Reading Mail with Gnus

Table of Contents

Project XXX: Send Mail in Emacs....................2
Project XXX: Receive Mail in Gnus....................4
Synchronize Mail Locally....................4
Fetch Mail with Fetchmail....................5
Connect Directly to Your Mail Server....................6
Project XXX: Work with Mail in Gnus....................7
Project XXX: Read HTML Mail....................9
Project XXX: Search Mail....................9
Set up IMAP and NNIR....................10
Set up POP3 and NNIR....................11
Search Your Mail with NNIR....................11
Project XXX: Organize Mail into Groups....................12
Automatically File Mail....................12
Find Your Groups....................13
Split Mail - Fancy....................14
Project XXX: Filter Spam....................16
Project XXX: Score Mail....................18
Configure Adaptive Scoring....................18
Create Scores by Example....................19
Create Complex Scoring Rules 20
Project XXX: Set up Posting Styles....................21

Yes, you can read mail in Emacs. In fact, there are more than five mail clients to choose from, ranging from simple message readers to complex mail clients that integrate with many other modules. In this chapter, you'll learn how to set up and use Gnus, a feature-rich mail client that comes with Emacs 22.

Why Emacs when there are already plenty of desktop and web-based mail clients to choose from? I keep coming back to Emacs and Gnus because all the other e-mail clients I've tried seem limited in comparison. Here are some of the things I love about Emacs:

- The customizable, keyboard-friendly interface will help you get through your mail without the distraction of advertising and unnecessary toolbars, graphics, and buttons. (Although you can still read HTML e-mail if you want to!)
- Gnus supports several mail search engines that can help you quickly retrieve messages based on keywords, author, date, and other criteria. (Project XXX: Search mail with Gnus)
- If you're on several mailing lists or you receive plenty of mail, you'll like how Gnus can help bring interesting mail to your attention. You can give mail higher or lower scores based on subjects, authors, keywords, or any combination thereof, making interesting messages float to the top. (Project XXX: Score messages)

- Gnus can also help you manage multiple identities. For example, you may want to use one e-mail signature for work, another signature for your personal mail, and yet other signatures for mailing lists. You may also want to change your return address or other details. See "Project XXX: Set up profiles in Gnus" for details.
- You'll also appreciate the integration with other components of Emacs, such as a contact database (see Chapter 6: Being Big Brother) and personal information managers that can help you take notes (Chapter 7), manage your schedule (Chapter 8), or keep track of your tasks (Chapter 9).

You can enjoy all of those features without giving up the convenience of your existing web-based or desktop e-mail client. If your mail server supports the Internet Mail Access Protocol (`IMAP`), you can synchronize your mail in Gnus with your mail in your regular e-mail client. For example, I use Emacs to read my Google Mail, and the messages I read offline are marked as read when I synchronize with the web copy. See **Project XXX: Set up Gnus with Synchronized IMAP** for more details.

Ready to start? Let's set up your system to send mail.

Project XXX: Send Mail in Emacs

First, let's find out if you already have a mail transfer agent (MTA) set up and ready to go. Type **M-x** mail to create a mail buffer. Fill in the details and e-mail yourself a note, typing **C-c C-c** (`mail-send-and-exit`) to send the message. Wait a few minutes, then check your mail using another mail client. If the message showed up, great! Skip this section. If the message didn't show up, read on.

If you're the administrator of a Linux or Unix-based system, you'll probably want to set up a proper mail transfer agent that can be used by other programs. That's outside the scope of this book, but you can probably find information in your distribution's administration guide. If you're in Microsoft Windows, you will probably want to set up a small mail transfer agent such as msmtp (*http://msmtp.sourceforge.net*), as the Emacs built-in modules for securely sending mail require extra programs that are not available for Microsoft Windows. Read the documentation for your mail transfer agent, configure it, check it by sending a message outside Emacs, then try sending a message from Emacs.

If you can't set up a mail transfer agent, you can use Emacs' built-in mailer. Emacs comes with a mail-sending library called smtpmail.el, which takes your messages and sends them to a mail server using the Simple Mail Transfer Protocol (SMTP). The mail server will then take care of passing on the messages through the Internet. Your Internet service provider will typically provide an SMTP server, and may block you from directly connecting to other SMTP servers so as to reduce spam. Some web-based mail services provide SMTP servers. Read their help files for more information.

Once you have the SMTP server name and other needed connection details such as port, username or password, you can set up smtpmail.el for sending mail. For example, if your provider lets you send mail through its SMTP server smtp.yourisp.com, add something like this to your *~/.emacs*:

```
(setq smtpmail-smtp-server "smtp.yourisp.com")
(setq send-mail-function 'smtpmail-send-it)
(setq message-send-mail-function 'smtpmail-send-it)
(require 'smtpmail)
```

