

LMTP process

LMTP in Mailman 3

[RFC 2033, Local Mail Transport Protocol \(LMTP\)](#) provides Mailman with a unique opportunity to provide a better user experience when accepting initial postings. Two improvements in the process are available with LMTP.

First, we can eliminate most of the integration cruft we currently have with supporting multiple MTAs for incoming mail. Most of the major SMTP servers support LMTP delivery, so by providing an LMTP server in Mailman, the hope is that we can avoid all the crufty MTA-specific alias hackery.

Second, and perhaps more importantly, we can support better anti-backscatter and anti-spam defenses for messages sent to the mailing lists, by rejecting messages in the SMTP dialog instead of having to make the determination way later and sending a bounce message. Ian Eiloart describes [what is possible](#).

How to set up LMTP delivery for Mailman 3

Here is detailed information about how Mailman should integrate with the various MTAs using LMTP.

Postfix

✓ Ubuntu servers

If you're getting `Connection refused` errors, you might be hitting [bug 340383](#). Try commenting out the `127.0.1.1` setting in your `/etc/hosts` file. Because Postfix runs under a chroot on Ubuntu, you'll also need to comment this out in `/var/spool/postfix/etc/hosts`.

Postfix uses the Postfix `lmtpl` client to transport messages to a LMTP server. You probably already have this in your Postfix `master.cf` file:

```

#
=====
=====
# service type  private unpriv  chroot
wakeup  maxproc  command + args
#                (yes)    (yes)    (yes)
(never) (100)
#
=====
=====
...
lmtpl      unix  -      -      -
-          -      lmtpl
...

```

We're going to use a transport map to tell Postfix to deliver all messages destined for a Mailman mailing list to its lmtpl service. Fortunately, Mailman is already able to write the correct transport map based on your configuration settings. All you need to do is add a `transport_maps` setting in your Postfix `main.cf` file:

```

transport_maps =
    ...

hash:/path/to/mailman/var/data/postfix_lmtpl
    ...

```

Now Postfix knows where it should transport messages to, but it doesn't know yet it should accept messages for the given recipients - it doesn't know the recipients are valid recipients. You can reuse the Mailman generated transport map for this by adding the following to your `main.cf` file:

```
local_recipient_maps =  
    ...  
  
hash:/path/to/mailman/var/data/postfix_lmtp  
    ...
```

Virtual domain

If the mailman list addresses are part of `$virtual_alias_domains` or `$virtual_mailbox_domains` add `postfix_lmtp` to the listing of `$virtual_alias_maps`:

```
virtual_alias_maps =  
    ...  
  
hash:/path/to/mailman/var/data/postfix_lmtp  
    ...
```

Relay domain

If the mailman list addresses are part of the relay domain namespace add `postfix_lmtp` to the listing of `$relay_recipient_maps`:

```
relay_recipient_maps =  
    ...  
  
hash:/path/to/mailman/var/data/postfix_lmtp  
    ...
```

Once Postfix has been reloaded the new settings will take effect.

A dedicated list server

For a dedicated list server e.g. `list.example.com` simply add the hostname as key to a transport map and specify the mailman LMTP server as destination:

# key	value
<code>list.example.org</code>	
<code>lmtp:inet:localhost</code>	

Be aware though that Postfix has no knowledge of valid recipients in the destination's (sub)domain (here: `list.example.org`) if you specify the list server as noted above. Postfix will accept and transport any message destined for `list.example.org` to the mailman LMTP server and it will be the LMTP servers task to reject messages for invalid or non-existing recipients.

There are two ways to prevent putting such load on the mailman LMTP server. First specify each valid recipient address in a transport table as shown in the initial example. Alternatively use the `reject_unverified_recipient` option and let the Postfix `verify(8)` daemon find out if the recipient address exists. This way Postfix will reject messages to non-existing recipients during the SMTP session with the client that attempts to deliver the message. See `ADDRESS_VERIFICATION_README` for details on implementing the `reject_unverified_recipient` option.

A dedicated lmtp client

⊖ As of Mailman 3.0a4, the service name used in the Mailman generated `postfix_lmtp` file is not configurable. See [bug 490030](#).

There may be situations where a dedicated lmtp client, that differs in its configuration from the default lmtp client settings, is required.

To create such a client add a new service (here: `mailman3`) to the Postfix `master.cf` configuration file and add the configuration options which should override the default lmtp client behaviour:

```

#
=====
=====
# service type private unpriv chroot
wakeup maxproc command + args
# (yes) (yes) (yes)
(never) (100)
#
=====
=====
...
mailman3 unix - - -
- - lmtpl
-o lmtpl_send_xforward_command=yes
-o disable_dns_lookups=yes
...

```

Then specify the new service in the transport table e.g. `/etc/postfix/mailman_lists:`

```

# key value
mailman@example.org
mailman3:inet:localhost
mailman-admin@example.org
mailman3:inet:localhost
mailman-bounces@example.org
mailman3:inet:localhost
mailman-confirm@example.org
mailman3:inet:localhost
...

```

Exim

There are two variants of LMTP support in Exim transports. The first of these exists as an option to the SMTP driver, instructing it to talk LMTP over the SMTP connection. Such a transport will look like this:

```
mailman_remote_lmtp:  
    driver = smtp  
    protocol = lmtp  
    hosts = localhost  
    allow_localhost
```

There is an independent LMTP transport driver, which is able to communicate using unix sockets. For reference, this transport will look like the following, excluding any additional local configuration options:

```
mailman_local_lmtp:  
    driver = lmtp  
    socket = "/var/run/path/to/unix/socket"  
    batch_max = 40  
    user = mailman
```

Email routing using Exim

The exim router should only need its transport option changed, the rest of the logic can remain the same. One can consult the output of `genaliases`:

```

mailman_aliases:
    driver = accept
    domains = +local_domains
    condition =
    ${lookup${local_part}lsearch{/path/to/genaliases/output}}
    transport = mailman_local_lmtp

```

Alternatively, one can check directly for a list's existence:

```

mailman_direct:
    driver = accept
    domains = +local_domains
    require_files =
    /mail/mailman/lists/${lc::$local_part}/config.pck
    local_part_suffix_optional
    local_part_suffix = -admin      : \
                        -bounces    :
-bounces+*          : \
                        -confirm    :
-confirm+*          : \
                        -join       : -leave
: \
                        -owner      : -request
: \
                        -subscribe :
-unsubscribe
    transport = mailman_local_lmtp

```

Access Controls in Exim

Once you've set up routing and transports, you can use Exim's call forwards in an ACL (with the `use_sender` option) to determine whether the sender is permitted to post to the list. You should get a definitive answer, and this mechanism allows you to use a remote Mailman installation as if it were local - that is, you don't need to consult any local files. See section 40.40 of the Exim docs.

Sendmail