



**By Vikram Vaswani**

This article copyright [Melonfire](#) 2000–2002. All rights reserved.

# Table of Contents

<b><u>The Need For Speed</u></b> .....	<b>1</b>
<b><u>Feature Overload</u></b> .....	<b>2</b>
<b><u>Road Test</u></b> .....	<b>4</b>
<b><u>Room With A View</u></b> .....	<b>6</b>
<b><u>Under The Hood</u></b> .....	<b>8</b>
<b><u>Don't Ask</u></b> .....	<b>9</b>
<b><u>Looking Good, Feeling Better</u></b> .....	<b>11</b>
<b><u>Three's A Crowd</u></b> .....	<b>14</b>
<b><u>Reach Out And Touch Someone</u></b> .....	<b>15</b>
<b><u>Hooking Up</u></b> .....	<b>16</b>
<b><u>Beep Beep</u></b> .....	<b>18</b>
<b><u>Webcrawling</u></b> .....	<b>20</b>

# The Need For Speed

Like most people who get a lot of email, and are compelled (by necessity or boredom) to respond to it, I'm pretty picky about my email client.

Me, I'm not a big fan of those graphically-rich clients, complete with detachable toolbars and searchable help. I'll take economy over excess any day – give me a nice, uncluttered interface, a simple man page and power to spare for those occasions when I need it, and I'm a happy guy. My dream email client is a lot like my dream car, the Ferrari F355 Spider: plain-vanilla seats, a simple dashboard and the ability to accelerate from 0 to 60 miles per hour in under six seconds.

Which is probably why I like Mutt so much. It fits the description above perfectly – it's small, fast, flexible and comes packed with enough features to make most other mail clients crumple up and cry for Mommy. It's also free, which immediately makes it a whole lot more affordable than the F355.

Now, I know there are going to be people out there who disagree with me on this. They're going to say that Mutt sucks, and that they know a dozen email clients which are better/faster/cooler. Take my advice and don't bother listening to their misinformed chatter. Instead, do what I do: drop into fifth and accelerate past them while they're still talking. Very few things are as satisfying.

# Feature Overload

Mutt was originally written by Michael Elkins, though it is now maintained by the user community. It is based in part on the elm mail client, although users of other mail clients will also find commonalities between those clients and Mutt. And, incidentally, that's where Mutt gets its name – it's a hybrid, or mongrel, created from the best features of other mail clients and its own good ideas.

In describing Mutt, the official Web page succinctly says, "All mail clients suck. This one just sucks less." It's an apt description, because Mutt comes with a feature set very few other clients can match. Here are the highlights:

**Performance:** Mutt comes with a very small memory and disk footprint (the compiled version is only about 450K). It's fast without sacrificing performance, which makes it a great choice for use over a network, and on multi-user systems.

**Support for common Internet protocols and technologies:** As a child of the open-source community, Mutt comes with out-of-the-box support for most common protocols and technologies. You can use Mutt to read POP3 and IMAP mail, send and receive MIME-encoded attachments, and encode and decode PGP-encrypted mail. Mutt also comes with support for other mailbox formats, allowing you to easily read and use mail from existing pine, mbox or MH mailboxes.

**Ease of use:** Novices may find Mutt's uncluttered approach to email a little intimidating at first; before long, though, they come to appreciate the simple, even spartan, interface. Since Mutt is text-based, most activity in the client takes place through keystrokes, not menus or buttons (although an unobtrusive toolbar and help system is available with a list of commonly-used keys for new users). color support makes it easy to identify important sections of a message, and "tagging" features allow users to select multiple messages at a time and perform operations on them simultaneously. Installation is a snap, and a detailed manual guides users through the nitty-gritty of configuring the client as per their own personal preferences.

**Key mapping capabilities:** While Mutt's default key bindings are pretty easy to memorize, its creators knew that their first choices for common keyboard operations might not necessarily match yours. Consequently, Mutt is one of the few mail clients which allows you to extensively remap its key bindings, reducing the learning curve and making it even easier to migrate from other clients.

**Support for macros:** Mutt allows you to define keyboard macros to automatically execute certain commands when you hit a particular key, and mailbox "hooks" to automatically perform certain actions when you enter or leave a mailbox. By allowing users to automate certain functions, Mutt goes a long way towards simpler, more efficient mail management.

