# The Rand MH Message Handling System:

# The UCI BBoards Facility

Marshall T. Rose[†]

Wed May 21 21:03:57 PDT 1986

ABSTRACT

This document discusses how to process BBoards using the Rand MH system. In particular, this guide discusses: checking the status of a BBoard, viewing new messages, archive handling, composing mail destined for a BBoard, and replying to a message posted to a BBoard.

Although this document is based on the standard MH user manual[MH], this document is meant to supplement, not supersede, that lengthier work.

Comments concerning this documentation should be addressed to the Internet mailbox `Bug-MH@ICS.UCI.EDU`.

[†] Computer Mail: `MRose@NRTC.NORTHROP.COM`.

# The Rand MH Message Handling System:

# The UCI BBoards Facility

## Acknowledgements

The MH system described herein is based on the original Rand MH system. It has been extensively developed (perhaps too much so) by Marshall Rose and John Romine at the University of California, Irvine. Einar Stefferud, Jerry Sweet, and Terry Domae provided numerous suggestions to improve the UCI version of MH.

In particular, the UCI BBoards facility, which was suggested by Einar Stefferud, has been in place at the University of California, Irvine (in one form or another) for the last two and one-half years. The UCI BBoards facilities runs under both MMDF and SendMail, and, in a more restricted form, under stand-alone MH.

## Disclaimer

The Regents of the University of California wish to make it known that:

> "Although each program has been tested by its contributor, no warranty, express or implied, is made by the contributor or the University of California, as to the accuracy and functioning of the program and related program material, nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the contributor or the University of California in connection herewith."

## Scope

This document explains how to use the UCI BBoards facility to a user familiar with MH and the UNIX[1] operating system in general. A large degree of expertise is not assumed. This document does not attempt to introduce MH to the novice user (for that task, consult the MH tutorial known as [MH.TUT]). Additional information on the programs discussed here (particularly *bbc*) is to be found in [MH].

[1] UNIX is a trademark of AT&T Bell Laboratories.

## Conventions

In this document, certain TEX-formatting conventions are adhered to:

1. The names of UNIX commands, such as *comp*, are presented in *text italics*.
2. Arguments to programs, such as `'msgs'`, are presented in `typewriter style` and delimited by single-quotes.
3. UNIX pathnames and envariables, such as */usr/uci/* and $*SIGNATURE*, are presented in *slanted roman*.
4. Text presenting an example, such as

   ```
   comp␣-editor␣zz
   ```

   is presented in `typewriter style`.

## Introduction

MH is a very powerful message handling system that runs under the UNIX operating system. One of the many features which MH offers is an interface to the UCI BBoards facility. This facility permits the efficient distribution of interest group messages on a single host, a group of hosts under a single administration, and the ARPA Internet community.

Described simply, a interest group is composed of a number of subscribers with a common interest. These subscribers post mail to a single address, known as a *distribution* address. From this distribution address, a copy of the message is sent to each subscriber. Each group has a *moderator*, which is the person that runs the the group. This moderator can usually be reached at a special address, known as a *request* address. Usually, the responsibilities of the moderator are quite simple, since the mail system handles the distribution to subscribers automatically. In some cases, the interest group, instead of being distributed directly to its subscribers, is put into a *digest* format by the moderator and then sent to the subscribers. Although this requires more work on the part of the moderator, such groups tend to be better organized.

Unfortunately, there are a few problems with the scheme outlined above. First, if two users on the same host subscribe to the same interest group, two copies of the message get delivered. This is wasteful of both processor and disk resources.

Second, some of these groups carry a lot of traffic. Although subscription to an group does indicate interest on the part of a subscriber, it is usually not interesting to get 50 messages or so delivered to the user's private maildrop each day, interspersed with *personal* mail, that is likely to be of a much more important and timely nature.

