

## Configuring Postfix to Deliver Passive Probe Mail to LISTSERV

For most list-related mail, LISTSERV uses a variety of standard mail aliases. For example, the `/etc/aliases` entries for a list called 'EXAMPLE' might look as follows:

```
# -- Aliases for 'example' mailing list example:
|/usr/local/bin/lsv_amin /home/listserv/spool example"
owner-example: "|usr/local/bin/lsv_amin /home/listserv/spool owner-example"
example-request: "|usr/local/bin/lsv_amin /home/listserv/spool example-request"
example-search-request: "|usr/local/bin/lsv_amin /home/listserv/spool example-
search- request"

example-server: "|usr/local/bin/lsv_amin /home/listserv/spool example-
server" example-subscribe-request: "|usr/local/bin/lsv_amin
/home/listserv/spool example- subscribe-request"
example-signoff-request: "|usr/local/bin/lsv_amin /home/listserv/spool
example- signoff-request"
example-unsubscribe-request: "|usr/local/bin/lsv_amin /home/listserv/spool
example- unsubscribe-request"
```

Bounces for regular list postings to the EXAMPLE list on the LISTSERV.EXAMPLE.ORG server would generally have a Return-Path of

`<owner-example@LISTSERV.EXAMPLE.ORG>`, get processed per the `owner-example` entry in `/etc/aliases`, and then be passed LISTSERV. However, for list postings sent as passive probes or for other kinds of mail-merged messages (such as those sent by LISTSERV Maestro), the Return-Path is customized on a per-user basis. So instead of bounces coming back to `<owner-example@LISTSERV.EXAMPLE.ORG>`, they

might come back to `<owner-example*foo**bar*-com@LISTSERV.EXAMPLE.ORG>` (where the e-mail address of the subscriber is [foo@bar.com](mailto:foo@bar.com)). Since there is no `/etc/aliases` entry for `owner-example*foo**bar*-com`, sendmail and Postfix don't know how to route the message and reject it with a "No such local user" error.

The same problem exists for DMARC rewrites, which are in the form of

```
*-dmarc-request@listserv.example.org.
```

Fortunately, it is possible to configure Postfix to route such mail, by way of a regular expression pattern match. This document explains how to configure that regexp match. Unless otherwise specified, each of these steps should be performed as the 'root' user.



## Configuring Postfix to Listen on All Network Interfaces

Many Linux/UNIX distributions (including Mac OS X) ship a default Postfix configuration that accepts mail on the local host interface only. In order to get mail to LISTSERV, you'll need to tell Postfix to listen to all network interfaces on the server.

Find the following line in

```
/etc/postfix/main.cf:
```

```
inet_interfaces = localhost
```

and change it to:

```
inet_interfaces = all
```

## Verifying that Postfix Supports Regexp Pattern Matches

In order to do regular expression pattern-matching, regexp support needs to be built into Postfix. Verify this with the following command:

```
#  
postconf -m  
btree  
cidr  
environ  
hash  
ldap  
nis  
pcre  
proxy  
regexp  
static  
unix
```

You should see 'regexp' in the list of supported formats. If not, you'll need to rebuild your Postfix binary to support regexp. (Rebuilding Postfix is beyond the scope of this document.)

## Adding a Transport for lsv\_amin

Next, we need to add a Postfix transport to route mail to lsv\_amin. Open `/etc/postfix/master.cf`, and add the following:

```
lsvamin unix - n n - - pipe flags=F user=listserv  
argv=/usr/local/bin/lsv_amin /home/listserv/spool ${user}
```



(Note that the above should be on two lines only; the first line begins with `?lsvamin'` and the second line begins with `?argv'` and ends with `${user}.`)

This example assumes that you've installed LISTSERV's `lsv_amin` binary to `/usr/local/bin/lsv_amin`, and LISTSERV's spool directory is `/home/listserv/spool`. If not, change the paths as appropriate for your installation.

## Creating a Regular Expression Transport Map

Now create a new file called `/etc/postfix/transport_regexp`. In that file, add 2 lines as follows:

```
/^owner(.*)\@/ lsvamin:
/(.*)-request@/ lsvamin:
```

This regexp tells Postfix to send any mail for an address that begins with `<owner*>` or `<*-request>` to the 'lsvamin' transport that you defined in the previous step. It will match the `<owner-example*foo**bar*-com@LISTSERV.EXAMPLE.ORG>` address in our original example, as well as any other Return-Path that begins with `<owner*>`, including all LISTSERV probe-style messages and mailings sent from LISTSERV Maestro. The second line will match any address that begins with `somettext` ending in `<*-request>` before the `@` sign. Save the `transport_regexp` file, and create a transport map database from it:

```
# postmap /etc/postfix/transport_regexp

# ls -l /etc/postfix/transport_regexp*

-rw-r--r- 1 root root 44 Jun 15 12:57 /etc/postfix/transport_regexp
-rw-r--r- 1 root root 12288 Jul 6 12:46 /etc/postfix/transport_regexp.db
```

## Configure Postfix to Use the New Regexp Transport

Next, we need to tell Postfix to use the new regexp. Open `/etc/postfix/main.cf`, and find the line for `local_recipient_maps`. Edit that line to include `'$transport_maps'`. For example:

```
local_recipient_maps = unix:passwd.byname $alias_maps
$transport_maps
```

Then either add or edit your `transport_maps` line to include the new `transport_regexp` that you've created:

```
transport_maps = regexp:/etc/postfix/transport_regexp
```

Save your edited `main.cf`, and you're finished! Restart postfix to activate the changes:



```
# postfix stop
postfix/postfix-script: stopping the Postfix mail system
# postfix start
postfix/postfix-script: starting the Postfix mail system
```

Postfix should now accept mail for <owner-example\*foo\*\*bar\*-com@LISTSERV.EXAMPLE.ORG>, match it to your `transport_regexp` file, and send it to the 'lsvamin' transport for delivery by `lsv_amin` to LISTSERV.