For SMTP servers outside your Internet service provider, you may need to connect to a non-standard port and specify authentication details. This authentication is handled by the smtpmail-auth-credentials variable. SMTP servers that require authentication typically also require a secure connection through either Transport Layer Security (`TLS`) or Secure Sockets Layer (`SSL`). For security, you'll need either gnutls-cli (`part of the GNU TLS package, which you can get from http://www.gnu.org/software/gnutls/`) or starttls (`get it from the archives of ftp://ftp.opaopa.org/pub/elisp/`). Both programs work on Linux, but not Microsoft Windows. After you set either program up, change your smtpmail-related code in *~/.emacs* to the following:

```
(setq smtpmail-smtp-server "smtp.example.com")     ❶
(setq smtpmail-smtp-service 587)                   ❷
(setq smtpmail-auth-credentials                    ❸
      (list (smtpmail-smtp-server
             smtpmail-smtp-service
             "yourusername"
             "yourpassword"))
```

Set your SMTP server❶ and port❷. SMTP servers usually use port 25, but some offer alternative ports so that you can use them even if port 25 is blocked by a firewall. (`Make sure your own computer lets Emacs connect to this port.`) Authentication information is stored in smtp-auth-credentials❸, which reuses the SMTP server and port and adds your username and password. If you change your server and port, you will need to set smtpmail-auth-credentials again.

You may also want to define a signature file that contains your contact information, a pithy saying, or anything that you would like to attach to the bottom of each message you send. Emacs includes the contents of ~/.signature (`or the file indicated by mail-signature-file`) in messages you create. Go ahead and set this to something like

```
Firstname Lastname
Long live Emacs!
```

When you compose a new message, your signature will be included.

Project XXX: Receive Mail in Gnus

There are several ways to get your mail into Gnus and several ways to store your mail once it's there. If you're on Linux or a UNIX-like system, then you can retrieve your mail outside Emacs and read it inside Emacs. If you're on Microsoft Windows, you will probably need to connect directly to your mail servers in order to retrieve or read your mail.

Synchronize Mail Locally

If your mail account supports the Internet Mail Access Protocol (IMAP)–and most do–and you want to be able to access your mail from other interfaces such as Gmail, then the best way is to run your own IMAP server locally. I use Dovecot IMAPd to store the files in Maildir format. Then I synchronize it with my mail server using OfflineIMAP.

Install Dovecot IMAPd using your distribution's package manager, or install it from *http://www.dovecot.org/* . After you install Dovecot, edit */etc/dovecot.conf* and set the following line:

```
    default_mail_env = maildir:%h/Maildir
```

Maildir will be faster to access under Gnus. Then install and set up OfflineIMAP from your distribution or from *http://software.complete.org/software/projects/show/offlineimap* . Here's an example *~/.offlineimaprc*:

```
[general]
accounts = ISP
maxsyncaccounts = 1
[Account ISP]
localrepository = Local
remoterepository = Remote

[Repository Local]
type = IMAP
remotehost = localhost
port = 143
remoteuser = YOUR_LOCAL_USERNAME ❶
remotepass = YOUR_LOCAL_PASSWORD ❷

[Repository Remote]
type = IMAP
remotehost = imap.yourserver.com
```

```
remoteuser = YOUR_REMOTE_USERNAME ❸
remotepass = YOUR_REMOTE_PASSWORD ❹
maxconnections = 1
realdelete = no
folderfilter = lambda foldername: foldername in ['INBOX'] ❺
ssl = yes
```

Set your user name❶ and password❷ on your computer, your user name❸ and password❹ on your mail server❷, and the folders that you would like to retrieve❺. If you specify your passwords❷❹, make sure that the file is protected from other people reading it by using the command `chmod go-rw ~/.offlineimaprc`. You can also omit the lines specifying your passwords. Run offlineimap, and it should synchronize your mail.

Configure Gnus to read from your local IMAP server by adding the following to your *~/.gnus*, creating it if necessary:

```
(setq gnus-select-method
      '(nnimap "Mail"
               (nnimap-address "localhost")
               (nnimap-stream network)
               (nnimap-authenticator login)))
```

This tells Gnus to read the mail from the local IMAP server. Save the file and type **M-x gnus** to start up Gnus, and you should see your INBOX in the list of groups. Press **RET** on the group to enter it, then read **Project XXX: Work with Mail in Gnus** to learn more.

Fetch Mail with Fetchmail

If you don't need to read your mail using anything else, then you can use Fetchmail or similar tools outside Emacs to download your mail to your computer. This allows you to keep using Emacs while your mail is being downloaded.

Install Fetchmail from your distribution or from *http://fetchmail.berlios.de/* . Here is an example *~/.fetchmailrc*:

```
poll YOUR_MAIL_SERVER  ❶
  protocol IMAP  ❷
  user "YOUR_REMOTE_USERNAME"  ❸
  password "YOUR_REMOTE_PASSWORD"  ❹
```

Specify your mail server❶, the protocol used to access it (IMAP or POP3)❷, and your username❸ and password❹ on the mail server.

Make sure other people can't read the file with the shell command `chmod go-rw .fetchmailrc`. Then type `fetchmail -v` at the command-line to retrieve the mail with some debugging output, or just `fetchmail` to retrieve the mail quietly. You can set it to retrieve mail every X seconds with the command `fetchmail -d X`. For example, to have it check mail every five minutes, type `fetchmail -d 300`.

