Managing Queue MH

1.0.0 Why MH for managing system mail?

The MH system is very different from most mail user agents. Instead of running one large program which handles all mail functions and keeps messages in one large file, MH is a collection of smaller single-purpose programs used to manipulate mail messages which are kept in individual files. MH may seem more complicated or harder to use than other mail systems, but MH has been designed to allow you to take full advantage of existing Unix commands and programs in connection with mail messages. For example, you can use your usual text editor, spelling program, and printer commands on individual messages. This flexibility makes MH easy to customize. Messages as separate files allows message level locking for multi-user access.

This document describes the basics of what this distribution is all about. To do that, we've added some MH basics for those that aren't MH users yet - ignore if you are. Skip to section 3 for installation instructions.

1.1.0 What have we done here?

It is quite simple actually. We use the concept of a fake user to receive incoming mail which gets automatically incorporated into an MH folder using **mhook**. All users that need access to that MH folder (queue) have a soft link to it. Then we added some scripts to perform regular tasks that provide tracking of the progress of work on the individual messages.

1.2.0 When should you use Queue MH?

This configuration of MH is designed to be used to manage any set of incoming mail that meets the following:

- The mail messages largely consist of requests or questions that require action by the recipient(s).

- Tracking of incoming messages, status of progress, and final outcome is desired.

- Multiple users act on incoming messages.

For example, we use this system for "trouble", "wiring", and "operations" mail queues. Trouble is our main address for support where users send requests and queries to support staff. Wiring is for hardware related requests. Operations is for dump/restore requests. Other system mail is aliased to trouble to ensure that someone sees it and any necessary action is taken, eg. root. Other possible uses for this are mail to postmaster, netnews (usenet), and special mail destinations to fit the support organization at your site.

2.0.0 Fundamental Concepts of Queue MH

2.1.0 Fundamental Concepts of MH

The way MH stores mail messages is much different than more common mail tools like /usr/ucb/mail, where your mail is stored as a single file (usually /usr/spool/mail/<*yourlogin*>). Instead, MH stores **each message as a separate file**. Groups of these files are called *folders*. What the folders are named and which messages are contained in them is decided by the MH user.

Usually, the user's folders are stored in ~/Mail. Within these folders, which are just run-of-the-mill directories, the mail messages are stored as standard text files with their file name being the message number in that folder. Old messages (ones that have been deleted) may be present with the name #<number>. These are easy to delete with a find(1) once a week or so.

Just as an example, here is a listing of ~trent/Mail:

```
aliases      cssun/      inbox/         outbox/       tag/
cm/          drafts/     incoming.log   repl.format   trap/
coggs/       evi/        marga/         sunbugs/      trash/
components   forwcomps   mhl.format     sysadm/       trouble/
context      forys/      misc/          systems/      workshop/
```

The directories you see here are actually used as folders. Some of the other files will be discussed later.

Here is a listing of ~trent/Mail/cm: (my "connection machine" folder)

```
1      11     13     15     17     3      5      7      9
10     12     14     16     2      4      6      8
```

Each of these files is an individual mail message. You could access these by using cat(1) or more(1), but there much nicer tools provided in the MH system, eg. **show**.

2.2.0 Additions for Queue MH

The following executables are part of this distribution. They are described in detail in the users' guide (UsingQueueMH) and man pages.

glom

glom [+<folder>] [-a glommed-message] message-list
This is used to combine messages from the queue folder ($FOLDER) into one message. This is useful when several users report the same problem.

notack

notack [+<folder>]
This command displays messages in the queue folder of which the sender has not received an acknowledgement of receipt. (see **stat**)

res

res [+<folder>] [msg]
This command is used when you are ready to resolve a queue message. It works very similar to **repl**, except that it always *moves* the message to your a temporary file (for locking) and copies the appropriate aliases. The original message is *not* returned to the queue folder (since it has been resolved).

stat

stat [-e] [+<folder>] [msg]
This is used to add status information or to acknowledge to the current queue message. It is basically a **repl** that appends the new status to the original message in the queue. During the reply, the message is *moved* to a tmp file for locking.

tmhappend

tmhappend <filename>
Used by **stat** to append info to the original queue message. It is not intended to be invoked manually.

tmhedit

tmhedit <filename>
Used by **stat** to grab the status portion of the edit session for appending to the original queue message. It is not intended to be invoked manually.

2.2.1 Quick reference to Queue MH

Command	Purpose
glom	combine queue messages of similar topic
notack	scan those queue messages that have not been acknowledged
res	resolve a queue message
stat	add status information to a queue message, acknowledge, or edit

2.3.0 Other queue MH files

These are the other files that queue MH uses to provide consistency and tracking for the mail queues.

2.3.1 Fake queue user files

The following are files that will be installed in the home directory of each fake queue user.

.forward

This file passes all incoming mail to the queue through the MH filter **slocal**.

.maildelivery

This file is used by **slocal** to determine how the incoming mail should be delivered.

Mail/glom.body.format

The MH form file used by **glom**.

Mail/replcomps

The MH form file used by **stat**.

Mail/rescomps

The MH form file used by **res**. This file can be removed if it is not different from *replcomps*.

2.3.2 Setup and prototype user files

The following files are installed in a setup directory.

install
: This script performs the configuration for an individual user of the system queues. It's use is described in the user guide. All it does is copy the following prototype files to the user's home directory and create the necessary folder links to the queue.

.mh_profile
: This file contains information about the user's MH environment, such as the default editor, what options they want to always use with particular commands, etc. The user should rarely need to edit their copy of this file.

Mail/aliases
: File to contain the user's personal mail aliases.

