



Stockmann Automatisering

Reigerstraat 18
4901 AH Oosterhout
tel. +31 162 471337
fax. +31 162 471338
email stockmann@stokkie.net

kvk Breda nr 20112432
SNS bank Oosterhout 9013.15.168
BTW nr.: NL1232.39.989.B01

20 maart 2004

QMAIL Aliases , excerpts from “The qmail Handbook” by Dave Sill, © 2002 Apress[tm], ISBN 1-893115-40-2

Friday 20 maart 2004

Copyright © 2004 Stockmann Automatisering All rights reserved. This proposal is for use of client personnel only. No part of it may be circulated, quoted, or reproduced for distribution outside of the client organisation without prior consent.



Os support: RedHat, SuSE, Debian, Mandrake, SCO, Solaris HW support: Intel, AMD, Sun, IBM Network: Cisco, 3COM, Nortel

Aliases

qmail's standard aliasing mechanism is a natural outgrowth of qmail's local delivery mechanism. `qmail-local` attempts to deliver a message addressed to `localpart@host` to a local user named `localpart`. If no matching user is found, the message is delivered to the `alias` user, a pseudo-user on all qmail systems whose home directory is usually `/var/qmail/alias`.

For example, say you want to create a `info@example.com` alias that forwards messages to user `zack`. On `example.com` as user `root`, do this:

```
# echo zack > /var/qmail/alias/.qmail-info
# chmod 644 /var/qmail/alias/.qmail-info
#
```

Chapter 4, "Using qmail", describes how to create the `.qmail` files that specify which aliases exist and what to do with messages sent to them.

Note that because of the way aliases are implemented in qmail, an alias can *never* override a valid user's deliveries. For example, if `rachel` is a normal user, then `~alias/.qmail-rachel` will not be used. An exception to this rule is the `qmail-users` mechanism discussed in the next section.

The `fastforward` package provides an alternative aliasing mechanism that puts multiple aliases in a single file compatible with Sendmail's alias database.

The next section describes another mechanism you can use to implement aliases.

The qmail-users Mechanism

`qmail-users` is a system for assigning addresses to users. Normally, local deliveries are handled by `qmail-spawn`, which runs `qmail-getpw` to determine which user the address belongs to, where their home directory resides, their UID, GID, and the breakdown of the addresses if it's an extension address.

However, if the file `/var/qmail/users/cdb` exists, `qmail-spawn` attempts to look up the delivery details first. The `users/cdb` file is a binary database generated from `users/assign` using the `qmail-newu` command. The `assign` file is a table of assignments. There are two kinds of assignments: simple and wildcard.

NOTE `assign` contains a series of assignments, one per line, followed by a line containing a single dot (.). If you create `assign` manually, don't forget to add the "dot" line.

Simple Assignment

A simple assignment look like

```
=address:user:uid:gid:directory:dash:extension:
```

What this means is that messages received for *address* will run as *user* user, with the specified *uid* and *gid*, and the file *directory/.qmaildashextension* will specify how the messages are to be delivered.

For example, say you want mail for the local recipient *info* to be handled user *andy* (UID=35, GID=20, directory=/home/andy) using the file /home/andy/.qmail-info. The following simple assignment will accomplish that:

```
=info:andy:35:20:/home/andy:-:info:
```

NOTE If multiple simple assignments specify the same address, *qmail-lspawn* will use the first one.

Wildcard Assignment

A wildcard assignment look like

```
+prefix:user:uid:gid:directory:dash:prepend:
```

What this means is that messages received for addresses of the form *prefixrest* will run as user *user*, with the specified *uid* and *gid*, and the *directory/.qmaildashprependrest* will specify how the messages are to be delivered.

For example, given the following wildcard assignment:

```
+andy-:andy:35:20:/home/andy:-:::
```

A message for *andy-info* will be delivered as user *andy* using the directions in /home/andy/.qmail-info.