**Support for mailing lists:** Mutt comes with a whole bunch of features aimed specifically at users who are subscribed to multiple, high-volume mailing lists (read: most geeks). Mutt is intelligent enough to understand which messages are coming to you from a list, and can adjust its behavior to ensure that any replies you send go to the list address, rather than to other subscribers or the list owner (thereby saving unwilling novices from getting flamed); it also comes with powerful message threading capabilities which allow you to sort and view list messages by thread (in much the same way as USENET news)

**Open licensing model:** Like many other open-source tools, Mutt is available for free, to anyone who wants it, over the Internet, in both source and binary form. Users may use and distribute it without restriction, and even

## A Man And His Mutt

charge a fee for it if they so desire.

Active community support: As you might imagine from the above, Mutt has a fairly large fan base, which delights in Mutt's power and is enthusiastic about supporting and improving the code tree. Since the source code for the program is freely available, bug fixes and patches are rapid, and new features are constantly being added to the client by independent developers. A number of mailing lists allow developers and users to share questions, comments and advice, and the Web site also lists a number of useful resources to get new users started.

Intrigued yet?

# Road Test

Getting and installing Mutt is a piece of cake, and shouldn't take you more than ten minutes on a good day. First, drop by the official Mutt Web site, <http://www.mutt.org/>, and pick up a copy of the latest release (currently, version 1.2.5, although, if you're feeling particularly daring, you can road-test the newer 1.3.4 release).

Next, untar the files into your working directory,

---

```
$ tar -xzvf mutt-1.2.5i.tar.gz
```

---

and start the traditional configure-build-install cycle.

There are a number of interesting options you can configure at compile-time – type

---

```
$ ./configure --help
```

---

for a list – but the ones I usually use are the

---

```
$ ./configure --enable-pop --enable-imap
```

---

options, which set up Mutt to access mail on POP3- and IMAP-compatible servers.

Once the configuration is done and a Makefile generated, it's time to build the program

---

```
$ make
```

---

and then install it.

---

```
$ make install
```

---

Unless you've specified an alternate installation path, Mutt (and its ancillary files) should get installed to /usr/local/bin/

You can test your installation by typing in the program name at the command line – Mutt should pop up and display the contents of your mail spool.

## A Man And His Mutt

---

\$ mutt

---

# Room With A View

Like most mail clients, Mutt offers two "views" of your mail. First, there's the list of messages, which Mutt's manual calls the "index". From this index, you can choose to read a particular message, which launches the "pager".

Both index and pager allow you to perform common mail operations – delete, forward, reply, save – and use standard keys for navigation. Here's a quick list of the default navigation keys (you can change these if they don't work for you, keep reading for details):

---

Key	What it does	Where it does it
j	Move one line up	Index
k	Move one line down	Index
z	Move one page up	Index
Z	Move one page down	Index
<enter>	Open message	Index
-	Move one page up	Pager
<space>	Move one page down	Pager
q	Close message	Pager

---

In addition to these navigation keys, Mutt also allows you to perform commonly-used functions just by tapping the appropriate key. Here's what you need to get started:

---

Key	What it does	Where it does it
m	Compose new message	Index, pager
r	Reply to current message	Index, pager
f	Forward current message	Index, pager
d	Delete current message	Index, pager
b	Bounce current message	Index, pager
c	View different mailbox	Index, pager
v	View attachment	Index, pager
a	Create alias	Index, pager
q	Quit	Index, pager

---

You can obtain a complete list from the Mutt manual, or by tapping the

---

?

---

key at any time for detailed help.

## A Man And His Mutt

When it comes to composing new messages or replying to existing ones, Mutt can use any text editor you like. On my system, it defaults to vi; however, you can easily modify it to use emacs, pico or any other editor that strikes your fancy. As soon as you decide to edit a message, Mutt will launch your selected editor, either with an empty document or with quoted message text, and allow you to edit it using standard editor commands. And as soon as you save the file, Mutt will offer you the option of sending it immediately, postponing it or trashing it.

If this seems simple, it is – new users should have very little difficulty getting their arms around Mutt's basic navigation and functions. But don't be fooled – there's a lot more you can do with Mutt, if you're willing to futz around with its default configuration. I'll be discussing this on the next page – try out the functions above and then come back for that.

# Under The Hood

If you only check your mail occasionally, Mutt's default settings will probably work fine for you, and you won't ever need to see a configuration file. If, on the other hand, your mail client is your closest friend, you're probably going to want to alter some of Mutt's factory settings.

