



Published on Free Software Magazine (<http://www.freesoftwaremagazine.com>)

Running GNU Mailman at home

Putting the most powerful mailing list manager on your desktop

By David A. Harding

GNU Mailman is the most popular free software mailing list manager, and probably the most configurable; however, it normally requires you have a web and mail server always connected to the Internet. With a little extra work, you can run Mailman from your intermittently-connected GNU/Linux desktop.

Introduction

Normally Mailman's workflow looks like figure 1. A mail server receives email, gives it to Mailman for processing, takes it back from Mailman, and delivers the email to the final recipients. The desktop Mailman workflow adjustments (see figure 2) add a few extra steps: mail is forwarded to, stored in, and downloaded from a mailbox before being received by a desktop mail server. These extra three steps are easy to setup, and this article integrates them into the normal Mailman setup procedure.

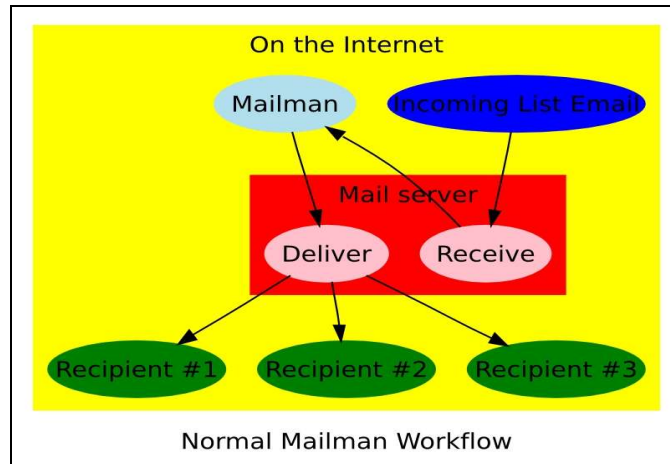


Figure 1: Mailman's Normal Workflow

Running GNU Mailman at home

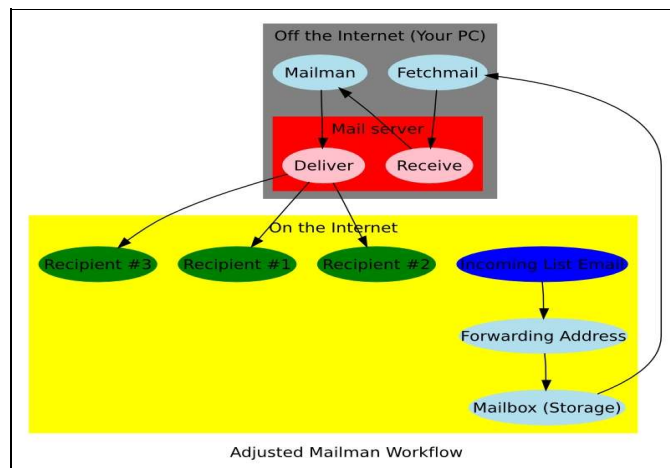


Figure 2: Mailman's Adjusted Workflow

Finding the right type of mailbox

Your desktop mailing list requires a mailbox to store messages on the Internet. Most ISPs provide one or more mailboxes with their standard service. Several websites, like Google, provide no-cost mailboxes as well. Choose freely: you can change mailbox addresses without notifying users or breaking the mailing list.

Your choice of mailboxes is limited by one constraint: a non-interactive program, Fetchmail, needs to retrieve the email from your mailbox. Fetchmail supports two protocols: the Post Office Protocol (POP) and the Internet Message Access Protocol (IMAP). The mailboxes available from most ISPs and Google Mail support at least one of these protocols.

If you already use a mailbox that meets these criteria, don't use it with these instructions; it will only complicate things. Create a brand new mailbox for your mailing list.

Getting a forwarding Address

Select a name for your mailing list. I recommend a short and descriptive name. A list called *doe* created by Jane Doe for communicating with her family will be used throughout the article as an example.

Mailman requires three email addresses for each mailing list. Each address must be at the same domain and must follow a particular format: `list@domain`, `list-request@domain`, and `list-owner@domain`. Replace *list* with the name of your list and replace *domain* with the name of your domain. Jane Doe, who knows people use her as an example, owns the domain `example.com`; the three email addresses she'll use for her *doe* list are `doe@example.com`, `doe-request@example.com`, and `doe-owner@example.com`.