Third, if a subscriber on the distribution list for a group becomes "bad" somehow, the originator of the message and not the moderator of the group is notified. It is not uncommon for a large list to have 10 or so bogus addresses present. This results in the originator being flooded with "error messages" from mailers across the ARPA Internet stating that a given address on the list was bad. Needless to say, the originator usually could not care less if the bogus addresses got a copy of the message or not. The originator is merely interested in posting a message to the group at large. Furthermore, the moderator of the group does care if there are bogus addresses on the list, but ironically does not receive notification.

To solve all of these problems, the UCI BBoards facility introduces a new entity into the picture: all interest group mail is handled by a special component of the mail system. The distribution address maps to a special *channel* that performs several actions. First, if local delivery is to be performed, then a copy of the message is placed in a global maildrop for the interest group with a timestamp and a unique number. Local users can read messages posted for the interest group by reading the file. Second, if further distribution is to take place, a copy of the message is sent to the distribution address in such a way that if any of the addresses are bogus, the failure notice is sent to the maintainer of the group and not the originator.

This scheme has several advantages: First, messages delivered to the host are processed and saved once in a globally accessible area. The UCI BBoards facility supports software which allows a user to query the interest group for new messages and to read those messages in the MH-style. Second, once a host subscribes to an interest group, a user can add or remove him/herself from the list without contacting the moderator. Third, a hierarchical distribution scheme can be constructed to further reduce the amount of message traffic. Fourth, errors are prevented from propagating. When an address on the distribution list goes bad, the request address immediately responsible for the address is notified. Usually, this is the local PostMaster and not the group moderator.

In addition to solving the problems outlined above, the UCI BBoards facility supports several other capabilities. BBoards may be automatically archived in order to conserve disk space and reduce processing time when reading them.

Special alias files may be generated which allow the MH user to shorten address type-in. For example, instead of sending to ``SF-Lovers@Rutgers'', a user of MH usually sends to ``SF-Lovers'' and the MH aliasing facility automatically makes the appropriate expansion in the headers of the outgoing message. Hence, one need only know the name of a interest group and not its address.

Finally, the UCI BBoards facility supports *private* interest groups using the UNIX group access mechanism. This allows a group of people on the same or different machines to conduct a private discussion.

The practical upshot of all this is that the UCI BBoards facility automates the vast majority of BBoards handling from the point of view of both the PostMaster and the user.

## BBoard Handling

Usually the term *BBoard* is used interchangeably with the terms *discussion group* and *interest group*. This is true of the discussion that follows.

The messages for a BBoard delivered locally are kept in the same format as a maildrop.[2] Unlike the user's private maildrop however, the *inc* program is not run to incorporate new BBoard messages into the user's MH `‘‘+inbox’’` folder. The programs which are used will be discussed momentarily.

Each message in a BBoard maildrop has a unique number and a timestamp. The number, called the *BBoard-ID*, is always ascending. The BBoard-ID of a message should **NOT** be confused with the message number of a message, which can change as messages are removed from the BBoard. The BBoard-ID is a value which is unique for every message delivered locally to the BBoard.

To read BBoards, the MH user invokes *bbc*. The *bbc* program has several switches to direct it's action. The `‘-topics’` switch to *bbc* tells the MH user about the status of a BBoard. The `‘-check’` switch to *bbc* lets the MH user check on the activity of a BBoard. The `‘-read’` switch to *bbc* invokes the *msh* program on the BBoard. *msh* is a monolithic program which contains most of the functionality of MH in a single program. These commands are now discussed in greater detail.

### *BBoard status*

The `‘-topics’` option to the *bbc* program can be used to report information about a BBoard that does not pertain to the user's reading habits. If the MH users types

```
bbc␣-topics
```

then *bbc* will list the following information for all BBoards received on the host:

- the official name of the BBoard
- the number of messages delivered to the BBoard (but not necessarily present)

[2] Actually, your site might be running with all BBoards kept on a single host. MH supports the remote access of BBoards using a modified version of the ARPA Post Office Protocol (POP). This has the advantage that it saves a lot of disk space, and incurs only a modest performance penalty.

