

# The Mutt E-Mail Client

by Michael Elkins <me@cs.hmc.edu>

version @VERSION@

“All mail clients suck. This one just sucks less.” -me, circa 1995

## Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>10</b> |
| 1.1      | Page d'accueil de Mutt . . . . .  | 10        |
| 1.2      | Listes de diffusion . . . . .   | 10        |
| 1.3      | Où trouver Mutt ? . . . . .   | 11        |
| 1.4      | IRC . . . . .   | 11        |
| 1.5      | USENET . . . . .  | 11        |
| 1.6      | Droits de reproduction et licence . . . . .                             | 11        |
| <b>2</b> | <b>Premiers pas</b>   | <b>11</b> |
| 2.1      | Déplacement dans les menus . . . . .                                    | 11        |
| 2.2      | Éditer une ligne . . . . .  | 12        |
| 2.3      | Lire du courrier - index et afficheur . . . . .                         | 12        |
| 2.3.1    | L'index des messages . . . . .  | 13        |
| 2.3.2    | Afficheur . . . . .   | 14        |
| 2.3.3    | Mode d'affichage en enfilades . . . . .                                 | 15        |
| 2.3.4    | Diverses autres fonctions . . . . .                                     | 16        |
| 2.4      | Envoyer du courrier . . . . .   | 17        |
| 2.4.1    | Éditer les en-têtes de courrier . . . . .                               | 18        |
| 2.4.2    | Utiliser Mutt avec PGP . . . . .  | 19        |
| 2.4.3    | Envoyer des messages anonymes par l'intermédiaire de mixmaster. . . . . | 20        |
| 2.5      | Faire suivre ou réexpédier un message . . . . .                         | 20        |
| 2.6      | Ajourner un courrier . . . . .  | 21        |
| <b>3</b> | <b>Configuration</b>  | <b>21</b> |
| 3.1      | Syntaxe des fichiers d'initialisation . . . . .                         | 21        |
| 3.2      | Définir/Utiliser des alias . . . . .                                    | 22        |
| 3.3      | Modifier les raccourcis clavier . . . . .                               | 23        |
| 3.4      | Définir des alias pour les jeux de caractères . . . . .                 | 24        |
| 3.5      | Positionner des variables selon la boîte aux lettres . . . . .          | 24        |
| 3.6      | Macros clavier . . . . .  | 25        |

---

|          |  |           |
|----------|--|-----------|
| 3.7      | Utiliser les attributs vidéos de couleur (color) et mono       | 25        |
| 3.8      | Ignorer (cacher) des en-têtes de messages                      | 27        |
| 3.9      | Listes de diffusion  | 28        |
| 3.10     | Utiliser plusieurs boîtes de type spoule                       | 28        |
| 3.11     | Définir les boîtes qui reçoivent du courrier                   | 29        |
| 3.12     | User defined headers   | 29        |
| 3.13     | Defining the order of headers when viewing messages            | 30        |
| 3.14     | Specify default save filename                                  | 30        |
| 3.15     | Specify default Fcc: mailbox when composing                    | 30        |
| 3.16     | Specify default save filename and default Fcc: mailbox at once | 30        |
| 3.17     | Change settings based upon message recipients                  | 30        |
| 3.18     | Change settings before formatting a message                    | 31        |
| 3.19     | Choosing the PGP key of the recipient                          | 31        |
| 3.20     | Adding key sequences to the keyboard buffer                    | 31        |
| 3.21     | Executing functions  | 31        |
| 3.22     | Message Scoring  | 32        |
| 3.23     | Setting variables  | 32        |
| 3.24     | Reading initialization commands from another file              | 33        |
| 3.25     | Removing hooks   | 33        |
| <b>4</b> | <b>Utilisation avancée</b>                                     | <b>33</b> |
| 4.1      | Expressions Rationnelles                                       | 33        |
| 4.2      | Motifs   | 36        |
| 4.2.1    | Modificateur de motifs   | 37        |
| 4.2.2    | Motifs Complexes   | 37        |
| 4.2.3    | Recherche par date   | 38        |
| 4.3      | Using Tags   | 39        |
| 4.4      | Using Hooks  | 39        |
| 4.4.1    | Message Matching in Hooks                                      | 40        |
| 4.5      | External Address Queries                                       | 40        |
| 4.6      | Mailbox Formats  | 41        |
| 4.7      | Mailbox Shortcuts  | 41        |
| 4.8      | Handling Mailing Lists   | 42        |
| 4.9      | Support de DSN (notification de statut d'acheminement)         | 43        |
| 4.10     | Support de POP3 (OPTIONNEL)                                    | 43        |
| 4.11     | Support d'IMAP (OPTIONNEL)                                     | 43        |
| 4.11.1   | Le sélecteur de dossiers                                       | 44        |