Now that you've retrieved your mail, you can set up Gnus to read it locally. Add the following to your *~/.gnus*, creating it if necessary:

```
(setq gnus-select-method '(nnml "")) ❶
(setq mail-sources '((file :path "/path/to/mail/file"))) ❷
```

This configures Gnus to use the nnml mail storage backend❶ and to check your mail file❷ for new mail. You will need to set that to the location of your mail file. This depends on your system. On my system, the mail file is at */var/spool/mail/USERNAME*.

When you type **M-x gnus**, Gnus will move the mail from your mail file to individual messages in the *~/Mail/* directory.

Connect Directly to Your Mail Server

Gnus can also directly connect to mail servers, although accessing a remote server can be a little slow. If you're on Microsoft Windows, this is probably how you'll need to access your mail. To connect to an IMAP mail server, put the following in your *~/.gnus*:

```
(setq gnus-select-method '(nnml "")) ❶
(setq gnus-secondary-select-methods
      '((nnimap "mail"
            (nnimap-address "your_mail_server")  ❶
            (nnimap-stream network) ❷
            (nnimap-authenticator login) ❸
            (nnimap-port 143))))
```

This configures Gnus to use the nnml mail backend❶ and connect to your IMAP server❷. By default, Gnus uses a plain network connection. If you need to connect through SSL, change the `nnimap-stream` to `ssl`❸. This configuration sends your username and password as plain text, which most IMAP servers support. If your server supports more authentication features, check out the Info manual for Gnus to learn more about `nnimap-authenticator`. You can save your authentication information in a file by adding another configuration setting like this:

```
(setq gnus-select-method '(nnml ""))
```

```
(setq gnus-secondary-select-methods
      '((nnimap "mail"
                (nnimap-address "your_mail_server")
                (nnimap-stream network)
                (nnimap-authenticator login)
                  (nnimap-authinfo-file "~/.authinfo")
                (nnimap-port 143))))
```

Here is an example *~/.authinfo*:

```
machine your_mail_server login your_remote_username password your_remote_password port imap
```

When you type **M-x gnus**, Gnus will connect to your mail server (prompting you for login information if necessary) and display the available mailboxes. Press **RET** to enter a mailbox, then read **Project XXX: Work with mail in Gnus** to learn more.

If your mail server does not support IMAP, it may support POP3. Put the following in your *~/.gnus*:

```
(setq gnus-select-method '(nnml ""))   ❶
(setq mail-sources '((pop :server "YOUR_MAIL_SERVER"   ❷
                          :user "YOUR_REMOTE_USERNAME")))    ❸
```

This sets your local mail backend to `nnml`❶ and configures Gnus to connect to your mail server❷ using the user name you specified❸. When you type **M-x gnus**, Gnus will connect to your mail server (prompting you for login information if necessary), download your mail, then display a list of groups. Press **RET** to enter a mailbox, then read **Project XXX: Work with mail in Gnus** to learn more.

Project XXX: Work with Mail in Gnus

Gnus is probably the most unusual mail client you will ever come across. That's because Gnus isn't really a mail client - it's a newsreader that geeks built to help them handle the large volumes of Usenet news. As a result, Gnus has many features that can help you with e-mail overload–and it also has some idiosyncrasies you'll need to watch out for and work around.

After you set up Emacs to send and receive mail (see **Project XXX: Send mail in Emacs** and **Project XXX: Receive mail in Gnus**), type **M-x gnus** to get to the main Gnus screen. First, gnus checks your mail or connects to the server you specified in *~/.gnus*. After that, Gnus displays a list of groups, which are similar to mail folders. Gnus also indicates the number of unread messages in each group. Press **RET** (`gnus-group-select-group`) on a group to see the messages in the group.

You will then see the Gnus summary view for that group. Here are some handy keyboard shortcuts:

Description	Shortcut	Command
Displays the selected message or scrolls down the displayed message	**RET**	`gnus-summary-scroll-up`
Show the next page of the selected message or go to the next unread message	**SPC**	`gnus-summary-next-page`
Show the previous page of the selected message	**<backspace>**	`gnus-summary-prev-page`
Reply to the author of the message, quoting the original message	**R**	`gnus-summary-reply-with-original`
Reply to the author of the message, but don't include the original message	**r**	`gnus-summary-reply`
Reply to all the people in the To:, From:, and Cc: fields, quoting the original message	**F**	`gnus-summary-followup-with-original`
Reply to all the people in the To:, From:, and Cc: fields, but don't include the original message	**f**	`gnus-summary-followup`
Compose a message	**m**	`gnus-summary-mail-other-window`
Delete the current message	**B DEL**	`gnus-summary-delete-article`
Move the current message to a different group	**B m**	`gnus-summary-move-article`
Exit the current group and go back to the group selection screen	**q**	`gnus-summary-exit`

Table 1: Keyboard shortcuts for the Gnus summary view

When you compose a message or a reply, type **C-c C-c** (`message-send-and-exit`) from the message buffer in order to send it.

Gnus makes it easy to delete, reply to, or move more than one message at a time by using process marks. To select multiple messages in the summary buffer, press **#** (`gnus-summary-mark-as-processable`). You can also specify the number of messages to mark by using the **C-u** prefix. For example, to mark the next five messages, type **C-u 5 #** (`gnus-summary-mark-as-processable`). You can mark all the visible messages with **M P a** (`gnus-uu-mark-all`). To unmark a message, type **M-#** (`gnus-summary-unmark-as-processable`). To unmark all messages, type **M P U** (`gnus-summary-unmark-all-processable`). The shortcuts for deleting, replying to, and moving messages all work with process marks.