- the date and time of the last message delivered to the BBoard

In addition to `'-topics'`, if the `'-verbose'` option is given to *bbc*, then more information is listed:

- any aliases the BBoard is known as
- the local leaders of the BBoard
- the file that the BBoard is locally delivered to
- the "global" distribution address
- the "global" request address
- the host that distributes the BBoard to the local host
- the addresses to which this host distributes
- miscellaneous information (presently only archiving status)

Naturally, *bbc* can be invoked with the `'-topics'` option and one or more BBoard names listed on its command line. For example

```
bbc␣-topics␣unix-wizards
```

is completely acceptable — it tells *bbc* to report the status of the BBoard `''unix-wizards''`.

*BBoard checking*

The `'-check'` option to the *bbc* program can be used to check for new BBoard messages in a synchronous fashion (i.e., when you specifically ask for it). The MH users types

```
bbc␣-check
```

and *bbc* consults the profile entry for `''bboards:''` in the user's *.mh_profile* file. For each BBoard listed, *bbc* prints one of several messages depending on the status of both the BBoard and the user's reading habits (for example, in the case of the mythical BBoard `''foo''`):

1. `''foo -- n items unseen''`
   This message indicates items in the BBoard have not been seen by the user. When *bbc* is invoked with the `''quiet''` switch, this is the only informative message that *bbc* will print out. Users of MH usually put

   ```
   bbc␣-check␣-quiet
   ```

   in their *$HOME/.login* file.

2. ‘‘`foo -- empty`’’
   The BBoard is empty.
3. ‘‘`foo -- n items (none seen)`’’
   The BBoard has $n$ items in it, but the user hasn't seen any.
4. ‘‘`foo -- n items (all seen)`’’
   The BBoard is non-empty, and the user has seen everything in it.
5. ‘‘`foo -- n items seen out of m`’’
   The BBoard has at most $m - n$ items that the user has not seen.

It is important to note that *bbc* performs its calculations on BBoard-ID:s and not the messages actually present in a BBoard. This means that the numbers given by *bbc* are maximal end-points. When *bbc* says $n$, *bbc* means "at most $n$".

Naturally, *bbc* can be invoked with the ‘`-check`’ option and one or more BBoards listed on its command line. For example

```
bbc␣-check␣info-c␣poli-sci
```

is completely acceptable — it tells *bbc* to check on the BBoards ‘‘`info-c`’’ and ‘‘`poli-sci`’’ only.

There are two ways to check for new BBoard messages in an asynchronous fashion: using the *CShell* variable $*mail* and running the *useto* program.

**Asynchronous Checking with the CShell**

The *CShell* has a variable called $*mail*. This variable can contain one or more words. Each word should be a filename where the shell should check for new mail. The check is performed after a specified time interval has elapsed just before the shell would prompt the user.

If the first word of $*mail* is a number, then this number specifies a different checking interval, in seconds, than the default, which is 10 minutes.

Whenever the time interval elapses and the shell is ready to prompt the user, the shell looks at the file and decides if new messages have arrived. If so, it says

```
You have new mail.
```

if only one file is present in $*mail*. Otherwise, if more than one file is present in $*mail*, then the shell says

```
New mail in foo.
```

whenever there is new mail in the file called ‘‘`foo`’’.

To find out what file is associated with a BBoard, say ``info-unix'', the MH user types

```
bbc -topics -verbose info-unix
```

Usually the local file for a BBoard has an extension of *.mbox*.

**Asynchronous Checking with Useto**

In contrast to using the *$mail* variable in the *CShell*, the MH user might employ *useto* instead.[3] The *useto* program is a continuous update display that prints information on the status line of your terminal. Needless to say, your terminal must support a status line in order to run *useto*. Not all terminals have this capability, but for those that do it's usually well worth the effort to run *useto*.