---

|          |  |           |
|----------|--|-----------|
| 4.11.2   | Authentification                                     | 44        |
| 4.12     | Gérer plusieurs comptes IMAP/POP (OPTIONNEL)         | 45        |
| 4.13     | Démarrer un navigateur web sur les URLs (EXTERNE)    | 45        |
| <b>5</b> | <b>Mutt's MIME Support</b>                           | <b>45</b> |
| 5.1      | Using MIME in Mutt                                   | 46        |
| 5.1.1    | Viewing MIME messages in the pager                   | 46        |
| 5.1.2    | The Attachment Menu                                  | 46        |
| 5.1.3    | The Compose Menu                                     | 46        |
| 5.2      | MIME Type configuration with <code>mime.types</code> | 47        |
| 5.3      | MIME Viewer configuration with <code>mailcap</code>  | 47        |
| 5.3.1    | The Basics of the mailcap file                       | 48        |
| 5.3.2    | Secure use of mailcap                                | 49        |
| 5.3.3    | Advanced mailcap Usage                               | 49        |
| 5.3.4    | Example mailcap files                                | 51        |
| 5.4      | MIME Autoview  | 52        |
| 5.5      | MIME Multipart/Alternative                           | 53        |
| <b>6</b> | <b>Référence</b>                                     | <b>53</b> |
| 6.1      | Options de la ligne de commande                      | 53        |
| 6.2      | Commandes de configuration                           | 54        |
| 6.3      | Configuration variables                              | 55        |
| 6.3.1    | <code>abort_nosubject</code>                         | 55        |
| 6.3.2    | <code>abort_unmodified</code>                        | 55        |
| 6.3.3    | <code>alias_file</code>                              | 55        |
| 6.3.4    | <code>alias_format</code>                            | 56        |
| 6.3.5    | <code>allow_8bit</code>                              | 56        |
| 6.3.6    | <code>allow_ansi</code>                              | 56        |
| 6.3.7    | <code>alternates</code>                              | 56        |
| 6.3.8    | <code>arrow_cursor</code>                            | 57        |
| 6.3.9    | <code>ascii_chars</code>                             | 57        |
| 6.3.10   | <code>askbcc</code>                                  | 57        |
| 6.3.11   | <code>askcc</code>                                   | 57        |
| 6.3.12   | <code>attach_format</code>                           | 57        |
| 6.3.13   | <code>attach_sep</code>                              | 58        |
| 6.3.14   | <code>attach_split</code>                            | 58        |
| 6.3.15   | <code>attribution</code>                             | 58        |
| 6.3.16   | <code>autoedit</code>                                | 59        |

---

|        |                   |    |
|--------|-------------------|----|
| 6.3.17 | auto_tag          | 59 |
| 6.3.18 | beep              | 59 |
| 6.3.19 | beep_new          | 59 |
| 6.3.20 | bounce_delivered  | 59 |
| 6.3.21 | charset           | 59 |
| 6.3.22 | check_new         | 59 |
| 6.3.23 | collapse_unread   | 60 |
| 6.3.24 | uncollapse_jump   | 60 |
| 6.3.25 | compose_format    | 60 |
| 6.3.26 | confirmappend     | 60 |
| 6.3.27 | confirmcreate     | 61 |
| 6.3.28 | connect_timeout   | 61 |
| 6.3.29 | copy              | 61 |
| 6.3.30 | date_format       | 61 |
| 6.3.31 | default_hook      | 61 |
| 6.3.32 | delete            | 62 |
| 6.3.33 | delete_untag      | 62 |
| 6.3.34 | digest_collapse   | 62 |
| 6.3.35 | display_filter    | 62 |
| 6.3.36 | dotlock_program   | 62 |
| 6.3.37 | dsn_notify        | 62 |
| 6.3.38 | dsn_return        | 63 |
| 6.3.39 | duplicate_threads | 63 |
| 6.3.40 | edit_headers      | 63 |
| 6.3.41 | editor            | 63 |
| 6.3.42 | encode_from       | 63 |
| 6.3.43 | envelope_from     | 64 |
| 6.3.44 | escape            | 64 |
| 6.3.45 | fast_reply        | 64 |
| 6.3.46 | fcc_attach        | 64 |
| 6.3.47 | fcc_clear         | 64 |
| 6.3.48 | folder            | 64 |
| 6.3.49 | folder_format     | 65 |
| 6.3.50 | followup_to       | 65 |
| 6.3.51 | force_name        | 66 |
| 6.3.52 | forward_decode    | 66 |
| 6.3.53 | forward_format    | 66 |

---

|        |                      |    |
|--------|----------------------|----|
| 6.3.54 | forward_quote        | 66 |
| 6.3.55 | from                 | 66 |
| 6.3.56 | gecos_mask           | 67 |
| 6.3.57 | hdrs                 | 67 |
| 6.3.58 | header               | 67 |
| 6.3.59 | help                 | 67 |
| 6.3.60 | hidden_host          | 67 |
| 6.3.61 | hide_limited         | 68 |
| 6.3.62 | hide_missing         | 68 |
| 6.3.63 | hide_top_limited     | 68 |
| 6.3.64 | hide_top_missing     | 68 |
| 6.3.65 | history              | 68 |
| 6.3.66 | honor_followup_to    | 68 |
| 6.3.67 | hostname             | 68 |
| 6.3.68 | ignore_list_reply_to | 69 |
| 6.3.69 | imap_authenticators  | 69 |
| 6.3.70 | imap_delim_chars     | 69 |
| 6.3.71 | imap_force_ssl       | 69 |
| 6.3.72 | imap_home_namespace  | 69 |
| 6.3.73 | imap_keepalive       | 70 |
| 6.3.74 | imap_list_subscribed | 70 |
| 6.3.75 | imap_pass            | 70 |
| 6.3.76 | imap_passive         | 70 |
| 6.3.77 | imap_peek            | 70 |
| 6.3.78 | imap_servernoise     | 70 |
| 6.3.79 | imap_user            | 71 |
| 6.3.80 | implicit_autoview    | 71 |
| 6.3.81 | include              | 71 |
| 6.3.82 | indent_string        | 71 |
| 6.3.83 | index_format         | 71 |
| 6.3.84 | ispell               | 74 |
| 6.3.85 | keep_flagged         | 74 |
| 6.3.86 | locale               | 74 |
| 6.3.87 | mail_check           | 74 |
| 6.3.88 | mailcap_path         | 74 |
| 6.3.89 | mailcap_sanitize     | 74 |
| 6.3.90 | maildir_trash        | 75 |