All three of the addresses need to be forwarding address; all must forward to the mailbox previously setup (see the previous section of this article). Jane Doe's domain name registrar lets her configure forwarding addresses. If your domain name registrar doesn't provide forwarding service, or if you don't have a domain name, you can use a free forwarding service (try a Google search for one) or a paid forwarding service, like the one that comes with a Free Software Foundation associate membership.

Configuring a mail server

Mailman needs a properly configured mail server in order to function. If you already have a working mail server, skip to the next section.

Running GNU Mailman at home

The original mail server, Sendmail, is free software, but is so difficult to configure that some people wrote simpler mail servers. As the simpler mail servers added features to match Sendmail, they too became difficult to configure. The free software community is now filled with powerful and difficult to configure mail servers, and none of them is standard on all free operating systems.

Among GNU/Linux distributions, the Sendmail, Postfix, and Exim mail servers are very common. You need to read the documentation on setting up your operating system's mail server, but I'll lay out the general settings you need for Mailman and give instructions for Postfix.

Mail servers perform two functions: sending and receiving email. Sending email was once easy: any mail server could talk to any other mail server. Then Microsoft Windows made email difficult. Through poor security, millions of computers are commandeered and used to send spam; today, many mail servers reject email from residential computers like your desktop. Instead, they only accept email from well-known computers. For example, an ISP is likely to only accept mail from its broadband customers.

Your ISP probably has a well-known mail server and you're probably permitted to use it. Configure your mail server to use your ISP's mail server as a "smarthost"—a server smart enough to route your email. Check the documentation on your ISP's website for outgoing email instructions. Where you see the word SMTP, look for a hostname like `smtp.isp.net`—that's the name of the smarthost. Now check how to configure your local mail server so that it uses your ISP's smarthost.

In Postfix, you will add this to the file `/etc/postfix/main.cf`:

```
relayhost = 207.126.122.88
```

Remember that mail servers often need to be restarted. If you're using postfix, you can either run `killall -HUP postfix` or `postfix reload`.

After configuring your mail server to use a smarthost, send a test email using the following command:

```
echo "To: you@yourisp.net" | /usr/lib/sendmail -t
```

`/usr/lib/sendmail` is a legacy of when Sendmail was the only mail server; it works with all of the mail servers installed by default on major GNU/Linux distributions. Replace `you@yourisp.net` with your usual email address.

If you receive the email at your usual address, your mail server is correctly configured for sending. If not, find the problem and fix it. I suggest starting by checking your mail logs (usually `/var/log/maillog`).

Configuring your mail server to receive mail for a specified email address is easy, since most mail servers are preconfigured so that they are ready to receive. Test the receiving mechanism by adding the following line to `/etc/aliases`:

```
testing123: you@yourisp.net
```

In Postfix, Remember to run the `newaliases` command after changing `/etc/aliases`. To test, replace `you@yourisp.net` with your usual email address and run the two following commands:

```
echo "To: testing123" | /usr/lib/sendmail -t
```

If you receive the email in your regular email account, continue; if you don't, check the logs and

documentation for your mail server.

Downloading email

Now that you can send and receive email, you need to be able to fetch the email that you will then send to your mailing list. The Fetchmail program creates a connection between your mailbox and your mail server. It downloads the email from your mailbox and gives it to your mail server as if it had been delivered to your mail server directly.

Fetchmail requires configuration. It needs to know the address of the POP or IMAP server your mail is stored on, and it needs your login name and password too. To find the POP or IMAP server name, check the documentation on your mailbox provider's website; look for instructions for downloading email; the hostname near the POP or IMAP setting is the server name.

Fetchmail needs to run periodically. If you install Fetchmail through some distributions, like Debian or Ubuntu, you will need to configure `/etc/default/fetchmail` (see below) and then start fetchmail by typing: `/etc/init.d/fetchmail start`. Users of other distributions should read the documentation that comes with Fetchmail to discover the best way to periodically run it.

For example, Jane Doe uses Debian and configures Fetchmail by adding the following lines to the file `/etc/fetchmailrc`:

```
poll mail.isp.net with proto POP3
  user 'jane' there with password 'secret1'
  to
    'doe@example.com'='doe'
    'doe-request@example.com'='doe-request'
    'doe-owner@example.com'='doe-owner'
    '*'='jane'
  here options fetchall
```

Use the following table to replace Jane's settings with your own. The settings are listed in the table in the order they appear in the file:

Jane's Setting	Replace With
mail.isp.net	The mail server name
POP3	Use IMAP for IMAP servers
jane	Your user name on the mail server
secret1	Your password on the mail server
doe@example.com	Your list address and first forwarding address
doe	Your list name and local variant of the first forwarding address
doe-request@example.com	The second forwarding address
doe-request	The local variant of the second forwarding address
doe-owner@example.com	The third forwarding address
doe-owner	The local variant of the third forwarding address
jane	Your username on the local computer

Setting Replacement Table

Running GNU Mailman at home

Fetchmail splits the email in your mailbox into four parts: what goes to your mailing list (for example, `doe`), what goes to Mailman (`doe-request`), what goes to the list administrator (`doe-owner`), and everything else (`jane`).

