

Mail at the command line

LEANER LETTER

The simple mailx command-line mail client handles mail either interactively or via command-line options. Although it lacks the convenience of a GUI-based tool, mailx compares well in scripting.

BY HEIKE JURZIK

Thunderbird, Evolution, or Kmail are not the only options as mail clients. Although the major graphical mail clients offer users a great deal of convenience, sometimes a command-line solution is an advantage – both Mutt [1] and the lean mailx [2] program support batch options in the shell.

Most distributions include mailx by default. OpenSUSE and Debian include the package on their installation media. On openSUSE, `/bin/mail` is a symlink that points to `/usr/bin/mailx`, whereas Debian systems do things the other way around: `/usr/bin/mailx` points to `/usr/`

`bin/mail`. The following commands and command options were all tested on Debian 4.0 “Etch.”

Launching the mail client with `mailx` at the shell prompt takes you into an interactive environment. The client shows the content of your inbox; the default setting for this on most systems is `/var/mail/username`. Then you will see an overview of the messages, including the sender, date, size, and subject line.

Typing `?` and pressing Enter outputs a quick help file about using the mailer. Entering the number and pressing Enter is the the easiest way to display a mes-

sage. The Enter key also scrolls down, although you can use the up and down arrows, too. Pressing `Q` quits the display and takes you back to the main screen. If the messages have scrolled out of the window, you can press `H` and Enter to display them again.

Mailx has two approaches to terminating the program: pressing `X` and Enter quits the mail client without removing messages you have read from the inbox, and `Q` plus Enter does the following:

```
Saved 2 messages in /home/petros
ilie/mbox
```

```
Held 3 messages in /var/mail/pet
rosilie
```

Typing *D* and pressing Enter deletes messages. Again, you can select messages to delete by specifying their numbers. For example, *d 1 3* deletes the first and third messages. Also, you can select a range, as in *d 1-10*, which deletes the first 10 messages, or *d **, which deletes all messages. If you mistakenly delete a message, *u* will undelete it. The command uses the same combinations of digits as the *d* option. Entering *R* lets you answer a message, and pressing *M* starts a new message. To specify the target address, use the *m* command:

```
m petronella@huhnix.org
```

Mailx then prompts you to enter the subject line before you can move on to the message itself.

When you are done, type a period followed by Enter to add more recipients after Cc: (carbon copy):

```
Cc:ole@huhnix.org,max@huhnix.org
```

If you prefer to launch your favorite editor to compose messages, type *~v* directly after the subject line. Mailx evalu-

Using Mailx Options

Sending a copy

```
nonumber
$ mailx -s 'Hello!' petronella@
huhnix.org -c petrosilie@huhnix.
org [...]
```

Message with echo

```
nonumber
$ echo 'Hello, this is a test.' |
mailx -s 'Hello!' petronella@
huhnix.org -c petrosilie@huhnix.
org
```

Swapping out a message text

```
nonumber
$ mailx -s 'Hello!' petronella@
huhnix.org < mail.txt
```

Troubleshooting via status report

```
nonumber
$ <B>echo 'Hello, this is a
test.' | mailx -v -s 'Hello!'
petronella@huhnix.org<B>
Mail Delivery Status Report will
be mailed to <huh>.
```

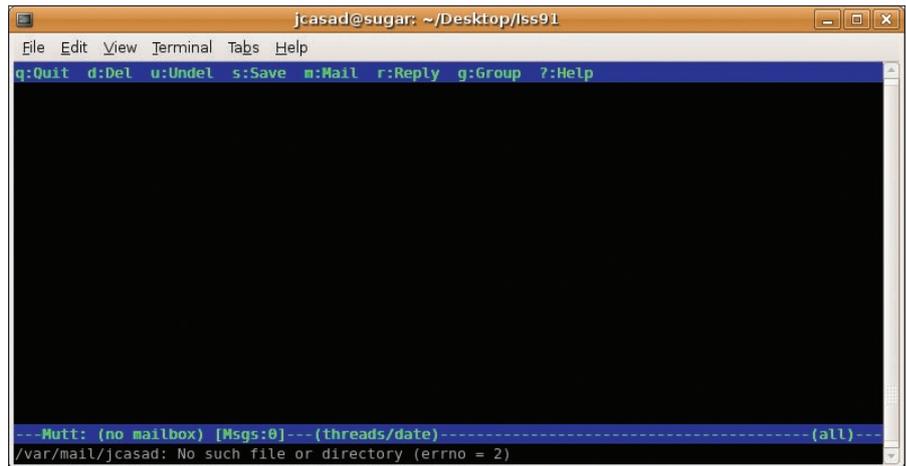


Figure 2: Mutt also comes with a rudimentary GUI, although the keyboard-driven menu system isn't much different from working at the command line.

ates the *\$VISUAL* environmental variable to identify your favorite editor. If you like to work with, for example, Joe, you can set up the editor for mailx by entering *export VISUAL = joe*.

Mailx shows its strength, flexibility, and speed in the shell. The generic command line is:

```
mailx [Option] user@target.tld
```

Users can define options, such as the subject line (*-s*), the mail address for the bcc (blind carbon copy) field (*-b*), further recipients to cc (*-c*), and so on. Don't forget to use quotes for blanks and non-standard characters.

The mail body – that is, the message text – is still missing. For a short message, *echo* is probably a good choice. Swapping the body text out to a file makes sense for longer texts. Mailx can read the file from standard input.

If you need to troubleshoot your setup, *-v* is a big help. On Debian systems, mailx just tells the user that it has

Sending Attachments

Attachments with uuencode and mailx

```
nonumber
$ uuencode hej.jpg hej.jpg >
attach_image.txt
$ mailx -s 'photo' petronella@
huhnix.org < attach_image.txt
```

Attachments with Mutt

```
nonumber
$ mutt -s 'Photo' -a hej.jpg
petronella@huhnix.org < mail.txt
```

sent a status report, which then turns up a short time later in the inbox. The “Using Mailx Options” box shows an example of the use of mailx options.

Using uuencode is the only way to add an attachment to a message. This tool converts binary files, such as images or executables, into printable ASCII characters. After doing so, the text file can be mailed (see the “Sending Attachments” box). The recipient then sees the file as a text block between the keywords *begin* and *end*. Uudecode converts the file back. The command *uudecode attach_image.txt* results in the file *hej.jpg*.

Things are easier if you run Mutt in batch mode. The Mutt mail client not only offers *-s* for subject, *-c* for copy, and *-b* for blind copy options referred to in this article, but also the *-a* parameter for simple attachment sending. See the “Sending Attachments” box. ■

INFO

[1] Mutt: <http://www.mutt.org/>

[2] Mailx: <http://www.scit.wlv.ac.uk/cgi-bin/mansec?1+mailx>

THE AUTHOR

Heike Jurzik studied German, Computer Science and English at the University of Cologne, Germany. She discovered Linux in 1996 and has been fascinated with the scope of the Linux command line ever since. In her leisure time you might find Heike hanging out at Irish folk sessions or visiting Ireland.