---

|         |                        |    |
|---------|------------------------|----|
| 6.3.91  | mark_old               | 75 |
| 6.3.92  | markers                | 75 |
| 6.3.93  | mask                   | 75 |
| 6.3.94  | mbox                   | 75 |
| 6.3.95  | mbox_type              | 75 |
| 6.3.96  | metoo                  | 76 |
| 6.3.97  | menu_scroll            | 76 |
| 6.3.98  | meta_key               | 76 |
| 6.3.99  | mh_purge               | 76 |
| 6.3.100 | mh_seq_flagged         | 76 |
| 6.3.101 | mh_seq_replied         | 76 |
| 6.3.102 | mh_seq_unseen          | 77 |
| 6.3.103 | mime_forward           | 77 |
| 6.3.104 | mime_forward_decode    | 77 |
| 6.3.105 | mime_forward_rest      | 77 |
| 6.3.106 | mix_entry_format       | 77 |
| 6.3.107 | mixmaster              | 78 |
| 6.3.108 | move                   | 78 |
| 6.3.109 | message_format         | 78 |
| 6.3.110 | pager                  | 78 |
| 6.3.111 | pager_context          | 78 |
| 6.3.112 | pager_format           | 79 |
| 6.3.113 | pager_index_lines      | 79 |
| 6.3.114 | pager_stop             | 79 |
| 6.3.115 | pgp_autosign           | 79 |
| 6.3.116 | pgp_autoencrypt        | 79 |
| 6.3.117 | pgp_ignore_subkeys     | 79 |
| 6.3.118 | pgp_entry_format       | 80 |
| 6.3.119 | pgp_good_sign          | 80 |
| 6.3.120 | pgp_long_ids           | 81 |
| 6.3.121 | pgp_replyencrypt       | 81 |
| 6.3.122 | pgp_replysign          | 81 |
| 6.3.123 | pgp_replysignencrypted | 81 |
| 6.3.124 | pgp_retainable_sigs    | 81 |
| 6.3.125 | pgp_show_unusable      | 81 |
| 6.3.126 | pgp_sign_as            | 82 |
| 6.3.127 | pgp_strict_enc         | 82 |

---

|         |                          |    |
|---------|--------------------------|----|
| 6.3.128 | pgp_timeout              | 82 |
| 6.3.129 | pgp_verify_sig           | 82 |
| 6.3.130 | pgp_sort_keys            | 82 |
| 6.3.131 | pgp_create_traditional   | 83 |
| 6.3.132 | pgp_decode_command       | 83 |
| 6.3.133 | pgp_getkeys_command      | 83 |
| 6.3.134 | pgp_verify_command       | 84 |
| 6.3.135 | pgp_decrypt_command      | 84 |
| 6.3.136 | pgp_clearsign_command    | 84 |
| 6.3.137 | pgp_sign_command         | 84 |
| 6.3.138 | pgp_encrypt_sign_command | 84 |
| 6.3.139 | pgp_encrypt_only_command | 84 |
| 6.3.140 | pgp_import_command       | 84 |
| 6.3.141 | pgp_export_command       | 85 |
| 6.3.142 | pgp_verify_key_command   | 85 |
| 6.3.143 | pgp_list_secring_command | 85 |
| 6.3.144 | pgp_list_pubring_command | 85 |
| 6.3.145 | forward_decrypt          | 85 |
| 6.3.146 | ssl_starttls             | 85 |
| 6.3.147 | certificate_file         | 86 |
| 6.3.148 | ssl_usesystemcerts       | 86 |
| 6.3.149 | entropy_file             | 86 |
| 6.3.150 | ssl_use_sslv2            | 86 |
| 6.3.151 | ssl_use_sslv3            | 86 |
| 6.3.152 | ssl_use_tlsv1            | 86 |
| 6.3.153 | pipe_split               | 86 |
| 6.3.154 | pipe_decode              | 87 |
| 6.3.155 | pipe_sep                 | 87 |
| 6.3.156 | pop_authenticators       | 87 |
| 6.3.157 | pop_auth_try_all         | 87 |
| 6.3.158 | pop_checkinterval        | 87 |
| 6.3.159 | pop_delete               | 88 |
| 6.3.160 | pop_host                 | 88 |
| 6.3.161 | pop_last                 | 88 |
| 6.3.162 | pop_reconnect            | 88 |
| 6.3.163 | pop_user                 | 88 |
| 6.3.164 | pop_pass                 | 88 |

---

|  |    |
|--|----|
| 6.3.165 post_indent_string . . . . .     | 89 |
| 6.3.166 postpone . . . . .               | 89 |
| 6.3.167 postponed . . . . .              | 89 |
| 6.3.168 preconnect . . . . .             | 89 |
| 6.3.169 print . . . . .                  | 89 |
| 6.3.170 print_command . . . . .          | 90 |
| 6.3.171 print_decode . . . . .           | 90 |
| 6.3.172 print_split . . . . .            | 90 |
| 6.3.173 prompt_after . . . . .           | 90 |
| 6.3.174 query_command . . . . .          | 90 |
| 6.3.175 quit . . . . .                   | 90 |
| 6.3.176 quote_regexp . . . . .           | 91 |
| 6.3.177 read_inc . . . . .               | 91 |
| 6.3.178 read_only . . . . .              | 91 |
| 6.3.179 realname . . . . .               | 91 |
| 6.3.180 recall . . . . .                 | 91 |
| 6.3.181 record . . . . .                 | 92 |
| 6.3.182 reply_regexp . . . . .           | 92 |
| 6.3.183 reply_self . . . . .             | 92 |
| 6.3.184 reply_to . . . . .               | 92 |
| 6.3.185 resolve . . . . .                | 92 |
| 6.3.186 reverse_alias . . . . .          | 92 |
| 6.3.187 reverse_name . . . . .           | 93 |
| 6.3.188 reverse_realname . . . . .       | 93 |
| 6.3.189 rfc2047_parameters . . . . .     | 93 |
| 6.3.190 save_address . . . . .           | 93 |
| 6.3.191 save_empty . . . . .             | 94 |
| 6.3.192 save_name . . . . .              | 94 |
| 6.3.193 score . . . . .                  | 94 |
| 6.3.194 score_threshold_delete . . . . . | 94 |
| 6.3.195 score_threshold_flag . . . . .   | 94 |
| 6.3.196 score_threshold_read . . . . .   | 95 |
| 6.3.197 send_charset . . . . .           | 95 |
| 6.3.198 sendmail . . . . .               | 95 |
| 6.3.199 sendmail_wait . . . . .          | 95 |
| 6.3.200 shell . . . . .                  | 96 |
| 6.3.201 sig_dashes . . . . .             | 96 |