Test Fetchmail by adding the following three lines to `/etc/aliases` before running `newaliases`. Replace *listname* with the name of your list and *you@yourisp.net* with your regular email address.

```
listname: you@yourisp.net
listname-request: you@yourisp.net
listname-owner: you@yourisp.net
```

Use your regular email client and account to send one email to each of the three forwarding address; place the To address in the subject line. If you receive all three emails back, Fetchmail is configured correctly; if not, check Fetchmail's log in `/var/logs/mail.log`. Fetchmail only downloads email once every five minutes, and you won't receive the emails until Fetchmail downloads them.

After a successful test, remove the three lines added to `/etc/aliases` and re-run `newaliases`.

Setting up a web server

Mailman's primary configuration interface is an HTML form. The form requires a web server. On your desktop, install the Apache web server (either version one or two), and test the server by loading the following URL in your web browser: <http://localhost/>

If you see a web page describing itself at the Apache default web page, Apache is successfully configured and Mailman will be able to use it.

Setting up Mailman

Install Mailman from your operating system's package manager (alternatively, you can download the source code from list.org and follow the installation instructions in the file `admin/www/mailman-install.pdf` in the source package).

Once Mailman is installed, Create a new mailing list by running the `newlist` command as root and following the instructions it gives you. Note that you may have to provide the full path to the `newlist` script, which in Debian and Ubuntu is `/usr/lib/mailman/bin/newlist`.

Mailman works by adding specific lines to `/etc/aliases` that forward all email from your three forwarding addresses to Mailman; then Mailman examines each of the emails for instructions and acts on those instructions. Usually, this means mail is sent out again; Mailman uses your mail server to send it. Note that you may have to add those specific lines to `/etc/aliases` yourself.

You can configure Mailman by using your web browser to access the following URL: <http://localhost/cgi-bin/mailman/admin>

You need to change one setting on Mailman's main configuration page for your list: the `host_name` variable should be sent to the host name for your forwarding addresses (whatever follows the `@` symbol in the forwarding addresses; in Jane Doe's case, `example.com`).

Testing the list

Don't test your list by subscribing to it using Mailman's web form. Subscribe to your list using the email interface: send an email to your list in the same format Jane Doe used, but use your email and your list's `-request` email addresses instead:

```
From: jane@janes-isp.net
To: doe-request@example.com
Subject: subscribe
```

The body of the message will be disregarded by Mailman.

If your mailing list accepts new subscriptions, you'll receive an email from Mailman requesting confirmation before adding your email address to the list; reply to it and you'll be your mailing list's first subscriber. Try posting a test message to your list; use Jane Doe's test message as an example:

```
From: jane@janes-isp.net
To: doe@example.com
Subject: Testing
```

I'm testing the family mailing list.

-Jane

A few seconds or minutes after you send the email, you should receive it in your inbox: your mailing list works—now put it to work.

Spread the word

Jane Doe created her list for family communication. She manually adds the email addresses of her immediate family members to Mailman as subscribers using the membership management options in Mailman's web configuration interface. But Jane isn't sure if her distant relatives want to subscribe, and so she'll tell them how to subscribe by email: "send an email to `doe-request@example.com` with the subject `subscribe`. If you need any help, ask me."

You can encourage other people to join your list by placing those two sentences, adjusted for your list, in an email on a web page. By default Mailman lets anyone join your list, but you can disable that in the Mailman web administration panel by toggling the option `subscribe_policy` on the Privacy Options page.

After you've configured one list using Mailman, adding more lists is easy: create three new forwarding addresses pointing to the same mailbox, add three corresponding rules to the `/etc/fetchmailrc` file, and run the `newlist` command. By repeating these last instructions, your desktop Mailman can host thousands of lists.

Good luck!

Biography

[David A. Harding](/user/43640) (/user/43640" title="View user profile.): David A. Harding frequently writes about free software. He has a Linux Professional Institute level 1 certification and a small dog.

Copyright information

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is available at <http://www.gnu.org/copyleft/fdl.html>.

Source URL:

http://www.freesoftwaremagazine.com/articles/running_gnu_mailman_at_home