Since Mutt doesn't come with a preferences editor – who needs the overhead? – this is accomplished by editing Mutt's configuration file, usually located in

---

```
/etc/muttrc (for system-wide configuration)
```

---

or in each user's home directory

---

```
~/.muttrc (for per-user configuration)
```

---

Mutt's configuration file is pretty easy to understand – it consists of a bunch of variable–value pairs, which allow you to control everything from where Mutt stores your mail to the colors and formatting of email messages. A sample configuration file is included with the Mutt distribution – you should take a look at this well–commented sample to understand the configuration options available.

The variables you'll encounter in this configuration file come in different flavours. Some are Boolean (all you have to do is set them to true to activate them) while others require a specific value (for example, the location of your mail folders or the color to use for quoted text). Mutt has a default setting for every variable in this configuration file; if you don't specify a value for a particular variable, this default setting will be used instead.

The configuration file is read by Mutt every time it starts up, and it's also possible to reconfigure Mutt's behaviour "hot" – while it's running – by editing the variable–value pairs through the client itself.

Let's see how this works in practice, with a few extracts from my own configuration file.

# Don't Ask

By default, Mutt assumes that your mailboxes are stored in a directory named "Mail" under your home directory. If this isn't to your liking (it annoyed me immensely until I found out how to change it), you can specify a different location:

---

```
set folder=~/.mail # mail folder
```

---

While you're at it, you should also define the locations for your inbox, outbox and postponed-messages box:

---

```
set mbox=~/.mail/inbox # location of inbox
set record=~/.mail/outbox # location of outbox
set postponed=~/.mail/postponed # location of postponed
messages
```

---

You already know that the

---

```
a
```

---

key is used to capture email addresses and create aliases for them. You can specify the location of the file which stores these aliases:

---

```
set alias_file=~/.mail/aliases.mutt # location of alias file
```

---

If you'd like this file read into memory every time Mutt starts up, you'll need to source it, like this:

---

```
source ~/.mail/aliases.mutt # read all my aliases into memory
source ~/.mail/aliases.personal
source ~/.mail/aliases.business
```

---

Once these aliases are loaded into memory, you can address messages to recipients using the alias instead of the full email address – Mutt will automatically expand the alias before sending the message.

Next, you'll have noticed that Mutt frequently prompts you for confirmation before deleting or moving messages. This is incredibly useful when you're first getting used to the client, but becomes tedious after a while. Which is why my configuration file also has these lines:

## A Man And His Mutt

```
set postpone=ask-yes # ask me before postponing
set delete=ask-yes # ask me before deleting
set quit=yes # skip exit prompt
```

---

With this in place, Mutt skips the exit confirmation message, but asks for confirmation prior to deleting or postponing messages.

You can also control how Mutt handles composing new messages,

```
set askcc # display CC field
set askbcc # display BCC field
unset edit_headers # don't allow me to edit headers while
composing
```

---

and replying to existing ones.

```
set include=yes # include body of previous message in reply
set fast_reply # skip prompts before replying
set reply_to=yes # use Reply-To field for reply address
```

---

# Looking Good, Feeling Better

Mutt also allows you to control the display of email messages – you can weed out unwanted headers, colorize different sections of a message and decide which fields appear in the message index.

Let's take the headers first. Since I'm not usually interested in seeing all the message headers, I've used the following commands to weed out all but the most important ones:

---

```
ignore * # weed out all headers
unignore date from: to cc subject organization # now show me
these...
```

---

Note, however, that I can still view the complete headers of any message by hitting

---

```
h
```

---

in the pager.

Next up, colors. Mutt allows you to highlight both your message list and different sections of individual messages with its "color" command. Take a look at the following commands, which tell Mutt to always highlight new messages in the index in bold:

---

```
color index brightwhite default ~N # color for new messages
```

---

The first color is the foreground, the second is the background.

If you'd like to, you can also alter the color of Mutt's status bar

---

```
color status brightblue magenta # ugh!
```

---

or even highlight signatures

---

```
color signature brightyellow black # bumblebee
```

---

Within the message body, you can colorize different sections of the message – headers, quoted text, signatures, email addresses, URLs and the like. While this is a nice feature, it's now always as useful as you might think; when I tried it out, the various hues and shades distracted more than they assisted, and I ended up turning them off. Right now, all I use are the following commands, which highlight attachments and quoted

## A Man And His Mutt

text in yellow, search matches in red, and email addresses and URLs in light blue.