---

|                                   |     |
|-----------------------------------|-----|
| 6.3.202 sig_on_top . . . . .      | 96  |
| 6.3.203 signature . . . . .       | 96  |
| 6.3.204 simple_search . . . . .   | 96  |
| 6.3.205 smart_wrap . . . . .      | 96  |
| 6.3.206 smileys . . . . .         | 97  |
| 6.3.207 sleep_time . . . . .      | 97  |
| 6.3.208 sort . . . . .            | 97  |
| 6.3.209 sort_alias . . . . .      | 97  |
| 6.3.210 sort_aux . . . . .        | 98  |
| 6.3.211 sort_browser . . . . .    | 98  |
| 6.3.212 sort_re . . . . .         | 98  |
| 6.3.213 spoolfile . . . . .       | 98  |
| 6.3.214 status_chars . . . . .    | 99  |
| 6.3.215 status_format . . . . .   | 99  |
| 6.3.216 status_on_top . . . . .   | 101 |
| 6.3.217 strict_threads . . . . .  | 101 |
| 6.3.218 suspend . . . . .         | 101 |
| 6.3.219 text_flowed . . . . .     | 101 |
| 6.3.220 thread_received . . . . . | 101 |
| 6.3.221 thorough_search . . . . . | 102 |
| 6.3.222 tilde . . . . .           | 102 |
| 6.3.223 timeout . . . . .         | 102 |
| 6.3.224 tmpdir . . . . .          | 102 |
| 6.3.225 to_chars . . . . .        | 102 |
| 6.3.226 tunnel . . . . .          | 102 |
| 6.3.227 use_8bitmime . . . . .    | 103 |
| 6.3.228 use_domain . . . . .      | 103 |
| 6.3.229 use_from . . . . .        | 103 |
| 6.3.230 use_ipv6 . . . . .        | 103 |
| 6.3.231 user_agent . . . . .      | 103 |
| 6.3.232 visual . . . . .          | 104 |
| 6.3.233 wait_key . . . . .        | 104 |
| 6.3.234 weed . . . . .            | 104 |
| 6.3.235 wrap_search . . . . .     | 104 |
| 6.3.236 wrapmargin . . . . .      | 104 |
| 6.3.237 write_inc . . . . .       | 104 |
| 6.3.238 write_bcc . . . . .       | 105 |

|   |            |
|---|------------|
| 6.4 Fonctions . . . . .                       | 105        |
| 6.4.1 generic . . . . .                       | 105        |
| 6.4.2 index . . . . .                         | 106        |
| 6.4.3 pager . . . . .                         | 107        |
| 6.4.4 alias . . . . .                         | 108        |
| 6.4.5 query . . . . .                         | 109        |
| 6.4.6 attach . . . . .                        | 109        |
| 6.4.7 compose . . . . .                       | 109        |
| 6.4.8 postpone . . . . .                      | 110        |
| 6.4.9 browser . . . . .                       | 110        |
| 6.4.10 pgp . . . . .                          | 110        |
| 6.4.11 editor . . . . .                       | 110        |
| <b>7 Divers</b>                               | <b>111</b> |
| 7.1 Remerciements . . . . .                   | 111        |
| 7.2 À propos de cette documentation . . . . . | 112        |

## 1 Introduction

Petit mais très puissant, **Mutt** est un client pour la lecture de courrier en mode texte. Il supporte notamment les messages en MIME. Mutt est très paramétrable et s'adapte parfaitement aux utilisateurs chevronnés par ses nombreux raccourcis, ses macros clavier, l'affichage en enfilade des courriers, les recherches par expressions rationnelles, et par sa fonction de sélection de groupes de messages par motifs (des expressions désignant des en-têtes ou des sujets par exemple).

### 1.1 Page d'accueil de Mutt

<http://www.mutt.org/>

### 1.2 Listes de diffusion

Pour s'inscrire à une des listes de diffusion suivantes, envoyez un message contenant le mot *subscribe* dans le corps du message à `nom-de-la-liste-request@mutt.org`.

- [mutt-announce@mutt.org](mailto:mutt-announce@mutt.org) – liste à faible trafic pour les annonces
- [mutt-users@mutt.org](mailto:mutt-users@mutt.org) – aide, rapports de bug, demande de fonctionnalités
- [mutt-dev@mutt.org](mailto:mutt-dev@mutt.org) – liste pour les développeurs de mutt

**Remarque :** tous les messages de *mutt-announce* sont transférés à *mutt-users*, il n'est pas nécessaire de s'abonner aux deux. listes.

### 1.3 Où trouver Mutt ?

- <ftp://ftp.mutt.org/pub/mutt/>
- <ftp://ftp.guug.de/pub/mutt/>

Une liste des sites miroir est disponible à l'adresse <http://www.mutt.org/download.html> .

### 1.4 IRC

Venez sur le canal `#mutt` sur [OpenProjects.Net \(www.openprojects.net\)](http://www.openprojects.net) pour discuter avec d'autres personnes intéressées par Mutt.

### 1.5 USENET

Vous pouvez consulter le groupe de discussion [comp.mail.mutt](mailto:comp.mail.mutt) .

### 1.6 Droits de reproduction et licence

Copyright (C) 1996-2000 Michael R. Elkins <me@cs.hmc.edu> ainsi que d'autres personnes.

Ce programme est un logiciel libre : vous pouvez le redistribuer et / ou le modifier selon les termes de la "GNU General Public License", tels que publiés par la "Free Software Foundation"; soit la version 2 de cette licence ou (à votre choix) toute version ultérieure.

Ce programme est distribué dans l'espoir qu'il sera utile, mais SANS AUCUNE GARANTIE, ni explicite ni implicite; sans même les garanties de commercialisation ou d'adaptation dans un but spécifique.

Se référer à la "GNU General Public License" pour plus de détails.

Vous devriez avoir reçu une copie de la "GNU General Public License" en même temps que ce programme; sinon, écrivez à la "Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA".

## 2 Premiers pas

Cette section est un petit tour du propriétaire de Mutt. Beaucoup de fonctionnalités de Mutt sont décrites dans d'autres sections du manuel, dans la FAQ, ou bien sur diverses pages web. La page web de [Mutt](#) vous donnera plus de détails à ce propos.