When you exit and re-enter a group, all the messages you've already read will be hidden from view. Don't panic. You can view old messages by typing **/ o** (`gnus-summary-insert-old-articles`) and pressing **RET** to accept the default of all messages, or specifying the number of old messages you want to retrieve. To keep a message visible, press **!** (`gnus-summary-tick-article-forward`) to flag it as important. To remove the flag, press **M-u** (`gnus-summary-clear-mark-backward`).

Project XXX: Read HTML Mail

Thanks to spammers, phishers, and all sorts of nefarious malcontents on the Internet, it is generally not a good idea to automatically render HTML mail in Gnus. To disable HTML mail entirely, add the following line to your *~/.gnus*:

```
(setq mm-automatic-display (remove "text/html" mm-automatic-display))
```

If you must read the occasional HTML message, however, you can manually display it by adding the following to your *~/.gnus*:

```
(setq mm-text-html-renderer nil)
(defun wicked/gnus-article-show-html ()
 "Show the current message as HTML mail."
 (interactive)
 (let ((mm-automatic-display (cons "text/html" mm-automatic-display)))
   (gnus-summary-show-article)))
(define-key gnus-article-mode-map "WH" 'wicked/gnus-article-show-html)
```

You can then use **W H** (`wicked/gnus-article-show-html`) to show the current message in an external browser. To configure this to open a different web browser, use **M-x customize** to change the value of `browse-url-browser-function`. For more details, read **Project XXX: Set your default web browser**.

Project XXX: Search Mail

There are several ways to find messages in Emacs. From the summary buffer, you can use / o (`gnus-summary-insert-old-articles`) to display all or some old messages. You can then scan through the headers in the summary buffer by using **C-s** (`isearch-forward`), or you can limit the displayed messages with these commands:

Description	Shortcut	Command
Messages from a given author	/ a	gnus-summary-limit-to-author
Messages whose subjects matching a given regular expression	/ /	gnus-summary-limit-to-subject
Messages that match a given extra header	/ x	gnus-summary-limit-to-extra-headers
Messages at least N days old	/ t	gnus-summary-limit-to-age

Table 2: Shortcuts for filtering messages

Limits work on the messages that are currently displayed, so you can apply multiple limits. If you make a mistake, use **/ w** (`gnus-summary-pop-limit`) to remove the previous limit. You can repeat **/ w** (`gnus-summary-pop-limit`) until satisfied. To remove all the limits, type **C-u / w** (`gnus-summary-popl-limit`).

If you specify a prefix, the limit's meaning is reversed. For example, **C-u / a** (`gnus-summary-limit-to-author`) will remove the messages from the matching author or authors.

You can use Gnus to search the currently-displayed messages by using **M-s** (`gnus-summary-search-article-forward`) and **M-r** (`gnus-summary-search-article-backward`).

If you want to search a lot of mail, you'll find NNIR handy. NNIR is a front-end to mail search engines which can index your mail and return search results quickly. If you want to use NNIR with a local or remote IMAP server, you will need to use *nnir.el* and *imap.el*. If you download your mail using fetchmail or connect to a POP3 server and use an nnml backend, you can use NNIR with a search engine such as swish-e to search your *~/Mail* directory efficiently.

Set up IMAP and NNIR

If you use IMAP, then your mail is stored on the mail server and you'll need to use the IMAP search interface to search through it. Download *nnir.el* from *http://www.emacswiki.org/cgi-bin/wiki/download/nnir.el* and save it to your *~/elisp* directory. You will also need an *imap.el* that is newer than the one that comes with Emacs 22. Download *imap.el* from *http://www.emacswiki.org/cgi-bin/wiki/download/imap.el* and save it to your *~/elisp* directory as well. Because Gnus comes with an older version of *imap.el*, you will need to make sure that the new version of *imap.el* is loaded. Add the following to your *~/.gnus*:

```
(add-to-list 'load-path "~/elisp")
(load-file "~/elisp/imap.el")
(require 'nnir)
```

Restart your Emacs. You can check if the correct version of *imap.el* has been loaded by typing **M-x locate-library** and specifying `imap.el`. If Emacs reports `~/elisp/imap.el`, then Gnus is configured to use the updated *imap.el*.

Set up POP3 and NNIR

If you use the configuration for POP3 that is suggested in this chapter, then your mail is stored in the nnml backend, which uses one file per message. To search this using NNIR, to install nnir.el and an external search mail engine. The Namazu search engine runs on Linux, UNIX, and Microsoft Windows, so that's what we'll talk about here. To find and configure other mail search engines supported by NNIR, check out the comments in nnir.el.

First, you'll need to download and install Namazu. If Namazu is available as a package for your distribution, install it that way, as it depends on a number of other programs. An installer for Microsoft Windows can be found at *http://www.namazu.org/windows/* . If you need to build Namazu from source, you can get the source code and instructions from *http://www.namazu.org* .

