GNU Mailman - List Administration Manual

Release 2.1

Barry A. Warsaw

March 2, 2015

Abstract

This document describes the list administrator's interface for GNU Mailman 2.1. It contains information a list owner would need to configure their list, either through the web interface or through email. It also covers the moderator's interface for approving held messages and subscription notices, and the web interface for creating new mailing lists. In general, it does not cover the command line interface to Mailman, installing Mailman, or interacting with Mailman from the point of view of the user. That information is covered in other manuals.

Contents

1 Introduction to GNU Mailman

GNU Mailman is software that lets you manage electronic mailing lists. It supports a wide range of mailing list types, such as general discussion lists and announce-only lists. Mailman has extensive features for controlling the privacy of your lists, distributing your list as personalized postings or digests, gatewaying postings to and from Usenet, and providing automatic bounce detection. Mailman provides a built-in archiver, multiple natural languages, as well as advanced content and topic filtering.

Mailman provides several interfaces to its functionality. Most list administrators will primarily use the web interface to customize their lists. There is also a limited email command interface to the administrative functions, as well as a command line interface if you have shell access on the Mailman server. This document does not cover the command line interface; see the GNU Mailman site administrator's manual for more details.

1.1 A List's Email Addresses

Every mailing list has a set of email addresses that messages can be sent to. There's always one address for posting messages to the list, one address that bounces will be sent to, and addresses for processing email commands. For example, for a mailing list called *mylist@example.com*, you'd find these addresses:

- mylist@example.com – this is the email address people should use for new postings to the list.
- mylist-join@example.com – by sending a message to this address, a new member can request subscription to the list. Both the Subject: header and body of such a message are ignored. Note that mylist-subscribe@example.com is an alias for the -join address.
- mylist-leave@example.com – by sending a message to this address, a member can request unsubscription from the list. As with the -join address, the Subject: header and body of the message is ignored. Note that mylist-unsubscribe@example.com is an alias for the -leave address.
- mylist-owner@example.com – This address reaches the list owner and list moderators directly.
- mylist-request@example.com – This address reaches a mail robot which processes email commands that can be used to set member subscription options, as well as process other commands.
- mylist-bounces@example.com – This address receives bounces from members whose addresses have become either temporarily or permanently inactive. The -bounces address is also a mail robot that processes bounces and automatically disables or removes members as configured in the bounce processing settings. Any bounce messages that are either unrecognized, or do not seem to contain member addresses, are forwarded to the list administrators.
- mylist-confirm@example.com – This address is another email robot, which processes confirmation messages for subscription and unsubscription requests.

There's also an -admin address which also reaches the list administrators, but this address only exists for compatibility with older versions of Mailman.

1.2 Administrative Roles

There are two primary administrative roles for each mailing list, a list owner and a list moderator. A list owner is allowed to change various settings of the list, such as the privacy and archiving policies, the content filtering settings, etc. The list owner is also allowed to subscribe or invite members, unsubscribe members, and change any member's subscription options.

The list moderator on the other hand, is only allowed to approve or reject postings and subscription requests. The list moderator can also do things like clear a member's moderation flag, or add an address to a list of approved non-member posters.

Normally, the list owner and list moderator are the same person. In fact, the list owner can always do all the tasks a list moderator can do. Access to both the owner's configuration pages, and the moderation pages are protected by the same password. However, if the list owner wants to delegate posting and subscription approval authority to other people, a separate list moderator password can be set, giving moderators access to the approval pages, but not the configuration pages. In this setup, list owners can still moderate the list, of course.

In the sections that follow, we'll often use the terms list owner and list administrator interchangably, meaning both roles. When necessary, we'll distinguish the list moderator explicitly.

1.3 A List's Web Pages

Every mailing list is also accessible by a number of web pages. Note that the exact urls are configurable by the site administrator, so they may be different than what's described below. We'll describe the most common default configuration, but check with your site administrator or hosting service for details.

Mailman provides a set of web pages that list members use to get information about the list, or manage their membership options. There are also list archive pages, for browsing an online web-based archive of the list traffic. These are described in more detail in the GNU Mailman user's manual.

Mailman also provides a set of pages for configuring an individual list, as well as a set of pages for disposing of posting and subscription requests.

For a mailing list called *mylist* hosted at the domain *lists.example.com*, you would typically access the administrative pages by going to `http://lists.example.com/mailman/admin/mylist`. The first time you visit this page, you will be presented with a login page, asking for the list owner's password. When you enter the password, Mailman will store a session cookie in your browser, so you don't have to re-authenticate for every action you want to take. This cookie is stored only until you exit your browser.

To access the administrative requests page, you'd visit `http://lists.example.com/mailman/admindb/mylist` (note the *admindb* url as opposed to the *admin* url). Again, the first time you visit this page, you'll be presented with a login page, on which you can enter either the list moderator password or the list owner password. Again, a session cookie is dropped in your browser. Note also that if you've previously logged in as the list owner, you do not need to re-login to access the administrative requests page.

1.4 Basic Architectural Overview

This section will outline the basic architecture of GNU Mailman, such as how messages are processed by the sytem. Without going into lots of detail, this information will help you understand how the configuration options control Mailman's functionality.