Les raccourcis clavier listés ici sont ceux définis par défaut dans Mutt. Votre administrateur système peut avoir modifié la configuration par défaut pour votre site. Dans tous les menus, vous pouvez toujours taper "?" pour afficher les raccourcis clavier actuellement actifs.

La première chose à faire est de lancer Mutt depuis la ligne de commande en tapant tout simplement "mutt". Il y a diverses options pour la ligne de commande, elles sont décrites dans la page de manuel ou bien dans la [6.1](#) (référence).

### 2.1 Déplacement dans les menus

Les informations sont présentées sous forme de menus, similairement à ELM (Electronic Mail for UNIX). Voici une table des touches de navigation dans les menus de Mutt.

|                    |                |                                       |
|--------------------|----------------|---------------------------------------|
| j ou <Flèche Bas>  | next-entry     | élément suivant                       |
| k ou <Flèche Haut> | previous-entry | élément précédent                     |
| z ou <Page Suiv.>  | page-down      | page suivante                         |
| Z ou <Page Préc.>  | page-up        | page précédente                       |
| = ou <Début>       | first-entry    | aller au premier élément              |
| * ou <Fin>         | last-entry     | aller au dernier élément              |
| q                  | quit           | quitter le menu courant               |
| ?                  | help           | lister les raccourcis claviers actifs |

## 2.2 Éditer une ligne

Mutt a un éditeur de ligne intégré qui est utilisé pour entrer des données de type texte comme des adresses email ou des noms de fichiers. Les touches utilisées pour se déplacer pendant l'édition d'une ligne sont similaires à celles d'Emacs.

|                       |                 |   |
|-----------------------|-----------------|---|
| ^A ou <Début>         | bol             | aller en début de ligne                 |
| ^B ou <Flèche gauche> | backward-char   | aller un caractère à gauche             |
| Esc B                 | backward-word   | aller un mot à gauche                   |
| ^D ou <Suppr.>        | delete-char     | supprimer le caractère sous le curseur  |
| ^E ou <Fin>           | eol             | aller en fin de ligne                   |
| ^F ou <Right>         | forward-char    | aller un caractère à droite             |
| Esc F                 | forward-word    | aller un mot à droite                   |
| <Tab>                 | complete        | compléter un nom de fichier ou alias    |
| ^T                    | complete-query  | compléter une adresse avec requête      |
| ^K                    | kill-eol        | supprimer jusqu'à la fin de la ligne    |
| ESC d                 | kill-eow        | supprimer jusqu'à la fin du mot         |
| ^W                    | kill-word       | supprimer le mot avant le curseur       |
| ^U                    | kill-line       | supprimer la ligne                      |
| ^V                    | quote-char      | insérer littéralement un caractère      |
| <Up>                  | history-up      | ligne précédente dans l'historique      |
| <Down>                | history-down    | ligne suivante dans l'historique        |
| <BackSpace>           | backspace       | supprimer le caractère avant le curseur |
| Esc u                 | upcase-word     | mettre le mot en majuscules             |
| Esc l                 | downcase-word   | mettre le mot en minuscules             |
| Esc c                 | capitalize-word | mettre le mot en capitales              |
| ^G                    | n/a             | annuler                                 |
| <Return>              | n/a             | terminer l'édition                      |

Vous pouvez réassigner les fonctions de l'*éditeur* en utilisant la commande [3.3](#) (bind). Par exemple pour que la touche Suppression (Delete en anglais) supprime le caractère avant le curseur plutôt que celui en dessous, vous pourriez utiliser :

```
bind editor <delete> backspace
```

## 2.3 Lire du courrier - index et afficheur

Comme beaucoup d'autres logiciels de messagerie, il y a deux modes de lecture du courrier dans Mutt. Le premier est l'index des messages dans une boîte aux lettres, c'est ce qui est appelé "index" dans Mutt. Le second mode, nommé "pager" (afficheur), correspond à l'affichage du contenu d'un message.

Les prochaines sections décrivent les fonctions de chacun de ces modes.

## 2.3.1 L'index des messages

|          |  |
|----------|--|
| c        | changer de boîte aux lettres                                   |
| ESC c    | changer de dossier et passer en mode lecture seule             |
| C        | copier le message courant dans une autre boîte aux lettres     |
| ESC C    | décoder un message et le copier dans un dossier                |
| ESC s    | décoder un message et le sauvegarder dans un dossier           |
| D        | supprimer les messages suivant une expression rationnelle      |
| d        | supprimer le message courant                                   |
| F        | marquer comme 'important'                                      |
| l        | afficher les messages suivant une expression rationnelle       |
| N        | marquer le message comme 'nouveau'                             |
| o        | changer l'ordre de tri actuel                                  |
| O        | inverser l'ordre de tri de la boîte aux lettres                |
| q        | sauvegarder les modifications et quitter                       |
| s        | sauvegarder le message   |
| T        | marquer les messages suivant une expression rationnelle        |
| t        | ajouter / supprimer une marque sur un message                  |
| ESC t    | ajouter / supprimer une marque sur les messages d'une enfilade |
| U        | récupérer des messages suivant une expression rationnelle      |
| u        | récupérer un message   |
| v        | afficher les pièces jointes                                    |
| x        | annuler les modifications et quitter                           |
| <Entrée> | afficher le message  |
| <Tab>    | aller au prochain nouveau message                              |
| @        | afficher l'adresse complète de l'auteur                        |
| \$       | sauvegarder les changements de la boîte aux lettres courante   |
| /        | rechercher   |
| ESC /    | rechercher en sens inverse                                     |
| ^L       | redessine l'écran  |
| ^T       | supprimer des marques suivant une expression rationnelle       |

**Indicateurs d'état** En plus de l'expéditeur du message et de son sujet, un court résumé de l'état de chaque message est affiché à côté du numéro du message. Aucun ou plusieurs des indicateurs suivants peuvent être affichés :

**D**

le message est supprimé (marqué pour la suppression)