For example, users of MH could put

```
useto -bepf 'tcp-ip sftp' '%D %M %d %h:%m%z %b %n.tty%t:%l1'
```

in their *$HOME/.login* file. This command line to *useto* says to inform the user of

- the current date and time
- new mail for the user
- new messages for the BBoards ``tcp-ip'' and ``sftp''
- the name of the host and tty that the user is logged in on
- the 5–minute load average of that host

The *useto* program is really quite amusing and useful.[4]

*BBoard reading*

If *bbc* is not given either the '-check' or '-topics' option, the *bbc* program reads BBoard messages. For each BBoard listed in the MH user's profile entry for ``bboards:'', *bbc* checks to see if there is unread mail. If so, *bbc* starts *msh* on the BBoard, telling *msh* which messages haven't been seen.[5]

When *msh* starts it identifies the BBoard being read and indicates how many messages are present and how many the user has read. Usually, in the user's MH profile, the user has the entry

```
msh: -scan
```

[3] Not all sites have *useto*; contact the same people who supplied MH to get a copy.

[4] To be honest, the author considers computing environments without *useto* to be less than adequate.

[5] If the '-verbose' option is given to *bbc*, then *bbc* will start *msh* on the BBoard regardless of whether there are unseen messages there.

This says that when *msh* starts, it should print a *scan listing* of the messages which the user hasn't seen yet.

The *msh* program now prompts the user for MH commands. The user may type most of the normal MH command. The syntax and semantics of the commands typed to *msh* are identical to their MH counterparts. For example, to reply to a message on the BBoard, the MH user types ``repl''; other MH commands likewise may be applied to BBoard messages. In cases where the nature of *msh* would be inconsistent (e.g., specifying a `+folder` with some commands), *msh* will duly inform the user. In addition to supporting most MH commands, *msh* also has a ``help'' command which gives a brief overview.

The only command that behaves entirely differently in *msh* is the ``mark'' command when given no arguments. The *msh* program maintains a special sequence, ``unseen'', which it uses to keep track of the messages you've seen. If the ``mark'' command is given without any arguments, then *msh* will interpret it as

```
mark␣-sequence␣unseen␣-delete␣-nozero␣all
```

Hence, to discard all of the messages in the current BBoard being read, the MH user types ``mark'' which says to remove all messages from sequence called ``unseen''.

To leave *msh* use the ``quit'' command. This tells *msh* to terminate and *bbc* to go to the next BBoard. Instead, if the user types EOT (usually CTRL-D), then *bbc* will exit as well, updating whatever information was appropriate.

### Current BBoards

There are many, many active interest groups. Consult the BBoard called ``list-of-lists'' for a comprehensive description. Here are a few of the more popular:

`system` Important announcements for the local system are posted here.

`mh-users` A discussion group for users of MH.

`arpanet-bboards` Redistribution address for all known BBoards on the ARPAnet.

`editor-people` Discussion of topics related to computerized text editing, display editors, and human factors in man/machine interaction. The theoretical discussion is catholic, but practical discussion focuses particularly on Tops20[6] and UNIX.

`franz-friends` Discusses the Franz Lisp language.

[6] Tops20 is a trademark of Digital Equipment Corporation.

`header-people` Interest specifically in the format of message headers and related issues such as inter-network mail formats/standards, etc.

`human-nets` Human-Nets has discussed many topics, all of them related in some way to the theme of a world-wide computer and telecommunications network usually called WorldNet. The topics have ranged very widely, from something like tutorials, to state of the art discussions, to rampant speculation about technology and its impact.

`info-micro` Information/discussion list on the general interest topic of microcomputers.

`info-unix` Info-UNIX is intended for question/answer discussion, where "novice" system administrators can pose questions.

`msggroup` Interest in electronic mail, message formats, message systems, and the sociological implications of the above.

`poli-sci` A permanent distributed political "bull" session.

`sf-lovers` Science Fiction lovers. SF-Lovers has discussed many topics, all of them related in some way to the theme of science fiction or fantasy.

`space` Discussions (daily digest) on space-related topics.