When mail enters the system from your mail server, it is dropped into one of several Mailman *queues* depending on the address the message was sent to. For example, if your system has a mailing list named *mylist* and your domain is *example.com*, people can post messages to your list by sending them to *mylist@example.com*. These messages will be dropped into the *incoming* queue, which is also colloquially called the *moderate-and-munge* queue. The incoming queue is where most of the approval process occurs, and it's also where the message is prepared for sending out to the list membership.

There are separate queues for the built-in archiver, the bounce processor, the email command processor, as well as the outgoing email and news queues. There's also a queue for messages generated by the Mailman system. Each of these queues typically has one *queue runner* (or "qrunner") that processes messages in the queue. The qrunners are idle when there are no messages to process.

Every message in the queues is represented by two files, a message file and a metadata file. Both of these files share the same base name, which is a combination of a unique hash and the Unix time that the message was received. The metadata file has a suffix of '`.db`' and the message file has a suffix of either '`.msg`' if stored in plain text, or '`.pck`' if stored in a more efficient internal representation[1].

As a message moves through the incoming queue, it performs various checks on the message, such as whether it matches one of the moderation criteria, or contains disallowed MIME types. Once a message is approved for sending to the list membership, the message is prepared for sending by deleting, adding, or changing message headers, adding footers, etc. Messages in the incoming queue may also be stored for appending to digests.

2 The List Configuration Pages

After logging into the list configuration pages, you'll see the configuration options for the list, grouped in categories. All the administrative pages have some common elements. In the upper section, you'll see two columns labeled "Configuration Categories". Some categories have sub-categories which are only visible when you click on the category link. The first page you see after logging in will be the "General Options" category. The specific option settings for each category are described below.

On the right side of the top section, you'll see a column labeled "Other Administrative Activities". Here you'll find some other things you can do to your list, as well as convenient links to the list information page and the list archives. Note the big "Logout" link; use this if you're finished configuring your list and don't want to leave the session cookie active in your browser.

Below this common header, you'll find a list of this category's configuration variables, arranged in two columns. In the left column is a brief description of the option, which also contains a "details" link. For many of the variables, more details are available describing the semantics of the various available settings, or information on the interaction between this setting and other list options. Clicking on the details link brings up a page which contains only the information for that option, as well as a button for submitting your setting, and a link back to the category page.

On the right side of the two-column section, you'll see the variable's current value. Some variables may present a limited set of values, via radio button or check box arrays. Other variables may present text entry boxes of one or multiple lines. Most variables control settings for the operation of the list, but others perform immediate actions (these are clearly labeled).

At the bottom of the page, you'll find a "Submit" button and a footer with some more useful links and a few logos. Hitting the submit button commits your list settings, after they've been validated. Any invalid values will be ignored and an error message will be displayed at the top of the resulting page. The results page will always be the category page that you submitted.

2.1 The General Options Category

The General Options category is where you can set a variety of variables that affect basic behavior and public information. In the descriptions that follow, the variable name is given first, along with an overview and a description of what that variable controls.

[1] Specifically, a Python pickle

General list personality

These variables, grouped under the general list personality section, control some public information about the mailing list.

real_name Every mailing list has both a *posting name* and a *real name*. The posting name shows up in urls and in email addresses, e.g. the `mylist` in `mylist@example.com`. The posting name is always presented in lower case, with alphanumeric characters and no spaces. The list's real name is used in some public information and email responses, such as in the general list overview. The real name can differ from the posting name by case only. For example, if the posting name is `mylist`, the real name can be `MyList`.

owner This variable contains a list of email addresses, one address per line, of the list owners. These addresses are used whenever the list owners need to be contacted, either by the system or by end users. Often, these addresses are used in combination with the `moderator` addresses (see below).

moderator This variable contains a list of email addresses, one address per line, of the list moderators. These addresses are often used in combination with the `owner` addresses. For example, when you email `mylist-owner@example.com`, both the owner and moderator addresses will receive a copy of the message.

description In the general list overview page, which shows you every available mailing list, each list is displayed with a short description. The contents of this variable is that description. Note that in emails from the mailing list, this description is also used in the comment section of the To: address. This text should be relatively short and no longer than one line.

info This variable contains a longer description of the mailing list. It is included at the top of the list's information page, and it can contain HTML. However, blank lines will be automatically converted into paragraph breaks. Preview your HTML though, because unclosed or invalid HTML can prevent display of parts of the list information page.

subject_prefix This is a string that will be prepended to the Subject: header of any message posted to the list. For example, if a message is posted to the list with a Subject: like:

```
Subject: This is a message
```

and the `subject_prefix` is `[My List] ` (note the trailing space!), then the message will be received like so:

```
Subject: [My List] This is a message
```

If you leave `subject_prefix` empty, no prefix will be added to the Subject:. Mailman is careful not to add a prefix when the header already has one, as is the case in replies for example. The prefix can also contain characters in the list's preferred language. In this case, because of the vagaries of the email standards, you may or may not want to add a trailing space.

from_is_list This applies to all non-digest messages sent by the list. For settings that apply only to messages whose From: domain publishes a DMARC `p=reject` or `p=quarantine` policy, see the `dmarc_moderation_action` description in section 2.7.

If set to *Munge From*, it replaces the From: header address with the list's posting address to mitigate issues stemming from the original From: domain's DMARC or similar policies and puts the original From: address in a Reply-To: header.