After you've installed Namazu, create a directory for Namazu's index files, such as *~/.namazu-mail*. Then index your mail by typing this at the command-line:

```
mknmz --mailnews -O ~/.namazu-mail ~/Mail
```

and add the following to your *~/.gnus*:

```
(add-to-list 'load-path "~/elisp")
(require 'nnir)
(setq nnir-search-engine 'namazu)
(setq nnir-namazu-index-directory (expand-file-name "~/.namazu-mail"))
(setq nnir-namazu-remove-prefix (expand-file-name "~/Mail"))
(setq nnir-mail-backend gnus-select-method)
```

Search Your Mail with NNIR

From the group buffer displayed by **M-x gnus**, you can type **G G** (`gnus-group-make-nnir-group`) to search your mail for keywords. If you're using the Namazu search engine, then you can use more sophisticated search queries such as:

Query	Description
Linux Emacs	messages that contain both "Linux" and "Emacs"
Linux or Emacs	messages that contain either "Linux" or "Emacs"
Emacs not Linux	messages that contain "Emacs" but not "Linux"
Emacs and (Linux or Windows)	messages that contain "Emacs" and either "Linux" or "Windows"
"apple pie"	messages that contain the phrase "apple pie"
{apple pie}	messages that contain the phrase "apple pie"
+from:example@example.com	messages with example@example.com in the From: header
+subject:"apple pie"	messages with the phrase "apple pie" in the Subject: header
+subject:apple +subject:pie	messages whose Subject: headers contain both "apple" and "pie"

Table 3: Sample queries

If matching messages are found, then you will see a temporary group with the results. Although you can't delete messages from this view, reading and replying to these messages is the same as reading and replying to regular messages.

To see a message in its original context, type **G T** (`gnus-summary-nnir-goto-thread`) from the summary buffer. This opens the message's original group. If Gnus asks you how many articles to load, press **RET** to accept the default of all the articles.

Project XXX: Organize Mail into Groups

People handle large volumes of mail in different ways. Keeping everything in one mailbox can quickly become unmanageable because messages you need to read get lost among messages you don't need to read.

You can move mail manually by selecting them in the summary buffer and typing **B m** (`gnus-summary-move-article`). Then type the name of the group to which you would like to move the message. The group will be created if it doesn't exist.

To move multiple messages, mark them with **#** (`gnus-summary-mark-as-processable`) and then type **B m** (`gnus-summary-move-article`). To unmark a message, type **M-#** (`gnus-summary-unmark-as-processable`). To unmark all messages, type **M P U** (`gnus-summary-unmark-all-processable`).

Automatically File Mail

Moving messages by hand is tedious and time-consuming. One way to deal with this is to set up rules that automatically file mail into different groups (or folders, as they're called in other mail clients). Gnus calls this "splitting" mail, and you can split mail on IMAP servers as well as mail downloaded from POP3 servers to your computer.

For example, if you're using Gnus to read mail from an IMAP server, you can split your messages by adding this to your *~/.gnus*:

```
(setq nnimap-split-inbox "INBOX") ;; ❶
(setq nnimap-split-predicate "UNDELETED") ❷
(setq nnimap-split-rule
      '(
        ("INBOX.emacs" "^Subject:.*emacs")
        ("INBOX.work" "^To:.*you@work.example.com")
        ("INBOX.personal" "^To:.*you@personal.example.com")
        ("INBOX.errors" "^From:.*\\(mailer.daemon\\|postmaster\\)")
       ))
```

If you use a different inbox, change the value of `nnimap-split-inbox`❶. Any messages in the inbox will be split according to `nnimap-split-rule`❷, which is a list where each element is a list containing the group's name and a regular expression matching the header of messages that should be filed in the group. In this example, Gnus will move mail with subjects containing the word "emacs" to INBOX.emacs, mail directed to you@work.example.com to the INBOX.work group, mail directed to you@personal.example.com to the INBOX.personal group, and mail error messages to INBOX.errors. All other messages will be stored in INBOX.

If you're downloading your mail from a POP3 server and storing it in nnml, add this to your *~/.gnus* instead:

```
(setq nnmail-split-methods
     '(
       ("mail.emacs" "^Subject:.*emacs")
       ("mail.work" "^To:.*you@work.example.com")
       ("mail.personal" "^To:.*you@personal.example.com")
       ("mail.errors" "^From:.*\\(mailer.daemon\\|postmaster\\)")
      ))
```

All other messages will be stored in mail.misc.

Start **M-x gnus** again, and your mail will be split into the different groups.

Find Your Groups

If you don't see your new groups in the group buffer displayed by **M-x gnus**, type **A A** (`gnus-group-list-active`) to see all the groups. Go to the group that you would like to add to the group buffer, then type u (`gnus-group-unsubscribe-current-group`) to toggle its subscription. In this example, INBOX.automated is not subscribed to, but INBOX is.

```
 U    13: INBOX.automated
      76: INBOX
```

When you type **M-x** gnus again, you'll see your subscribed groups if they have unread messages.

`nnimap-split-rule` and `nnmail-split-methods` allow you to filter interesting or uninteresting mail into different groups based on their headers. Gnus comes with an even more powerful mail splitting engine: fancy mail splitting.

Split Mail - Fancy

With fancy mail splitting and some configuration, you can split mail based on a combination of criteria. You can even manually file a message and have Gnus automatically file incoming replies in the same group.