Mail/components
: File that contains the 'blank' message **comp** will use when creating a mesage.

Mail/forwcomps
: This file describes how a forwarded message will look when using **forw**.

Mail/mhl.format
: This file describes how what format will be used to display a message with **show**. This can be easily changed by each user to their liking.

Mail/repl.format
: This file describes what format will be used with replying to a message with **repl**. This should not be changed so that it is consistent for all users of the queue.

2.3.3 Documentation

Man pages for the user commands will be installed. A users' guide is provided, *UsingQueueMH*. It won't be installed anywhere, but it is customized by the installation script for each queue installed. Each customized version should be saved and provided to each new user of a specific queue to help them get started.

3.0.0 Installing Queue MH

3.1.0 Installation Issues

The installation process adds a new user to /etc/passwd and a new group to /etc/group for each queue that is installed. If you are running something like Sun's NIS (YP) or NeXT's Netinfo, make sure the installation is done on the master server and the databases get re-made and propagated.

We recommend that you **not** name a queue as a system account that already exists, eg. root. If you must have your queue name match an existing account, go ahead with the installation normally and remember to manually fix up passwd when you are done. If you aren't sure of what you are doing, don't do it. The filesystem of the home directory of the fake user should have sufficient disk space for operation of the queue (don't put it in /).

Give some thought to what mail aliases or lists that should be copied on **stat** and **res** mail. Our convention is to copy a log file alias for both, and the distribution alias for the group on **res**. This way all mail is tracked, and the other users in the group can keep abreast of what else is going on. Some users like to see all the **stat** mail as well so we allow them to add themselves to the logging alias. The makefile does not create or verify any aliases. See the example in section 4.

The executables are installed with execute permission by group staff. You'll need to change that if staff cannot be used as a group for all queue users.

3.2.0 Installation Procedure

- Un-tar this distribution in a build directory. Hopefully you've already gotten this far.

- Edit the Makefile to provide the paths, queue information and other localisms.

- Execute "make all". This will compile tmhappend and tmhedit and localize all the files with the local information.

- Execute "make install". This will install the executables, user setup files, and man pages.

- Execute "make queue". This will create the fake user for the mail queue.

- Each user to access the new mail queue needs to go through the procedures listed in the users' guide.

- To create additional queues, run "make clean", edit the Makefile with the new queue fake user info, run "make queue" again.

- The "clean" target only removes the queue specific files. Run "make scrub" to remove all the localized files if you change path info.

4.0.0 Example aliases configuration

The following is our configuration as an example.

At CU Boulder, we use the same aliases file on all unix systems on campus. This provides consistency and allows users be free from having to remember home hosts for local users. We are able to do this because we also have a flat username space (finally). Given the size of some universities, this scenario isn't always possible. However, the example is applicable at any scale of shared aliases information.

These are our aliases relevant to queue MH.

```
   trouble:           :include:/usr/local/adm/unixops/trouble.alias
operations:        :include:/usr/local/adm/unixops/operations.alias
wiring:          :include:/usr/local/adm/unixops/wiring.alias  tmr:
troubletrap@boulder,hardt,manchek,jorgy,bri                    wmr:
wiringtrap@boulder   omr:              operatortrap@boulder,kevinj
cstmr:               troubletrap@anchor,dyker,dxb      wiringtrap:
"/usr/local/adm/logs/wiringmail"                      troubletrap:
"/usr/local/adm/logs/troublemail" root:           trouble fortune:
```

`trouble accessgripe: trouble spacegripe: trouble printer: trouble,haleden,shari`

The *trouble*, *operations*, and *wiring* aliases go to their respective MH queues. The use of the :include: aliases allows each machine to go to a different queue. In this way, administrative domains can manage queues separately. For example, mail to *trouble* on korbel goes to trouble@anchor, where mail to *trouble* on kirk goes to trouble@refuge. We have two main system administrative domains reflected here.

Important note: If you use an include alias, it must have a different name than the name of the queue. Otherwise, mail to that alias will loop. In our setup, the queues are actually "troubmh", "omh", and "wmh". So on anchor our include file for trouble contains just the user troubmh. Each user could remake the link to the queue MH folder and call it whatever they wish.

The **mr* aliases are mail record aliases for respective queues. These aliases are copied on **stat** and **res** mail. Notice there are separate record aliases for the anchor and refuge *trouble* queues.

Many other system aliases are aliased to various MH queues like *trouble* in the example above. The :include: alias for trouble also allows additional mail destinations per machine. For example, on latour, the /usr/local/adm/unixops/trouble.alias file looks like: trouble@anchor, schwartz Schwartz is the primary user/owner of that machine. Including him there allows him to get a copy of mail to *trouble* including root mail.

5.0.0 Acknowledgements

This handful of MH stuff is a snapshot of a portion of the evolving local support customs of Unixops and CSops at the University of Colorado at Boulder. It was bundled together and genericized because we had requests from other groups that know about how we use MH. Although we don't support this distribution in a strict sense and are actually even somewhat embarassed about it, we use it daily. Please send any questions, problems or suggestions you have to "systems@colorado.edu".

Significant contributors include (alpha order):

Robert Coggeshall (coggs@boulder.colorado.edu)
Unixops ringmaster

Barb Dyker (dyker@boulder.colorado.edu)
CSops ringmaster and did all the generic packaging

Tinsley Galyean (tag@boulder.colorado.edu)
Co-conspirator in the formative years

John Hardt (hardt@boulder.colorado.edu)
Did much of the (not so) recent evolution

Trent Hein (trent@boulder.colorado.edu)
Co-conspirator in the formative years