If set to *Wrap Message* it wraps the original message as a MIME subpart of an outer message with From: and Reply-To: headers as above.

anonymous_list This variable allows you to turn on some simple anonymizing features of Mailman. When you set this option to *Yes*, Mailman will remove or replace the From:, Sender:, and Reply-To: fields of any message posted to the list.

Note that this option is simply an aid for anonymization, it doesn't guarantee it. For example, a poster's identity could be evident in their signature, or in other mail headers, or even in the style of the content of the message. There's little Mailman can do about this kind of identity leakage.

Reply-To header munging

This section controls what happens to the Reply-To: headers of messages posted through your list.

Beware! Reply-To: munging is considered a religious issue and the policies you set here can ignite some of the most heated off-topic flame wars on your mailing lists. We'll try to stay as agnostic as possible, but our biases may still peak through.

Reply-To: is a header that is commonly used to redirect replies to messages. Exactly what happens when your users reply to such a message depends on the mail readers your users use, and what functions they provide. Usually, there is both a "reply to sender" button and a "reply to all" button. If people use these buttons correctly, you will probably never need to munge Reply-To:, so the default values should be fine.

Since an informed decision is always best, here are links to two articles that discuss the opposing viewpoints in great detail:

- Reply-To Munging Considered Harmful
- Reply-To Munging Considered Useful

The three options in this section work together to provide enough flexibility to do whatever Reply-To: munging you might (misguidingly :) feel you need to do.

first_strip_reply_to This variable controls whether any Reply-To: header already present in the posted message should get removed before any other munging occurs. Stripping this header will be done regardless of whether or not Mailman will add its own Reply-To: header to the message.

If this option is set to *No*, then any existing Reply-To: header will be retained in the posted message. If Mailman adds its own header, it will contain addresses which are the union of the original header and the Mailman added addresses. The mail standards specify that a message may only have one Reply-To: header, but that that header may contain multiple addresses.

reply_goes_to_list This variable controls whether Mailman will add its own Reply-To: header, and if so, what the value of that header will be (not counting original header stripping – see above).

When you set this variable to *Poster*, no additional Reply-To: header will be added by Mailman. This setting is strongly recommended.

When you set this variable to *This list*, a Reply-To: header pointing back to your list's posting address will be added.

When you set this variable to *Explicit address*, the value of the variable `reply_to_address` (see below) will be added. Note that this is one situation where Reply-To: munging may have a legitimate purpose. Say you have two lists at your site, an announce list and a discussion list. The announce list might allow postings only from a small number of approved users; the general list membership probably can't post to this list. But you want to allow comments on announcements to be posted to the general discussion list by any list member. In this case, you can set the Reply-To: header for the announce list to point to the discussion list's posting address.

reply_to_address This is the address that will be added in the Reply-To: header if `reply_goes_to_list` is set to *Explicit address*.

Umbrella list settings

TBD. Note that umbrella lists are deprecated and will be replaced with a better mechanism for Mailman 3.0.

Notifications

Mailman sends notifications to the list administrators or list members under a number of different circumstances. Most of these notifications can be configured in this section, but see the Bounce Processing and Auto-responder categories for other notifications that Mailman can send.

send_reminders By default Mailman sends all list members a monthly password reminder. This notice serves two purposes. First, it reminds people about all the lists they may be subscribed to on this domain, including the lists where their subscription may be disabled. Second, it reminds people about their passwords for these lists, as well as the url for their personal options pages, so that they can more easily configure their subscription options.

Some people get annoyed with these monthly reminders, and they can disable the reminders via their subscription options page. For some lists, the monthly reminders aren't appropriate for any of the members, so you can disable them list-wide by setting the `send_reminders` variable to *No*.

welcome_msg When new members are subscribed to the list, either by their own action, or the action of a list administrator, a welcome message can be sent to them. The welcome message contains some common boilerplate information, such as the name of the list, instructions for posting to the list, and the member's subscription password. You can add additional information to the welcome message by typing the text into the `welcome_msg` text box. Note that because this text is sent as part of an email, it should **not** contain HTML.

send_welcome_msg This flag controls whether or not the welcome message is sent to new subscribers.

goodbye_msg Like the `welcome_msg`, a "goodbye" message can be sent to members when they unsubscribe from the list. Unlike the welcome message, there's no boilerplate for the goodbye message. Enter the entire goodbye message you'd like unsubscribing members to receive into the `goodbye_msg` text box.

send_goodbye_msg This flag controls whether or not the goodbye message is sent to unsubscribing members.

admin_immed_notify List moderators get notifications of pending administrative actions, such as subscription or unsubscription requests that require moderator approval, or posted messages that are being held for moderator approval. List moderators will always get a daily summary of such pending requests, but they can also get immediate notifications when such a request is made. The `admin_immed_notify` variable controls whether these immediate notifications are sent or not. It's generally a good idea to leave this set to *Yes*.

admin_notify_mchanges This variable controls whether the list administrators should get notifications when members join or leave the list.

respond_to_post_requests This variable controls whether the original sender of a posting gets a notice when their message is held for moderator approval.