To configure an IMAP connection to use fancy mail splitting, add the following to your *~/.gnus*:

```
 (setq nnimap-split-inbox "INBOX")
 (setq nnimap-split-predicate "UNDELETED")
 (setq nnmail-split-fancy ;; ❶
       '(|                                  ❷
         (: gnus-registry-split-fancy-with-parent) ❸
         ;; splitting rules go here         ❹
         "INBOX"                            ❺
        ))
 (setq nnimap-split-rule 'nnmail-split-fancy)
 (setq nnmail-split-methods 'nnimap-split-fancy) ❻
 (gnus-registry-initialize) ❼
```

This configures IMAP to use the `nnmail-split-fancy` function to determine the group for messages. Note that we're setting the `nnmail-split-fancy` variable here. If you want to process your IMAP mail separately from your other mail, you can set the `nnimap-split-fancy` variable instead. If so, also set `nnimap-split-rule` to `'nnimap-split-fancy`. Using `nnmail-split-fancy` here makes the other examples easier to understand, though.

The `nnmail-split-fancy` variable controls the splitting behavior❶. The | symbol means that that the first matching rule is used❷. For example, if the message being processed is a reply to a message that Gnus knows about, then the `gnus-registry-split-fancy-with-parent` function will return the name of the group, and `nnmail-split-fancy` will file the message there❸. You can add other splitting rules as well❹. If messages don't match any of these rules, the last rule specifies that the messages will be filed in INBOX❺. Set `nnmail-split-methods` to `nnimap-split-fancy` as well in order to work around some assumptions in other parts of the code❻. After that, initialize the Gnus registry❼, which is responsible for tracking moved and deleted messages. This allows you to automatically split replies into the same folders as the original messages.

To configure fancy mail splitting with an nnml backend (suggested configuration for POP3), add the following to your *~/.gnus* instead:

```
(gnus-registry-initialize)
(setq nnmail-split-fancy
      '(|
        (: gnus-registry-split-fancy-with-parent)
        ;; splitting rules go here
        "mail.misc"                           ;; ❶
       ))
(setq nnmail-split-methods 'nnmail-split-fancy)
```

This code is similar to the IMAP example, except that the default mailbox name for nnml is mail.misc❶.

Here's how the previous rules in `nnmail-split-methods` would be translated to `nnmail-split-fancy` rules for an IMAP configuration:

```
(setq nnmail-split-fancy
     '(|
       (: gnus-registry-split-fancy-with-parent)
        ;; splitting rules go here
       (from mail "INBOX.errors")   ;; ❶
       (any "you@work.example.com" "INBOX.work")    ❷
       (any "you@personal.example.com" "INBOX.personal") ;;
       ("subject" "emacs" "INBOX.emacs") ❸
       "INBOX"    ;; or "mail.misc" for nnml/POP3
      ))
```

The `from` keyword matches against the "From", "Sender", and "Resent-From" fields, while the mail keyword matches common mail system addresses❶. The corresponding `to` keyword matches against the "To", "Cc", "Apparently-To", "Resent-To" and "Resent-Cc" headers, while `any` matches the fields checked by the `from` and `to` keywords❷. You can also compare against the subject and other headers❸.

You can use logic in splitting rules, too. For example, if you like reading the jokes on joke-mailing-list@example.com, but you don't like the ones sent by vi-guy@example.com (he not only has a bad sense of humor, but also likes picking on Emacs!), you can use a rule like this in your `nnmail-split-fancy`:

```
        ;; ... other splitting rules go here...
        (any "joke-mailing-list@example.com"    ;; ❶
             (| (from "vi-guy@example.com" "INBOX.junk") ❷
                "INBOX.jokes")) ❸
        ;; ... other splitting rules go here
```

The first rule matches all messages with "joke-mailing-list@example.com" in from- or to-related headers. Matching messages are processed with another split rule, which moves messages from vi-guy@example.com to a separate group❷ and files the other messages in INBOX.jokes❸. To learn more about creating complex rules, read the Gnus Info manual for "Fancy Mail Splitting".

Project XXX: Filter Spam

Ah, spam, the bane of our Internet lives. There is no completely reliable way to automatically filter spam. Spam messages that slip through the filters and perfectly legitimate messages that get labeled spam are all part of the occupational hazards of using the Internet.

The fastest way to filter spam is to use an external spam-filtering program such as Spamassassin or Bogofilter, so your spam can be filtered in the background and you don't have to spend time in Emacs filtering it yourself. In an ideal world, this would be done on the mail server so that you don't even need to download unwanted messages. If your inbox isn't full of ads for medicine or stocks, your mail server is probably doing a decent job of filtering the mail for you.

As spam filtering isn't an exact science, you'll want to find out how you can check your spam folder for misclassified mail. If you download your mail through POP, find out if there's a webmail interface that will allow you to check if any real mail has slipped into the junk mail pile. If you're on IMAP, your mail server might automatically file spam messages in a different group. Here's how to add the spam group to your list of groups:

1. Type **M-x gnus** to bring up the group buffer.
2. Type **^** (`gnus-group-enter-server-mode`).
3. Choose the nnimap: entry for your mail server and press **RET** (`gnus-server-read-server`).
4. Find the spam or junk mail group if it exists.