**d**

le message a des pièces jointes marquées pour la suppression

**K**

contient une clé PGP publique

**N**

nouveau message

**O**

ancien message

**P**

le message est chiffré avec PGP

r  
une réponse a été envoyée pour ce message

S  
le message est signé par PGP et la signature a été vérifiée

s  
le message est signé par PGP

!  
le message a l'indicateur 'important' de positionné

\*  
le message est marqué

Certains des indicateurs peuvent être positionnés ou retirés en utilisant :

- **set-flag** (w par défaut)
- **clear-flag** (W par défaut)

En outre, les indicateurs suivants reflètent le destinataire du message. Ils peuvent être personnalisés à l'aide de la variable :

#### 6.3.225 (\$to\_chars)

|   |  |
|---|--|
| + | vous êtes l'unique destinataire du message                           |
| T | vous êtes destinataire et d'autres personnes ont été placées en 'Cc' |
| C | une copie carbone du message vous a été adressée                     |
| F | vous êtes l'auteur du message  |
| L | une de vos listes de diffusion est destinataire du message           |

### 2.3.2 Afficheur

Mutt utilise par défaut son afficheur intégré pour afficher le corps des messages. L'afficheur est comparable au programme Unix nommé *less* même s'il ne dispose pas de toutes ses fonctionnalités.

|          |  |
|----------|--|
| <Entrée> | descendre d'une ligne  |
| <Espace> | page suivante (ou message suivant si on est en fin de message) |
| -        | page précédente  |
| n        | poursuivre la recherche vers l'avant                           |
| S        | sauter le texte cité (les lignes commençant par "< ")          |
| T        | afficher / masquer le texte cité                               |
| ?        | lister les raccourcis clavier                                  |
| /        | rechercher une expression rationnelle (un motif)               |
| ESC /    | rechercher en sens inverse                                     |
| \        | activer / désactiver la coloration du texte recherché          |
| ^        | aller au début du message                                      |
| \$       | aller à la fin du message                                      |

De plus, de nombreuses fonctionnalités de l'*index* sont disponibles depuis l'afficheur, par exemple *delete-message* (pour supprimer un message) ou bien *copy-message* (pour les copies). C'est un avantage par rapport à l'utilisation d'un autre afficheur que celui intégré à Mutt.

L'afficheur intégré supporte également quelques fonctionnalités avancées. Ainsi, il est capable de traduire les séquences nroff "standard" pour le gras et le souligné. Ces séquences sont des séries de caractères contenant soit la lettre, la touche d'effacement arrière (backspace, `^H`), et de nouveau la lettre pour le gras ou bien la lettre, retour arrière, `"_"` pour exprimer le soulignement. Mutt essaiera d'afficher le texte concerné respectivement en gras ou en souligné, si votre terminal le permet. Dans le cas contraire, vous pouvez définir les 3.7 (couleurs) `"bold"` et `"underline"` pour spécifier une certaine couleur ou l'attribut `"mono"` pour elles.

De même, l'afficheur intégré supporte les séquences d'échappement ANSI pour les attributs de caractères. Mutt les traduit pour qu'ils apparaissent correctement. Les séquences supportées sont :

```
ESC [ Ps;Ps;Ps;...;Ps m
avec Ps =
0 Désactive tous les attributs
1 Active le gras
4 Active le soulignement
5 Active le clignotement
7 Active la vidéo inverse
3x Mettre la couleur du texte à x
4x Mettre la couleur de fond à x
```

Les couleurs sont données par :

```
0 noir
1 rouge
2 vert
3 jaune
4 bleu
5 magenta
6 cyan
7 blanc
```

Mutt utilise ces attributs pour gérer les messages text/enriched (type correspondant au texte enrichi), ils peuvent aussi être utilisés pour la mise en surbrillance dans un script de 5.4 (visualisation automatique) externe. **Note :** Si vous changez les couleurs de votre affichage, par exemple en modifiant la couleur associée à `"color2"` dans votre xterm, c'est cette couleur qui sera utilisée à la place du vert.

### 2.3.3 Mode d'affichage en enfilades

Quelques nouvelles fonctions font leur apparition lorsque la boîte aux lettres est 6.3.208 (triée) par *enfilades* (threads) pour les modes d'*index* et d'*afficheur*.

|                    |                                 |  |
|--------------------|---------------------------------|--|
| <code>^D</code>    | <code>delete-thread</code>      | supprime les messages de l'enfilade          |
| <code>^U</code>    | <code>undelete-thread</code>    | récupère les messages de l'enfilade          |
| <code>^N</code>    | <code>next-thread</code>        | aller au début de la prochaine enfilade      |
| <code>^P</code>    | <code>previous-thread</code>    | aller au début de la précédente enfilade     |
| <code>^R</code>    | <code>read-thread</code>        | marquer l'enfilade comme lue                 |
| <code>ESC d</code> | <code>delete-subthread</code>   | supprimer la sous-branche de l'enfilade      |
| <code>ESC u</code> | <code>undelete-subthread</code> | récupérer la sous-branche de l'enfilade      |
| <code>ESC n</code> | <code>next-subthread</code>     | aller au début de la prochaine sous-branche  |
| <code>ESC p</code> | <code>previous-subthread</code> | aller au début de la précédente sous-branche |
| <code>ESC r</code> | <code>read-subthread</code>     | marquer la sous-branche comme lue            |
| <code>ESC t</code> | <code>tag-thread</code>         | marquer l'enfilade                           |
| <code>ESC v</code> | <code>collapse-thread</code>    | cacher / afficher l'enfilade                 |
| <code>ESC V</code> | <code>collapse-all</code>       | cacher / afficher toutes les enfilades       |
| <code>P</code>     | <code>parent-message</code>     | aller au message parent dans l'enfilade      |

**Note :** Si vous cachez une enfilade, seul le premier des messages de l'enfilade sera affiché. Cela est particulièrement utile lorsque les enfilades contiennent tellement de messages que vous ne pouvez plus voir que quelques enfilades à l'écran. Voir %M dans 6.3.83 (\$index\_format). Par exemple, vous pourriez utiliser "%?M?(#%03M)&(%4l)?" dans 6.3.83 (\$index\_format) pour afficher le nombre de messages si l'enfilade est cachée.