Additional settings

This section contains some miscellaneous settings for your mailing list.

emergency When this option is enabled, all list traffic is emergency moderated, i.e. held for moderation. Turn this option on when your list is experiencing a flamewar and you want a cooling off period.

new_member_options Each member has a set of subscription options which they can use to control how they receive messages and otherwise interact with the list. While the members can change these settings by logging into their personal options page, you might want to set the default for a number of the member options. You can do that with this variable, but see also the other categories for other member defaults you can set.

This variable presents a set of checkboxes which control the defaults for some of the member options. *Conceal the member's address* specifies whether or not the address is displayed in the list roster. *Acknowledge the member's posting* controls whether or not Mailman sends an acknowledgement to a member when they post a message to the list. *Do not send a copy of a member's own post* specifies whether a member posting to the list will get a copy of their own posting. *Filter out duplicate messages to list members (if possible)* specifies whether members who are explicitly listed as a recipient of a message (e.g. via the Cc: header) will also get a copy from Mailman.

Of course, members can always override these defaults by making changes on their membership options page.

administrivia This option specifies whether Mailman will search posted messages for *admimistrivia*, in other words, email commands which usually should be posted to the `-request` address for the list. Setting this to *Yes* helps prevent such things as unsubscribe messages getting erroneously posted to the list.

If a message seems to contain administrivia, it is held for moderator approval.

max_message_size This option specifies a maximum message size, in kilobytes, over which the message will be held for moderator approval.

host_name This option specifies the host name part of email addresses used by this list. For example, this is the `example.com` part of the posting address `mylist@example.com`.

It's generally not a good idea to change this value, since its default value is specified when the mailing list is created. Changing this to an incorrect value could make it difficult to contact your mailing list. Also note that the url used to visit the list's pages is not configurable through the web interface. This is because if you messed it up, you'd have to have the site administrator fix it.

include_rfc2369_headers RFC 2369 is an internet standard that describes a bunch of headers that mailing list managers should add to messages to make it easier for people to interact with the list. Mail reading programs which support this standard may provide buttons for easy access to the list's archives, or for subscribing and unsubscribing to the list. It's generally a good idea to enable these headers as it provides for an improved user experience. These headers are often called the `List-*` headers.

However, not all mail readers are standards compliant yet, and if you have a large number of members who are using non-compliant mail readers, they may be annoyed at these headers. You should first try to educate your members as to why these headers exist, and how to hide them in their mail clients. As a last resort you can disable these headers, but this is not recommended.

include_list_post_header The List-Post: header is one of the headers recommended by RFC 2369. However for some announce-only mailing lists, only a very select group of people are allowed to post to the list; the general membership is usually not allowed to post to such lists. For lists of this nature, the List-Post: header is misleading. Select *No* to disable the inclusion of this header. (This does not affect the inclusion of the other `List-*` headers.)

2.2 The Passwords Category

As mentioned above, there are two primary administrative roles for mailing lists. In this category you can specify the password for these roles.

The list owner has total control over the configuration of their mailing list (within some bounds as specified by the site administrator). Note that on this page, for historical reasons, the list owner role is described here as the *list administrator*. You can set the list owner's password by entering it in the password field on the left. You must type it twice for confirmation. Note that if you forget this password, the only way for you to get back into your list's administrative pages is to ask the site administrator to reset it for you (there's no password reminders for list owners).

If you want to delegate list moderation to someone else, you can enter a different moderator password in the field on the right (typed twice for confirmation). Note that if you aren't going to delegate moderation, and the same people are going to both configure the list and moderate postings to the list, don't enter anything into the moderator password fields. If you do enter a separate moderator password, be sure to fill in the `moderator` variable in the *General options* category page.

2.3 The Language Options Category

Mailman is multilingual and internationalized, meaning you can set up your list so that members can interact with it in any of a number of natural languages. Of course, Mailman won't translate list postings. :)

However, if your site administrator has enabled its support, you can set your list up to support any of about two dozen languages, such as German, Italian, Japanese, or Spanish. Your list members can then choose any of your supported languages as their *preferred language* for interacting with the list. Such things as their member options page will be displayed in this language. Each mailing list also has its own *preferred language* which is the language the list supports if no other language context is known.

These variables control the language settings for your mailing list:

preferred_language This is the list's preferred language, which is the language that the list administrative pages will be displayed in. Also any messages sent to the list owners by Mailman will be sent in this language. This option is presented as a drop-down list containing the languages enabled in the `available_languages` variable.

available_languages This set of checkboxes contains all the natural languages that your site administrator has made available to your mailing lists. Select any language that you'd either like your members to be able to view the list in, or that you'd like to be able to use in your list's `preferred_language` variable.

encode_ascii_prefixes If your mailing list's preferred language uses a non-ASCII character set and the `subject_-prefix` contains non-ASCII characters, the prefix will always be encoded according to the relevant standards. However, if your subject prefix contains only ASCII characters, you may want to set this option to *Never* to disable prefix encoding. This can make the subject headers slightly more readable for users with mail readers that don't properly handle non-ASCII encodings.

Note however, that if your mailing list receives both encoded and unencoded subject headers, you might want to choose *As needed*. Using this setting, Mailman will not encode ASCII prefixes when the rest of the header contains only ASCII characters, but if the original header contains non-ASCII characters, it will encode the prefix. This avoids an ambiguity in the standards which could cause some mail readers to display extra, or missing spaces between the prefix and the original header.