5. Type **u** (`gnus-browse-unsubscribe-current-group`) to toggle the subscription. Subscribed groups will appear in your **M-x gnus** screen if they contain at least one unread message.

Another alternative is to have all the mail (spam and non-spam) delivered to your inbox, and then let Gnus be in charge of filing it into your spam and non-spam groups. If other people manage your mail server, ask them if you can have your mail processed by the spam filter but still delivered to your inbox. If you're administering your own mail server, set up a spam filtering system such as SpamAssassin or BogoFilter, then read the documentation of your spam filtering system to find out how to process the mail.

Spam filtering systems typically add a header such as "X-Spam-Status" or "X-Bogosity" to messages in order to indicate which messages are spam or even how spammy they are. To check if your mail server tags your messages as spam, open one of your messages in Gnus and type **C-u g** (gnus-summary-show-article) to view the complete headers and message. If you find a spam-related header such as X-Spam-Status, you can use it to split your mail. Add the following to your *~/.gnus*:

```
(setq spam-use-regex-headers t) ;; ❶
(setq spam-regex-headers-spam "^X-Spam-Status: Yes")    ❷
(require 'spam) ❸
(spam-initialize) ❹
```

This configures spam.el to detect spam based on message headers❶. Set `spam-regex-headers-spam` to a regular expression matching the header your mail server uses to indicate spam.❷ This configuration should be done before the spam.el library is loaded❸ and initialized❹, because *spam.el* uses the spam-use-* variables to determine which parts of the spam library to load.

In order to take advantage of this, you'll also need to add a rule that splits spam messages into a different group. If you haven't set up mail splitting yet, read the instructions on setting up fancy mail splitting in **Project XXX: Organize mail into groups**. Add `(: spam-split)` to either `nnmail-split-fancy` or `nnimap-split-fancy`, depending on your configuration. For example, your *~/.gnus* may look like this:

```
(setq nnmail-split-fancy
      '(
        ;; ... other split rules go here ...
        (: spam-split)
      ;; ... other split rules go here ...
        "mail.misc")) ; default mailbox
```

Project XXX: Score Mail

One of the most interesting features of Gnus is the way it can remember what you find interesting and what you find boring. When you're dealing with a high-volume mailing list or newsgroup, or even with a large inbox, it helps to have interesting messages bubble up to the top and boring messages sink to the bottom. Gnus can do that with a combination of manual and adaptive scoring.

Gnus can calculate message scores based on headers, keywords, arbitrary Lisp expressions, or combinations of all three. These rules can be permanent or temporary. Gnus can then perform different actions based on those message scores, such as sorting the message list by score, marking messages with scores higher or lower than a particular threshold, or even automatically deleting messages with a low score.

You can define these scoring rules manually or have Gnus generate them from your reading behavior. For example, Gnus could raise the score of message threads you've read or replied to, while lowering the score of message threads you marked as read without actually reading.

For a full discussion of scoring, read the "Scoring" section of the Gnus info manual. In this project, you'll learn how to create simple manual and adaptive scoring rules.

Configure Adaptive Scoring

Adaptive scoring creates scoring rules based on your reading behavior. For example, you could configure Gnus to automatically raise the score of messages you read, but lower the score of messages you kill with **C-k** (`gnus-summary-kill-same-subject`) or mark as read without actually reading by using **c** (`gnus-summary-catchup-and-exit`). To do so, add the following to your *~/.gnus*:

```
(setq gnus-use-adaptive-scoring t)
(setq gnus-default-adaptive-score-alist
     '((gnus-unread-mark)
       (gnus-ticked-mark (subject 5))
       (gnus-read-mark (subject 1))
       (gnus-killed-mark (subject -5))
       (gnus-catchup-mark (subject -1))))
```

Each element in the list represents a mark used in the Gnus summary buffer and whatever actions to take when the mark is found. The actions specify which header to match and how much to raise or lower the score. The headers you can use are:

You can then go about reading only the mail that looks interesting, typing **!** (`gnus-summary-tick-article-forward`) to keep track of particularly interesting messages, **C-k** (`gnus-summary-kill-same-subject`) whenever you read something uninteresting, and **c** (`gnus-summary-catchup-and-exit) to mark all other messages as read. Gnus will automatically highlight threads you've found interesting in the past.

You can modify multiple scores in one rule. For example, if you also want to change people's scores based on your reading behavior, you can change your scoring in *~/.gnus* to read:

```
(setq gnus-use-adaptive-scoring t)
(setq gnus-default-adaptive-score-alist
     '((gnus-unread-mark)
       (gnus-ticked-mark (from 5) (subject 5))
       (gnus-read-mark (from 1) (subject 1))
       (gnus-killed-mark (from -1) (subject -5))
       (gnus-catchup-mark (from -1) (subject -1))))
```

Your favorite authors and subjects will eventually float to the top. Messages with positive scores will automatically be highlighted and flagged with +. Similarly, messages with negative scores will be flagged with -. You can sort messages by score with **C-c C-s C-i** (gnus-summary-sort-by-score).