---

```
color attachment brightyellow default # file attachments
color search brightred default # search matches
color quoted brightyellow default # quoted text in replies
color quoted1 magenta default # quoted text in replies

color body cyan default "((ftp|http|https)://|news:)[^
>)\\" \t]+" # URLs
color body cyan default "[-a-z_0-9.]+@[-a-z_0-9.]+" # email
```

---

You can customize the message display in the index with the "index\_format" variable, which allows you to specify which elements to display in each message index line. The default setting is

---

```
set index_format="%4C %Z %{%b %d} %-15.15L (%4l) %s" # format
index
```

---

which translates to

---

```
set index_format="number status date sender lines subject"
```

---

Want a no-frills version? This one only displays the date, sender and subject:

---

```
set index_format="%{%b %d} %F (%s)" # format index
```

---

A complete list of formatting codes, together with what each means, is available in the Mutt manual – feel free to experiment with them until Mutt's displays contain exactly what you need.

Mutt also allows you to alter the attribution that appears when you forward or reply to a message. Setting the variable

---

```
set attribution="%n, who happens to be smarter than you,
thinks:"
# format replies
```

---

adds the line



## A Man And His Mutt

---

```
"John Doe, who happens to be smarter than you, thinks:"
```

---

to the top of every reply, while

---

```
set forward_format="Fw: %s" # format forwarded messages
```

---

prefixes the letters "Fw: " to the subject line of forwarded messages.



# Three's A Crowd

Mutt comes with a whole bunch of features designed specifically to help you work with mailing lists – more, in fact, than any other mail client I've seen. There are two basic commands you need to understand here: "lists" and "subscribe".

The "lists" command helps Mutt identify mailing lists which you are not subscribed to – for example, lists to which you can send email but do not receive mail from. The "subscribe" command is used to identify lists which you are actually subscribed to.

In either case, Mutt identifies list messages via the email address specified – so you should make sure that the address is not ambiguous.

Once Mutt knows that a particular message is from a mailing list, it allows you to use its enhanced list reply function, activated by hitting

---

L

---

on your keyboard, to reply directly to the list address, rather than to the original sender of the email.

In addition to this, Mutt attaches a Mail-Followup-To: header to messages that you send to mailing lists. This header tells other mail clients which addresses to use in their replies, and helps to cut down on duplicate messages (assuming, of course, that the other clients in the equation know how to handle this header).

Another very useful feature when dealing with mailing lists is Mutt's message threading feature. If you tell Mutt to sort messages by thread, it will automatically build a hierarchical, indented message tree, allowing you to quickly view messages in context to each other.

You can use the

---

<esc>v (that's Esc-v)

---

key combination to collapse and expand individual threads, and the

---

<esc>V (that's Esc-Shift-V)

---

combination to expand and collapse all threads.

# Reach Out And Touch Someone

If you've compiled Mutt with POP3 and IMAP support, you can use it to retrieve email from a POP3 server, or access mailboxes on an IMAP server. Here's an excerpt from my POP3 configuration section:

---

```
set pop_host="myhost" # POP3 server
set pop_delete=no # don't delete messages after download
set pop_user="vikram" # POP3 username
```

---

Now, every time I hit

---

```
G (that's Shift-G)
```

---

Mutt will connect to the specified POP3 server and retrieve new messages from it. These messages are downloaded and stored in your local mail spool.

IMAP works in a slightly different way. To access mail on an IMAP server, you need to simply select a new folder, and specify the folder name in the following format:

---

```
{imap-server}folder
```

---

Mutt will prompt you for an IMAP username and password, and then connect to the IMAP server to display the messages in the selected folder. You can preset the IMAP user name and password in the configuration file, like this:

---

```
set imap_user="vikram" # IMAP username
set imap_pass="dunno" # IMAP password
```

---

For more information on how to use Mutt with IMAP, there's a good link at the end of this article.

# Hooking Up

As someone who's gone through the experience, I can tell you that relearning the keyboard every time you switch to a new mail client can be quite painful – which is why one of Mutt's nicest features is the ability to customize its keymap.