Voir aussi : 6.3.217 (\$strict\_threads).

### 2.3.4 Diverses autres fonctions

**create-alias** (par défaut : a)

Crée un nouvel alias basé sur le message en cours (ou bien propose la création d'un nouvel alias). Dès que l'édition en est terminée, une commande 3.2 (alias) est ajoutée au fichier spécifié par la variable 6.3.3 (\$alias\_file) pour une utilisation future. **Note :** cela ne rend pas pour autant l'alias disponible au prochain lancement de Mutt, il faut aussi 3.24 (sourcer) ce fichier.

**check-traditional-gpg** (par défaut : ESC P)

Cette fonction recherche des données signées ou codées de manière "traditionnelle" avec PGP, c'est à dire sans séparateur MIME, dans le message en cours. Techniquement parlant, cette fonction va temporairement changer le type MIME du corps du message contenant les données PGP ; ce comportement est similaire à celui de la fonction 2.3.4 (edit-type).

**display-toggle-weed** (par défaut : h)

Permet de masquer ou non les en-têtes des messages correspondant aux commandes 3.8 (ignore).

**edit** (par défaut : e)

Cette commande (disponible dans les modes "index" et "pager") permet d'éditer le message brut tel qu'il est stocké dans la boîte aux lettres. À la fin de l'édition, le message modifié sera ajouté au dossier courant, et le message d'origine sera marqué pour la suppression.

**edit-type**

(par default: ^E dans les menus des pièces jointes, de l'afficheur et de l'index ; ^T dans le menu d'édition de message)

Cette commande est utilisée pour éditer temporairement le type d'une pièce jointe pour, par exemple, corriger un mauvais jeu de caractères. Lorsqu'elle est invoquée depuis l'index ou l'afficheur, vous avez la possibilité d'éditer le type "racine" de la pièce jointe. Dans le 5.1.2 (menu attach) (permettant d'attacher des pièces jointes au message que vous allez envoyer), vous pouvez changer le type de n'importe quelle pièce jointe. Ces modifications ne sont pas permanentes et disparaissent après un changement de répertoire.

Cette commande est aussi disponible dans le 5.1.3 (menu compose) pour définir les propriétés des pièces jointes que vous allez envoyer.

**enter-command** (par défaut : ".")

Cette commande est utilisée pour exécuter n'importe quelle commande que vous mettriez normalement dans un fichier de configuration. Une de ses utilisations courante est de vérifier l'état d'une variable, ou bien de changer les préférences à la volée à l'aide de 3.6 (macros).

**extract-keys** (par défaut: ^K)

Cette commande extrait la ou les clé(s) publique(s) PGP du message courant ou des messages marqués pour les ajouter à votre anneau de clé publiques.

**forget-passphrase** (par défaut : ^F)

Cette commande efface la copie conservée en mémoire de la phrase de passe PGP. Elle est pratique lorsque vous avez entré une phrase erronée.

**list-reply** (par défaut : L)

Répondre au message courant ou aux messages marqués en prenant toutes les adresses qui sont listées dans les commandes 3.9 (lists ou subscribe), tout en respectant les champs de suivi (followup) si la variable 6.3.66 (\$honor\_followup\_to) est définie.

**pipe-message** (par défaut: |)

Exécute la commande Unix entrée par l'utilisateur en lui passant le contenu du message courant ou des messages marqués en entrée. Les variables 6.3.154 (\$pipe\_decode), 6.3.153 (\$pipe\_split), 6.3.155 (\$pipe\_sep), et 6.3.233 (\$wait\_key) définissent le comportement de cette fonction.

**resend-message** (par défaut : ESC e)

Avec resend-message, Mutt prend le message en cours comme modèle pour la création d'un nouveau message. Cette commande peut être utilisée pour faire suivre des messages MIME tout en conservant la structure d'origine du message. La quantité d'en-têtes inclus dépend de la valeur de la variable 6.3.234 (\$weed).

Cette fonction est aussi accessible depuis le menu des attachements. Elle permet de réexpédier un message qui avait été renvoyé (avec "bounce" par exemple) en tant que message/rfc822.

**shell-escape** (par défaut : !)

Exécute la commande Unix entrée par l'utilisateur. 6.3.233 (\$wait\_key) peut être utilisée pour contrôler si Mutt doit ou non attendre l'appui d'une touche après l'exécution de la commande (pour laisser à l'utilisateur le temps de lire un message, par exemple) en fonction du code de retour de la commande.

**toggle-quoted** (par défaut : T)

L'*afficheur* utilise la variable 6.3.176 (\$quote\_regexp) pour détecter le texte cité dans le corps du message. Cette fonction cache ou affiche le texte cité dans le message. Elle est particulièrement utile lorsque l'on est uniquement intéressé par la réponse et qu'il y a beaucoup de texte cité dans le message.

**skip-quoted** (par défaut : S)

Cette fonction va jusqu'à la prochaine ligne non-citée qui suit une ligne citée dans l'afficheur.

## 2.4 Envoyer du courrier

Dans le menu *index*, les raccourcis clavier suivants sont disponibles pour l'envoi de courrier.

m          compose          composer un nouveau message

|       |             |  |
|-------|-------------|--|
| r     | reply       | répondre à l'expéditeur                    |
| g     | group-reply | répondre à tous les destinataires          |
| L     | list-reply  | répondre à l'adresse de liste de diffusion |
| f     | forward     | faire suivre un message                    |
| b     | bounce      | réexpédier un message                      |
| ESC k | mail-key    | envoyer une clé publique PGP à quelqu'un   |

Renvoyer un message (“bounce”) le réexpédie tel quel au destinataire spécifié. Faire suivre un message vous permet d’y ajouter des commentaires ou de le modifier. Ces sujets sont abordés plus précisément dans le chapitre suivant 2.5 (“Forwarding and Bouncing Mail”).