You might also be interested in automatically scoring up replies to your messages. To do so, add the following to your *~/.gnus*:

```
(add-hook 'message-sent-hook 'gnus-score-followup-article)
```

When you send a message in Gnus, Gnus will add a scoring rule that automatically raises the score of any message that's a direct reply to yours.

Create Scores by Example

You can create rules that lower or raise message scores in the current group from the Gnus message summary. To create a scoring rule based on the current message, type **I** (gnus-summary-increase-score) or **L** (gnus-summary-lower-score).

The first prompt determines which message attribute is used to calculate the score. Press ? for a list of options or use one from the following table of common options.

Option	Header	Notes
a	From (author)	Keep track of people you like or don't like to read. Use a substring match for the e-mail address.
s	Subject	Raise or lower this message thread. Use fuzzy matching for best results.
b	Body	Use substrings or regular expressions to match the body of the message. Slow.
l	Lines	Score messages that are longer or shorter than a threshold. This is good for scoring down really long messages. Slow.

Table 4: Options for scoring

The next prompt determines the match type. For strings like From: or Body:, you probably want either substring matches or regular expression matches. The Subject: match works well with fuzzy matching, which strips typical reply markers such as Re:.

The next prompt determines whether this rule is a permanent rule that will be saved to the group's score configuration file and used from now on, a temporary rule that will be saved in the group's score configuration file and deleted after a week of disuse (or `gnus-score-expiry-days` days), or a rule to be applied immediately and not saved.

Finally, you will be prompted for any other parameters the rule requires. This generally defaults to values from the current message, but can be edited.

By default, an interactively-created rule will increase or decrease the scores of matching messages by 1000 points. This means that interactively-defined rules will usually override adaptive scoring rules. You can modify this by using a numeric prefix when you call the scoring function. For example, **C-u 10 L** (`gnus-summary-lower-score`) will lower the score of matching messages by 10 points instead. You can also customize the value of gnus-score-interactive-default-score to make your system more or less sensitive to interactively defined rules.

Create Complex Scoring Rules

You can create even more complex scoring rules with a little bit of Emacs Lisp. For example, you might be generally interested in vi-guy@example.com writes, but you'd rather not have to read his long diatribes (more than 100 lines) about the evils of Emacs (or indeed, anything about Emacs). However, when he goes all-out and writes something over 400 lines, you often find the resulting rant quite amusing.

To edit the score file for the current group, type **V e** (`gnus-score-edit-current-scores`). The score file is a list of scoring rules. The format is somewhat arcane, so we'll go through vi-guy's example here.

Back to vi-guy. To score such messages down, you would use a score file that looks like this:

```
(("from" ("vi-guy@example.com" 1000)) ❶
 (& ("from" "vi-guy@example.com")      ❷
    ("body" "emacs")                   ❸
    (& ("lines" 100 >)                 ❹
       ("lines" 400 <))
   -2000))                             ❺
```

This sets up a scoring rule so that any messages from vi-guy@example.com are scored up by 1000 points❶. However, the second rule adjusts the score of messages from vi-guy@example.com❷ about Emacs❸ that are between 100 and 400 lines❹, lowering the score by 2000 points for a total of -1000 points❺. Note the difference in parentheses between the first "from" filter❶ and the second "from" filter❷: the first is a simple rule, the second is a complex one that's part of an &.

Type **C-c C-c** (`gnus-score-edit-exit`) to save and close the score file.

You can find even more contrived and complicated examples on the "Advanced Scoring Examples" page in the Gnus info manual. If you ever need this kind of complexity, hats off to you. Emacs being Emacs, it's all built in.

Project XXX: Set up Posting Styles

You might want to use your work address and signature when responding to mail sent to your work address, your personal address and signature when responding to regular mail, and a nospam address otherwise. This is handled by `gnus-posting-styles`, which you can set in *~/.gnus*. gnus-posting-style is a list of rules, and all matching rules will be applied in order.

The easiest way to use `gnus-posting-styles` is to set up groups (see **Project XXX: Organize Mail into Groups**) and then set your posting styles based on the groups. For example, if all your work mail is in INBOX.work and all your personal mail is in INBOX.personal, you can use the following code in your *~/.gnus*:

```
(setq gnus-posting-styles
      '((".*"    ; Matches all groups of messages
         (address "jh@nospam.example.com")
         (signature-file "~/.signature"))
        ("INBOX.work"
          (address "jh@work.example.com")
          (signature
"J. Random Hacker
Grand Poobah, Example Inc.
http://example.com"))
         ("INBOX.personal"
          (address "jh@example.com")
          (signature
"J. Random Hacker
http://example.com"))))
```

Even if you don't split your mail into different groups, you can still change posting styles depending on the message. For example, this uses the To: header:

```
 (add-to-list 'gnus-extra-headers 'To)
 (setq gnus-posting-styles
       '((".*"    ; Matches all groups of messages
          (address "jh@nospam.example.com")
          (signature-file "~/.signature"))
         ((header "to" "jh@work.example.com")
          (address "jh@work.example.com")
          (signature "J. Random Hacker
Grand Poobah, Example Inc.
http://example.com"))
         ((header "to" "jh@example.com")
          (address "jh@example.com")
          (signature "J. Random Hacker
http://example.com"))))
```

To learn more about what you can change, read the Posting Styles page of the Gnus info manual.