For example, I personally prefer to use

---

```
c
```

---

to compose a new message, and

---

```
m
```

---

to switch mailboxes. Mutt's default keymap, however, works in the reverse manner. And so, my Mutt configuration file contains the following commands:

---

```
bind index c mail # "c" -> new message in index
bind index m change-folder # "m" -> switch mailbox in index
bind pager c mail # "c" -> new message in pager
bind pager m change-folder # "m" -> switch mailbox in pager
```

---

In a similar manner, you can assign keys to other functions as well. Here's another example, which maps the forward slash symbol (/) to the search command, as in vi,

---

```
bind pager / search
```

---

A complete list of the functions available in each of Mutt's menus is available in the manual, together with the default key assignment for each.

It's also possible to create keyboard macros, which allow you to execute a sequence of commands using a single keystroke. The most frequently-used example of this is setting up the external "urlview" program to harvest URLs from an email message and browse to them using lynx.

These commands set things up so that hitting

---

```
^u (that's Ctrl-U)
```

---

## A Man And His Mutt

within the index or pager launches urlview and displays a list of URLs found in the message.

---

```
macro index \cu |urlview\n # "^u" -> launch urlview in index
macro pager \cu |urlview\n # "^u" -> launch urlview in pager
```

---

Mutt also allows you to alter your configuration on the basis of specific rules – a feature referred to in Mutt-lingo as a "hook". For example, suppose I wanted to sort all messages in one mailbox by date, and messages in another by sender. Normally, I would have to manually resort the data each time; however, with Mutt's folder hooks, I can have it taken care of automatically.

---

```
folder-hook =inbox set sort=date-sent # sort inbox by
date
folder-hook =lists/melonfire-dev set sort=threads # sort
mailing lists
by thread
```

---

In addition to folder hooks, Mutt also offers hooks for saving and sending messages. Take a look at the manual, or the sample configuration files listed at the end of this article for more details.

# Beep Beep

A bunch of other miscellaneous variables and functions keep things interesting. Here's a quick list:

You can force Mutt to append a particular host name to every unqualified email address (handy if most of your mail is to users on the same domain) with the "hostname" variable.

---

```
set hostname="melonfire.com" # attach default hostname
```

---

You can tell Mutt to check for new email at periodic intervals with the "mail\_check" and "timeout" variables,

---

```
set mail_check=60 # check mail every 60 seconds
set timeout=30 # 30 seconds inactivity before checking
mail
```

---

and have it let you know with a beep.

---

```
set beep_new # beep if new mail comes in
```

---

By default, Mutt only checks your mail spool for new mail. In case you have an agent running in the background and filtering your email into different mailboxes – think procmail – then this feature isn't going to be all that useful to you. Which is why Mutt also allows you to specify a list of additional mailboxes it can check for new mail.

---

```
mailboxes =lists/melonfire-dev # mailboxes which get new mail
mailboxes =irc-friends
```

---

In case you have multiple email addresses, you can use the "alternates" variable to let Mutt know about them. This helps the program to identify which messages are addressed to you, and display status flags accordingly.

---

```
set alternates="\^(\  
(icarus@somehost\\.com)\  
|(i@someotherhost\\.com)\  
)$" # these are all my addresses
```

---

You can alter the "From: " field on every email message sent out with the "from" variable,

## A Man And His Mutt

```
set from="The Geek Prince <g@somehost.com>" # reset From:
field to this
```

---

and add custom headers to your email with the "my\_hdr" command.

---

```
my_hdr Weather: Cloudy with prospects of rain # custom
header
```

```
my_hdr Web_Ad: Visit us online at http://www.melonfire.com/ #
custom
header
```

---

And, in the same vein, you can have Mutt automatically attach a "User-Agent:" header identifying itself to every message you send out.

---

```
set user_agent # add User-Agent: header
```

---

# Webcrawling

And that just about brings us to the end of this guided tour of the wonder that is Mutt. I hope you found it interesting, and are perhaps even considering shifting allegiance to this fast and powerful mail client. If you're looking for more information on the topic, you might want to consider visiting the following links:

The official Mutt Web site, at <http://www.mutt.org/>

The Mutt FAQ, at <http://www.fefe.de/muttfaq/faq>

Sample Mutt configuration files, at <http://mutt.org/links.html#config>

Mutt and IMAP, at <http://mutt.sourceforge.net/imap/>

Till next time...stay healthy!

Note: All examples in this article have been tested on Linux/i586 with Mutt 1.2.5i. Examples are illustrative only, and are not meant for a production environment. YMMV!