NOTE More specific wildcard assignments take precedence over less specific wildcard assignments, and simple assignments take precedence over wildcard assignments.

qmail-users Programs

qmail-users has two helper programs: *qmail-newu* and *qmail-pw2u*.

qmail-newu processes the *assign* file and generates a constant database (CDB) file called *cdb* in /var/qmail/users. CDB is a binary (machine readable, not human readable) format that can be accessed quickly by *qmail-lspawn*, even when there are thousands of assignments.

qmail-pw2u converts the system user database, /etc/passwd, into a series of assignments suitable for *assign*. *qmail-pw2u* uses a set of files to modify the translation rules:



- . include—specific users to include in assign
- . exclude—specific users to exclude from assign
- . mailnames—alternative “mailnames” for users
- . subusers—extra addresses handled by a user, with an optional .qmail extension
- . append—miscellaneous assignments

CAUTION *If you use `qmail-pw2u`, don't forget to re-run `qmail-pw2u` and `qmail-newu` whenever you add users, remove users, change UIDs, or change GIDs.*

Conclusion

In this chapter you've learned about qmail's control files: what they do and how to change them. You've learned about relaying: the dangers of being an open relay and how to enable selective relaying based on host identification or user or host authentication. You've also learned how to configure qmail to support multiple domain names, both native and virtually hosted. And finally, you've learned how to use qmail's basic alias mechanism and `qmail-users` to map addresses into local mailboxes.

In Chapter 4, “Using qmail”, you'll learn how users can manage their mailboxes, control the appearance of the messages they send, and use the utilities provided by qmail for processing incoming and managing mailboxes.



Os support: RedHat, SuSE, Debian, Mandrake, SCO, Solaris HW support: Intel, AMD, Sun, IBM Network: Cisco, 3COM, Nortel

Configuring qmail: Advanced Options

Chapter 3, “CONFIGURING QMAIL: THE BASICS”, covered the fundamentals of qmail configuration: the control files, relay control, aliases, virtual domains, and the qmail-users mechanism.

This chapter will explore more advanced configuration topics.

Aliases

Sendmail uses aliases stored in a database, typically `/etc/aliases`, which are usually converted to a machine-readable database for faster lookups, usually `aliases.db` or `aliases.dir` or `aliases.pag`.

qmail implements aliases via `.qmail` files in the home directory of the alias user or via the optional qmail-users mechanism if `/var/qmail/users/cdb` exists.

When migrating from Sendmail to qmail, you have two choices: convert the Sendmail aliases to qmail aliases or install the fastforward package, which implements Sendmail-style alias databases under qmail.

Converting Sendmail-Style Aliases to qmail-Style Aliases

Entries in `/etc/aliases` are in the format

```
alias: expansion
```

where *expansion* is a comma-separated list of the following:

- . Forwarding address, for example: `maryjane@isp.example.net`
- . File delivery, for example: `/usr/local/majordomo/incoming-log`
- . Program delivery, for example: `|/usr/local/bin/info-responder`
- . File includes, for example: `:include: /usr/local/mail/lists/users`

Some of these might be enclosed in double quotes, especially program deliveries that contain spaces.

The general strategy for converting these to qmail aliases is

1. Create `/var/qmail/alias/.qmail-alias`
2. Put each forwarding address, program delivery, and file delivery, and file delivery in the alias in the `.qmail-alias` file, converting to qmail format as necessary.
3. For each file included, create a new `.qmail` file in `/var/qmail/alias` with the contents of the include file and add a forwarding entry to `.qmail-alias`



for the new file.

For example, say /etc/aliases contains this:

```
1 root: erica, rachel
2 users: root, :include:/usr/local/etc/users
3 help: "|/usr/local/bin/autohelp", /usr/local/log/help-mail
```

For line 1, you would create /var/qmail/alias/.qmail-root containing this:

```
&erica
&rachel
```

For line 2, you could create /var/qmail/alias/.qmail-users containing this:

```
&root
&user-list
```

And create /var/qmail/alias/.qmail-user-list containing the contents of /usr/local/etc/users. Make sure that .qmail-user-list is in the proper format: one address per line, each address beginning with a letter, number, or ampersand (&).

For line 3, you *could* create /var/qmail/alias/.qmail-help containing this:

```
 |/usr/local/bin/autohelp
 /usr/local/log/help-mail
```

However, if the delivery to autohelp fails, qmail-local won't deliver a copy to help-mail. To make the deliveries independent, you should put the autohelp delivery into a separate .qmail file. So, /var/qmail/alias/.qmail-help would contain this:

```
&help-autohelp
 /usr/local/log/help-mail
```

And /var/qmail/alias/.qmail-help-autohelp would contain this:

```
 |/usr/local/bin/autohelp
```

NOTE Kunt U het nu nog volgen dan ben U waarschijnlijk heel erg blij. Kunt U het nu nog volgen, dan kunt U meer als mij! :)

Using fastforward to Implement Sendmail-Style Aliases

fastforward is not included with the qmail source tarball. You'll have to download the fastforward source tarball, unpack it, build the binaries, install them, and configure qmail to use fastforward. To do this, follow these steps:

1. Download the source tarball using your Web browser or the `wget` utility. At the time of this writing, 0.51 is the current version of fastforward. For example, using `wget`:

```
$ wget http://cr.yip.to/software/fastforward-0.51.tar.gz
--03:33:40--  http://cr.yip.to/software/fastforward-0.51.tar.gz
      => `fastforward-0.51.tar.gz'
Resolving cr.yip.to... done.
Connecting to cr.yip.to[131.193.178.160]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 40,659 [application/x-gzip]

100%[=====] 40,659      100.78K/s   ETA
00:00

03:33:41 (100.78 KB/s) - `fastforward-0.51.tar.gz' saved [40659/40659]

$
```

2. Unpack the tarball:

```
$ zcat fastforward-0.51.tar.gz | tar xf -
$
```

3. Build and install the binaries:

```
$ cd fastforward-0.51
$ su
Password: rootpassword
# make setup check
( cat warn-auto.sh; \
echo CC=`head -1 conf-cc`; \
echo LD=`head -1 conf-ld`; \
...lots of output ending with something like:
./load instcheck hier.o auto_qmail.o strerr.a substdio.o \
error.a str.a
./instcheck
#
```

4. Convert `/etc/aliases` to the machine-readable format that fastforward uses:

```
# /var/qmail/bin/newaliases
#
```



This will create `/etc/aliases.cdb`

5. Configure `qmail` to use `fastforward`. This accomplished by invoking `fastforward` from `/var/qmail/alias/.qmail-default`, the `.qmail` file that catches all otherwise undeliverable mail. For example, if there is currently no `.qmail-default` file:

```
# cd /var/qmail/alias
# ls .qmail-default
ls: .qmail-default: No such file or directory
# echo "| fastforward -d /etc/aliases.cdb" > .qmail-default
#
```

That's it: Mail to all undeliverable local addresses will now be passed to `fastforward`, which will attempt to deliver mail via `/etc/aliases` before bouncing it as undeliverable.

Compatibility Issues

`fastforward` is not 100% `sendmail` compatible. Making `fastforward` completely `Sendmail`-compatible would introduce some security problems. Other incompatibilities result from additional flexibility or functionality provided by `fastforward`. Others are just a result of the way `fastforward` works. All of these incompatibilities most likely to impact a smooth migration from `Sendmail` to `qmail` are:

- . Aliases in `/etc/aliases` will *not* override valid mail users or their extension addresses. This is because `fastforward` is run from `/var/qmail/alias/.qmail-default`, which is only consulted for undeliverable addresses. The `qmail-users` mechanism can be used to override delivery to valid mail users. See Chapter 3, "Configuring `qmail`: The Basics", for more information on `qmail-users`.

- . File deliveries are not supported. File deliveries can and should be done through extension addresses and `.qmail` files specifying file delivery.

- . `Sendmail`'s behavior with circular aliases depends on the version of `Sendmail` employed. `fastforward`'s behavior is documented in the `setforward` man page.

- . `Sendmail` complains about duplicate aliases. `fastforward` silently uses the first one it finds.

- . `fastforward` doesn't handle backslashes (`\`) and quoting the same as `Sendmail`. `fastforward`'s quoting is Request For Comments (RFC)- compliant, and `Sendmail`'s backslash trick – where `\user` means "deliver to user's mailbox", not "forward to user" -- isn't needed with `qmail` to prevent loops.

- . `fastforward` doesn't allow vertical bars (`|`) before double quotes (`"`).

- . `Sendmail` skips deliveries to aliases containing missing or unreadable `:include:` files. `fastforward` defers such deliveries.



. if an alias includes the same recipient inside and outside of a mailing list, fastforward sends the message twice, once with each envelope sender. Sendmail sends the message once with an unpredictable envelope sender.

. Sendmail reads include files directly. fastforward reads machine-readable versions of include files generated by newinclude.

. fastforward include files cannot include file of program deliveries.

NOTE If you found this excerpt usefull, aiding you in the administration of your qmail mailserver(s), you really should consider purchasing "The qmail Handbook" by Dave Sill. Dave Sill is also author of the well known Life With Qmail (LWQ) <http://www.lifewithqmail.org/lwq.html> online documentation.

Hand typed, by

Robert M. Stockmann
stock@stokkie.net
<http://crashrecovery.org/>



Os support: RedHat, SuSE, Debian, Mandrake, SCO, Solaris HW support: Intel, AMD, Sun, IBM Network: Cisco, 3COM, Nortel