2.4 The Membership Management Category

The *Membership Management* category is unlike the other administrative categories. It doesn't contain configuration variables or list settings. Instead, it presents a number of pages that allow you to manage the membership of your list. This includes pages for subscribing and unsubscribing members, and for searching for members, and for changing various member-specific settings.

More details on membership management are described in the Membership Management section.

2.5 The Non-digest Options Category

Mailman delivers messages to users via two modes. List members can elect to receive postings in bundles called *digests* one or a few times a day, or they can receive messages immediately whenever the message is posted to the list. This latter delivery mode is also called *non-digest delivery*. There are two administrative categories available for separately controlling digest and non-digest delivery. You can even disable one or the other forms of delivery (but not both).

Both kinds of delivery can have list-specific headers and footers added to them which can contain other useful information you want your list members to see. For example, you can include instructions for unsubscribing, or a url to the lists digest, or any other information.

Non-digest deliveries can also be *personalized* which means certain parts of the message can contain information tailored to the member receiving the message. For example, the To: header will contain the address of the member when deliveries are personalized. Footers and headers can contain personalized information as well, such as a link to the individual user's options page.

In addition, personalized messages will contain extra information that Mailman can use to unambiguously track bounces from members. Ordinarily, Mailman does some pattern recognition on bounce messages to determine list members whose addresses are no longer valid, but because of the vagaries of mail systems, and the countless forwards people can put in place, it's often the case that bounce messages don't contain any useful information in them. Personalized messages avoid this problem by encoding information in certain headers that unambiguously identify the recipient of a message. If that message bounces, Mailman will know exactly which member it was intended for.

Note that because personalization requires extra system resources, it must be enabled by the site administrator before you can choose it.

Here are the variables which control non-digest delivery:

nondigestable This option controls whether members can receive immediate delivery or not. If not, they will be forced to receive messages in digests. You can't disable non-digest delivery if digests are already disabled.

personalize This option turns on message personalization.

msg_header This text box lets you enter information that will be included in the header of every non-digest message sent through the list.

See below for more information on what can go in the headers and footers. If you leave this text box empty, no header will be added.

msg_footer Just like with the header, you can add a footer to every message. The same rules apply to footers as apply to headers.

Headers and footers can contain any text you want. For non-English lists, the headers and footers can contain any character in the character set of the list's preferred language. The headers and footers can also contain *substitution variables* which Mailman will fill in with information taken from the mailing list. These substitutions are in Python string interpolation format, where something like `%(list_name)s` is substituted with he name of the mailing list. Note that the trailing '`s`' is required[2].

For example, a footer containing the following text:

```
This is the \%(list_name)s mailing list
Description: \%(description)s
```

might get attached to postings like so:

```
This is the Example mailing list
Description: An example of Mailman mailing lists
```

Here is the list of substitution variables available for your headers and footers:

real_name This is the value of the `real_name` configuration variable in the General options category.

list_name This is the canonical name of the mailing list. In other words it's the posting address of the list[3].

host_name This is the domain name part of the email address for this list.

[2]The site administrator can configure lists to use a simpler interpolation format, where `$list_name` or `${list_name}` would be substituted with the mailing list's name. Ask your site administrator if they've configured your list this way.

[3]For backward compatibility, the variable `_internal_name` is equivalent.

web_page_url This is the base url for contacting the list via the web. It can be appended with `listinfo/%(list_-name)s` to yield the general list information page for the mailing list.

description The brief description of the mailing list.

info This is the full description of the mailing list.

cgiext This is the extension added to CGI scripts. It might be the empty string, `.cgi`, or something else depending on how your site is configured.

Note that `real_name`, `host_name`, `description`, and `info` substitution variables take their values from the list configuration variables of the same name.

When personalization is enabled, the following substitution variables are also available:

user_address The address of the recipient of the message, coerced to lower case.

user_delivered_to The case-preserved address that the user subscribed to the mailing list with[4].

user_password The user's password, in clear text.

user_name The user's full name.

user_optionsurl The url to the user's personal options page.

2.6 The Digest Options Category

Digest delivery is a way to bundle many articles together into one package, which can be delivered once per day (if there were any posted articles), or whenever the package is bigger than a specified limit. Some users may prefer this style of delivery for higher traffic lists since they will receive fewer messages.

Mailman supports two standard digest formats, and if digests are enabled, users can select which of the two formats they receive. One is MIME digests, where each message is an attachment inside a multipart/digest. This format also contains a summary table of contents, and of course the an optional header and footer, and it retains most of the headers of the original messages.

The second type is called "plaintext" digests because they are readable in mail readers that don't support MIME. Actually, they adhere to the RFC 1153 digest standard. They retain some, but not all of the original messages, but can also include a summary and headers and footers.

Like non-digest delivery, you can enable or disable digest delivery, but you cannot disable both types of delivery. You can specify different headers and footers for digest and non-digest deliveries. You cannot personalize digest deliveries.

As list administrator, you may want to send an urgent message to all list members, bypassing the normal digest bundling. To do this, send the message with a Urgent: header, where the value of the header is the list administrator's password. Non-digest members will receive the message like normal, but digest members will receive the message immediately[5].

