```
# Service port
# port 465
port 25
```

12.1.3 Filters.txt

```
# Example filter configuration for Jaro Mail
```

```
# mailinglist filters are in order of importance
# syntax: to <list email> save <folder>
# below some commented out examples, note the use of a prefix,
# which makes it handy when browsing with file completion.
```

# Field	String match	Folder in Mail/
to	crypto@lists.dyne	save dyne.crypto
to	dynebolic	save dyne.dynebolic
to	freej	save dyne.freej
to	frei0r-devel	save dyne.frei0r
to	taccuino	save ml.freaknet
to	deadpoets	save ml.freaknet
to	linux-libre	save gnu.linux-libre
to	foundations@lists	save gnu.foundations
to	debian-mentors	save debian.mentors
to	debian-blends	save debian.blends
to	freedombox-discuss	save debian.freedombox

```
# Other filters for web 2.0 using folder names with a prefix:
# they can facilitate folder maintainance.
```

# Field	String match	Folder in Mail/
from	identi.ca	save web.identica
from	Twitter	save web.twitter
from	linkedin	save web.linkedin
from	googlealerts	save web.google
from	facebook	save web.facebook
from	FriendFeed	save web.friendfeed
from	academia.edu	save web.academia