`telecom` A broad spectrum moderated-digest-format discussion on telecommunictions technology: the telephone system, modems, and other more technical aspects of telecommunications systems.

`unix-emacs` Used for new release announcements and general discussions of Gosling's EMACS.

`unix-wizards` Distribution list for people maintaining machines running the UNIX operating system.

As discussed earlier, to find out about all of the BBoards that the local host subscribes to, the MH users types

```
bbc␣-topics
```

## More on BBoards

Finally, here are a few more operational details:

### *Creating a BBoard*

Contact the PostMaster at your host to have a BBoard created. Be sure to indicate its status (public or private) and scope (where distribution should occur).

*Subscribing to a BBoard*

If your local host already receives an interest group, then simply add the name of the BBoard to the ‘‘`bboards:`’’ entry in your MH profile. If not, ask the PostMaster to create the BBoard and contact the global request address for you.

*BBoard Archives*

BBoard messages are automatically archived on a weekly basis. Usually, this results in messages older than 12 days being moved to an *archive* area. To read the archives for a BBoard, the ‘`-archive`’ option is used. For example,

```
bbc -archive telecom
```

tells *bbc* to invoke *msh* on the archives for the ‘‘`telecom`’’ BBoard.

Note that the archives may not be present for all BBoards on a given host; also note that the archives may be periodically moved to tape and expunged from the system. Contact your local PostMaster for the details.

*BBoard Addresses*

Each BBoard has associated with it 4 addresses (for example, in the case of the mythical BBoard ‘‘`foo`’’):

`foo` The global distribution list
If you post a message addressed to `foo` then the message is distributed to everyone who subscribes to ‘‘`foo`’’.

`dist-foo` The local distribution list
If you post a message addressed to `dist-foo` then the message is distributed to the local BBoard for ‘‘`foo`’’ and to any sites which the local system distributes to.

`foo-request` The global moderator
If you post a message addressed to `foo-request` then the message is sent to the moderator for the entire interest group called ‘‘`foo`’’.

`local-foo-request` The local moderator
If you post a message addressed to `local-foo-request` then the message is sent to the person responsible for the BBoard ‘‘`foo`’’ on the local system.

These addresses are defined by the MH alias facility. Users of the BBoards facility who do not use MH may not be able to make use of them.

*Leading a BBoard*

Except for special circumstances, this task is wholly automated. For more information though, see the manual entries for *bbl*(1) and *bbleaders*(8).

## Extra for Experts

Some clever MH users might ask why BBoards aren't kept as folders instead of *pack*'d files. This is a good question. Perhaps some future release of MH and the UCI BBoards facility will treat BBoards as a variant of read-only folders.

The problem with *msh*, of course, is that it's a monolithic program, and although it does support input/output redirection and a few other primitive shell-like properties, it's still not the *CShell*.

## References

[MH] M.T. Rose, J.L. Romine. The Rand MH Message Handling System: User's Manual. UCI Version. Department of Information and Computer Science, University of California, Irvine (July, 1984).

[MH.TUT] M.T. Rose. The Rand MH Message Handling System: Tutorial. Department of Computer and Information Sciences, University of Delaware (October, 1984).

# Contents


| | Page |
|---|---|
| Acknowledgements | 1 |
| Disclaimer | 1 |
| Scope | 1 |
| Conventions | 2 |
| Introduction | 2 |
| BBoard Handling | 4 |
| BBoard status | 4 |
| BBoard checking | 5 |
| Asynchronous Checking with the CShell | 6 |
| Asynchronous Checking with Useto | 7 |
| BBoard reading | 7 |
| Current BBoards | 8 |
| More on BBoards | 9 |
| Creating a BBoard | 9 |
| Subscribing to a BBoard | 10 |
| BBoard Archives | 10 |
| BBoard Addresses | 10 |
| Leading a BBoard | 10 |
| Extra for Experts | 11 |
| References | 12 |


This document (version #2.6) was TEXset April 12, 1990 with DISS.STY v103.