Here are the variables which control digest delivery:

digestable The option controls whether members can receive digest deliveries or not. If not, they will be forced to receive immediate deliveries. You can't disable digests if non-digests are already disabled.

digest_is_default Controls which style of delivery is the default for new members. You can choose *Regular* (non-digest) or *Digest* delivery.

[4] Usually it makes no difference which of `user_address` and `user_delivered_to` is used, but it's important to remember that they can be different. When they're different, Mailman always uses the lower case address as the key to the member's subscription information, but it always delivers messages to the case-preserved version.

[5] They'll also receive the message in the digest.

mime_is_default_digest If a member is allowed to choose digests, this variable controls which is the default digest style they will receive. *Plain* digests are RFC 1153 format as described above.

digest_size_threshold Normally, digest members get at least one message per day, if there have been any messages posted to the list. However, for high volume lists, you may want to send out digests when the size has reached a certain threshold, otherwise, the one digest they receive could be huge. This variable controls the size threshold by specifying the maximum digest size in kilobytes. Note that this threshold isn't exact. Set this variable to zero to specify that there is no size threshold, in which case no more than one digest will be sent out per day, but ensure that digest_send_periodic is *Yes* in this case or no digests will be sent.

digest_send_periodic This variable actually controls whether or not a digest is sent daily when the size threshold has not yet been met. If set to *No*, then digests will only be sent when they are bigger than `digest_size_threshold`.

digest_header This text box lets you enter information that will be included in the header of every digest message sent through the list. The same information can go in this header as can go in the `msg_header`, except for the personalization variables.

digest_footer Just like with the header, you can add a footer to every message. The same rules apply to digest footers as apply to digest headers.

digest_volume_frequency Each digest is numbered with a volume and an issue. This variable controls how often a new digest volume is sent. When the digest volume number is incremented, the issue number is reset to 1.

_new_volume This is an action variable, which forces an increment of the volume number as soon as you submit the form.

_send_digest_now This is another action variable. Select *Yes*, submit the form, and the current digest is packaged up and sent to digest members, regardless of size (well, there has to be at least one message in the digest).

2.7 The Privacy Options Category

The Privacy category lets you control how much of the list's information is public, as well as who can send messages to your list. It also contains some spam detection filters. Note that this section is not used to control whether your list's archives are public or private; for that, use the **??** category.

There are four sub-categories:

- Subscription rules – i.e. the rules for joining and leaving your mailing list
- Sender filters – the rules for who may post messages to your list
- Recipient filters – moderation rules based on the recipient of the message
- Spam filters – some regular expression based rules for header matching

The sender, recipient, and spam filtering rules are part of the general list moderation features of Mailman. When a message is posted to the list, it is matched against a number of criteria, the outcome of which determines whether the message is reflected to the membership or not. In general, the outcome is one of four states:

- Approved or Accepted – the message may be sent on to the members of the mailing list.
- Hold – the message will be held for moderator approval. The list owners and moderators will then have to explicitly approve the message before the list members will see it.
- Reject – the message is bounced back to the original sender, often with a notice containing the reason the message was rejected. The list members never see rejected messages.

- Discard – the message is simply thrown away without further processing.

Many of the fields in this section are text boxes accepting addresses, one per line. Unless otherwise noted, these also accept regular expressions which will be matched against an address, if the line begins with a ˆ (caret) character.

Subscription rules

This subcategory controls the rules for exposing the existence of this list, and for what new members must do in order to subscribe to the list.

advertised This option controls whether this list will show up in the list overview for the site. Normally, an overview contains the name and short description of every mailing list in the virtual domain. By setting this variable to *No*, it will not show up in this overview, nor will it show up in the administrative overview. The only way then to find the list is to guess (or know!) its name.

subscribe_policy This option controls the steps that a new member must take to join the list. The available options may differ based on some defaults that the site administrator chooses. They are:

- None – No verification is done on the subscribing member. This is also called *open subscriptions* and is generally disabled by default. The site administrator must allow list admins to choose this option; if not, this option will not be presented to you.
- Confirm – An email confirmation step is required before the address is added to the list. When a member requests subscription, either via the web page or by sending a message to *yourlist*`-join@example.com`, Mailman will send a confirmation message to the requesting address. This mail-back confirmation contains a unique identifier, which the requester can present to Mailman in order to confirm their subscription. This can be done either by replying to the mail-back, or by visiting the url in the mail-back message. The url points to a page that lets the user either discard or confirm their request.
- Require approval – All subscription requests are held for approval of the list moderator. No mail-back confirmation is sent, but the list admins will recieve a message indicating that approval is pending.
- Confirm and approve – Here, a mail-back notice must first be confirmed by the requester. Once confirmed, the list moderator must then approve the request. This is the most secure method for users to subscribe since it both verifies the requesting address, and forces the list moderators to approve the request.

unsubscribe_policy Specifies whether the list moderator's approval is required for unsubscription requests. *No* is highly recommended, since it is exceedingly impolite to not allow people to leave a mailing list whenever they want (i.e. opt-out). *Yes* is useful in some specialized contexts; e.g. you may not want to allow employees to unsubscribe from the company newsletter.

ban_list This contains a list of addresses (or regular expressions), one per line, that are banned from ever subscribing to your mailing list. If a match occurs during the subscription process, the request will be automatically rejected, and the requester will get a rejection notice. You can use this to permanently ban troublesome posters to a members-only list.

private_roster This specifies who is allowed to view the roster of member addresses. If you choose *Anyone*, then the list membership is completely public. You can limit exposure of the roster to just list members, or just to the list administrators. In the former case, a user must enter a valid member's address and password before they can view the roster. In the latter case, a list administrator's password must be entered; if a matching admin password is entered, address field is ignored.

obscure_addresses Controls whether some simple obfuscation of addresses is used when member addresses are included on web pages. This should reduce the opportunity for email address harvesting by spammers, although it probably doesn't eliminate it.

Sender filters

When a message is posted to the list, a series of moderation criteria is applied to determine the disposition of the message. This section contains the moderation controls for postings from both members and non-members.

default_member_moderation Member postings are held for moderation if their *moderation flag* is turned on. Note that only the list administrators can change the value of a member's moderation flag.

You can control whether new members get their moderation flag turned on or off by default when they subscribe to the list. By turning this flag off by default, postings by members will be allowed without further intervention (barring other restrictions such as size or implicit recipient lists – see below). By turning the flag on, you can quarantine new member postings to make sure that they meet your criteria for netiquette, topicality, etc. Once you determine that the new member understands the community's posting rules, you can turn off their moderation flag and let their postings go through unstopped.

E-newsletter style lists can also be set up by using the moderation flag. By setting the `member_moderation_action` to *Reject*, and by turning off the moderation flag for just the few approved senders, your list will operate in essentially a one-way direction. Note that you'd also need to reject or discard postings from non-members.

member_moderation_action This is the action to take for postings from a member who's moderation flag is set. For typical discussion lists, you'll likely set this to *Hold* so that the list moderator will get a chance to manually approve, reject, or discard the message. For e-newsletter and announcement lists, you might want to set this to *Reject* or *Discard*.

Note that when a moderated member posts to your list, and the `member_moderation_action` is set to *Hold*, the message will appear on the administrative requests page. When you dispose of the message, you will be given an opportunity to clear the moderation flag at the same time. If you're quarantining new posts, this makes it very convenient to both approve a new member's post and de-moderate them at the same time.

member_moderation_notice When a member's moderation flag is turned on and `member_moderation_action` is *Reject*, this variable contains the text sent in the rejection notice.

The next group of settings control messages whose From: domain publishes a DMARC `p=reject` or `p=quarantine` policy.

dmarc_moderation_action These actions, *Accept*, *Munge From*, *Wrap Message*, *Reject* and *Discard* apply only to messages whose From: domain publishes a DMARC `p=reject` or optionally (see the next setting) `p=quarantine` policy. If the message is From: such a domain and this action is other than *Accept*, this action applies to the message. Otherwise the `from_is_list` setting in section 2.1 applies. See the `from_is_list` setting in section 2.1 for a description of the *Munge From* and *Wrap Message* actions.

dmarc_quarantine_moderation_action If this is set to *Yes*, the above `dmarc_moderation_action` applies to messages with From: domain DMARC policy `p=quarantine` as well as `p=reject`.

dmarc_moderation_notice When `dmarc_moderation_action` applies and is *Reject*, this variable contains the text sent in the rejection notice. If empty, a generic notice mentioning DMARC is sent.

The next batch of variables controls what happens when non-members post messages to the list. Each of these accepts one email address per line; regular expressions are allowed if the line starts with the ^ (caret) character. These address lists are always consulted in the order in which they're presented on this page (i.e. accepts first, followed by holds, rejections, and discards).

accept_these_nonmembers Postings from non-members whose addresses match this list are accepted, barring other list restrictions due to size, implicit recipients, etc. You might want to add alternative addresses of approved posters to this list.

hold_these_nonmembers Postings from non-members whose addresses match this list are held for moderator approval.

reject_these_nonmembers Postings from non-members whose addresses match this list are rejected, i.e. bounced back to the original sender. There is currently no way to add additional text to the rejection message.

discard_these_nonmembers Postings from non-members whose addresses match this list are discarded, with no bounce back message. You might want to add the addresses of known spammers to this list.

generic_nonmember_action This variable controls what happens to non-member posts when the address of the sender doesn't match any of the above four lists. If you set this to *Hold*, the posting will appear on the administrative requests page, and you will be given an opportunity to add the non-member to one of the above four lists at the same time you dispose of the held message.

forward_auto_discards When messages from non-members are discarded, either because the sender address matched `discard_these_nonmembers`, or because `generic_nonmember_action` is *Discard*, you can choose whether such messages are forwarded to the list administrators or not.

Recipient Filters

The variables in this section control various filters based on the recipient of the message.

require_explicit_destination This controls whether the mailing list posting address must be explicitly named in the To: or Cc: recipient lists. The main reason why it wouldn't is if the message was blind-carbon-copied (i.e. Bcc:'d) to the list. Spammers like to do this, but sometimes legitimate messages are forwarded to the list this way.

If the list is not explicitly addressed and this setting is turned on, the message will be held for moderator approval.

acceptable_aliases This is the list of alternative addresses that are acceptable as a list posting address when `require_explicit_destination` is enabled. This is useful for when there aliases for the main posting address (e.g. `help@example.com` may be an alias for `help-list@example.com`).

max_num_recipients This is the maximum number of explicit recipients that are allowed on the posted message. Spammers sometimes send messages with lots of explicit recipients, so setting this number to a reasonable value may cut down on spam.

Spam Filters

This section provides some adjuncts to spam fighting tools; it doesn't replace dedicated anti-spam tools such as SpamAssassin or SpamBayes.

bounce_matching_headers This variable contains header regular expressions, one per line, and if any of a message's headers matches one of these patterns, it will be held for moderation. The format is a colon separated header and value, where the header is case insensitive and the value is any valid Python regular expression. Lines that start with # are ignored.

This variable can be used to catch known spammers by writing regexps that match against To: or Cc: lines, or known-bad Message-ID:s. Perhaps more useful though are patterns that match headers added by spam detection tools higher up in the tool chain. For example, you might configure SpamAssassin to add an X-Spam-Score: header with between zero and 5 stars depending on the spam score. Then you can add a line to this variable like:

```
X-Spam-Score: [*]{3,5}
```

This line will match from 3 to 5 stars in the value of this field.

2.8 The Bounce Processing Category

These policies control the automatic bounce processing system in Mailman. Here's an overview of how it works:

When a bounce is received, Mailman tries to extract two pieces of information from the message: the address of the member the message was intended for, and the severity of the problem causing the bounce. The severity can be either *hard* for fatal errors, or *soft* for transient errors. When in doubt, a hard severity is used.

If no member address can be extracted from the bounce, then the bounce message is usually discarded. Every member has a *bounce score*, initialized at zero, and every time we encounter a bounce from a member we increment that member's score. Hard bounces increment by 1 while soft bounces increment by 0.5. We only increment the bounce score once per day, so even if we receive ten hard bounces from a member per day, their score will increase by only 1 for that day.

When a member's bounce score is greater than the *bounce score threshold* (see below), the member's subscription is disabled. Once disabled, the member will not receive any postings from the list until their membership is explicitly re-enabled, either by the list administrator or the user. However, they will receive occasional reminders that their membership has been disabled, and these reminders will include information about how to re-enable their membership. You can control both the number of reminders the member will receive and the frequency with which these reminders are sent.

There is one other important configuration variable; after a certain period of time – during which no bounces from the member are received – the bounce information is considered stale and discarded. Thus by adjusting this value, and the score threshold, you can control how quickly bouncing members are disabled. You should tune both of these to the frequency and traffic volume of your list.

bounce_processing Specifies whether or not this list should do automatic bounce processing.

bounce_score_threshold This is the bounce score above which a member's subscription will be automatically disabled. When the subscription is re-enabled, their bounce score will be reset to zero. This value can be a floating point number.

bounce_info_stale_after The number of days after which a member's bounce information is considered stale. If no new bounces have been received in the interim, the bounce score is reset to zero. This value must be an integer.

bounce_you_are_disabled_warnings The number of notices a disabled member will receive before their address is removed from the mailing list's roster. Set this to 0 to immediately remove an address from the list once their bounce score exceeds the threshold. This value must be an integer.

bounce_you_are_disabled_warnings_interval The number of days between each disabled notification.

bounce_unrecognized_goes_to_list_owner This variable controls whether unrecognized bounces are discarded, or forwarded on the list administrator. The bounce detector isn't perfect, although personalization can make it much more accurate. The list owner may want to receive unrecognized bounces so that they can manually disable or remove such members.

bounce_notify_owner_on_disable This option controls whether or not the list owner is notified when a member's subscription is automatically disabled due to their bounce threshold being reached.

bounce_notify_owner_on_removal This option controls whether or not the list owner is notified when a member is removed from the list after their disabled notifications have been exhausted.

2.9 The Archiving Options Category

Mailman comes with a built-in web-based archiver called *Pipermail*, although it can be configured to use external, third party archivers.

archive This option tells Mailman whether to archive messages it receives or not, regardless of whether Pipermail or a third party archiver is used. Turn this off if you don't want to archive messages.

Note that senders can control whether their own posts are archived, on an individual per-message basis. If the posted message has a X-No-Archive: header (regardless of value), or a X-Archive: header with a value of `No` (case insensitive), then the message will not be archived, although it will be treated as normal in all other ways.

archive_private Controls whether Pipermail archives are private or public. Private archives require a valid member address and password, or a list administrator password in order to access them. This option has no effect when a third party archiver is used.

archive_volume_frequency Controls how Pipermail splits messages in the archive. The most common option is *Monthly* meaning a new archive volume is started every month. Very high volume lists may want a shorter frequency (e.g. *Weekly* or *Daily*) where as lower volume lists may want a longer frequency (e.g. *Yearly*). This option has no effect when a third party archiver is used.

2.10 The Mail/News Gateway Category

Mailman has a sophisticated mail-to-news gateway feature. It can independently gate messages from news to mail and vice versa, and can even be used to manage moderated newsgroups.

2.11 The Auto-responder Category

2.12 The Content Filtering Category

2.13 The Topics Category

3 Membership Management

4 Tending to Pending Moderator Requests

5 Editing the Public HTML Pages

6 Deleting the Mailing List

A This is an Appendix

To create an appendix in a Python HOWTO document, use markup like this:

```
\appendix

\section{This is an Appendix}

To create an appendix in a Python HOWTO document, ....


\section{This is another}

Just add another \section{}, but don't say \appendix again.
```