Mutt vous présentera alors le menu *compose* et vous demandera d’entrer les destinataires à placer dans le champ d’en-têtes “To:”. Ensuite, il vous demandera l’objet (champ “Subject:”) du message ou le positionnera automatiquement si vous répondez à ou faites suivre un message. Voir aussi 6.3.11 (\$askcc), 6.3.10 (\$askbcc), 6.3.16 (\$autoedit), and 6.3.45 (\$fast\_reply) pour modifier la manière dont Mutt vous pose ces questions.

Mutt lancera alors votre éditeur de texte (6.3.41 (\$editor)) pour composer le corps du message. Si la variable 6.3.40 (\$edit\_headers) est positionnée, les en-têtes apparaîtront au dessus du corps dans votre éditeur. Chaque courrier auquel vous répondez est ajouté au message dans l’ordre de tri, avec les chaînes 6.3.15 (\$attribution), 6.3.82 (\$indent\_string) et 6.3.165 (\$post\_indent\_string) convenables. Lorsque vous faites suivre un message, si la variable 6.3.103 (\$mime\_forward) n’est pas positionnée, une copie du message à faire suivre sera incluse. Si vous avez spécifié une signature (6.3.203 (\$signature)), elle sera ajoutée à votre courrier.

Quand vous avez fini d’éditer le corps de votre message, Mutt revient au menu *compose*. Les options suivantes sont disponibles :

|       |                   |   |
|-------|-------------------|---|
| a     | attach-file       | attacher un fichier   |
| A     | attach-message    | attacher un (des) message(s) à ce courrier                      |
| ESC k | attach-key        | attacher une clé publique PGP                                   |
| d     | edit-description  | éditer la description de l’attachement                          |
| D     | detach-file       | effacer l’entrée courante                                       |
| t     | edit-to           | éditer le champ d’en-têtes To (destinataires)                   |
| ESC f | edit-from         | éditer le champ d’en-têtes From (auteur)                        |
| r     | edit-reply-to     | éditer le champ d’en-têtes Reply-To (répondre à)                |
| c     | edit-cc           | éditer le champ d’en-têtes Cc (copie carbone)                   |
| b     | edit-bcc          | éditer le champ d’en-têtes Bcc (copie aveugle)                  |
| y     | send-message      | envoyer le message  |
| s     | edit-subject      | éditer l’objet du message (champ Subject)                       |
| f     | edit-fcc          | spécifier une boîte aux lettres ‘Fcc’                           |
| p     | pgp-menu          | choisir les options PGP   |
| P     | postpone-message  | ajourner l’envoi de ce message                                  |
| q     | quit              | annuler (quitter) l’envoi de ce message                         |
| w     | write-fcc         | écrire ce message dans une boîte aux lettres                    |
| i     | ispell            | vérification l’orthographique (si disponible sur votre système) |
| ^F    | forget-passphrase | effacer la phrase de passe PGP de la mémoire                    |

**Remarque :** la fonction *attach-message* vous demandera de choisir un dossier d’où attacher les messages. Vous pouvez alors marquer (“tag”) les messages de cette boîte que vous souhaitez attacher à votre courrier. Notez que certaines opérations comme composer un nouveau message, répondre, faire suivre, etc. ne sont pas autorisées à ce moment. La séquence de format %r dans 6.3.215 (\$status\_format) prendra la valeur ‘A’ pour indiquer que vous vous trouvez dans le mode d’attachement de message.

### 2.4.1 Éditer les en-têtes de courrier

Lorsque de l’éditing des en-têtes de vos messages sortant, quelques fonctionnalités spéciales sont disponibles.

Si vous spécifiez

`Fcc: fichier`

Mutt utilisera *fichier* comme si vous aviez utilisé la fonction *edit-fcc* dans le menu *compose*.

Vous pouvez également attacher des fichiers à votre message en spécifiant

`Attach: fichier [ description ]`

où *fichier* est le fichier à attacher et *description* est une chaîne de caractère optionnelle donnant la description du fichier attaché.

Lorsque vous répondez à des messages, si vous supprimez le champ d'en-têtes *In-Reply-To:* (en réponse à), Mutt ne générera pas de champ *References:*, ce qui vous permet de créer une nouvelle enfilade.

Voir également [6.3.40](#) (`edit_headers`).

### 2.4.2 Utiliser Mutt avec PGP

Si vous souhaitez utiliser PGP, vous pouvez spécifier

`Pgp: [ E | S | S<id> ]`

“E” chiffre, “S” signe et “S<id>” signe à l’aide de la clé donnée, positionnant [6.3.126](#) (`$pgp_sign_as`) de manière permanente.

Si vous avez demandé à Mutt de chiffrer un message, il vous guidera dans les étapes de sélection de clé lorsque vous tenterez d’envoyer le courrier. Mutt ne vous demandera rien à propos des clés possédant un identifiant utilisateur (ID) correspondant à l’une des adresses de destinataires du message. Cependant peuvent apparaître des situations dans lesquelles il existe plusieurs clés, des champs d’identifiant utilisateur sont faiblement certifiés, ou aucune clé ne correspond.

Dans de tels cas, Mutt affiche un menu présentant une liste de clés que vous pouvez sélectionner. Lorsque vous quittez ce menu, ou si Mutt ne parvient pas à trouver une clé adéquate, il vous sera demandé un identifiant utilisateur. Comme d’habitude, il est possible d’annuler cette invite à l’aide de `^G`. Mutt retournera alors au menu *compose*.

Dès que la sélection de clé est achevée avec succès, le message est chiffré à l’aide de la clé choisie puis envoyé.

La plupart des champs des entrées du menu de sélection de clé (voir également [6.3.118](#) (`$pgp_entry_format`)) ont une signification évidente. Mais quelques explications sur les possibilités, drapeaux, et champ de validité sont nécessaires.

La séquence drapeaux (`%f`) est remplacée par l’un des drapeaux suivant :

|   |   |
|---|---|
| R | La clé a été révoquée et ne peut pas être utilisée.     |
| X | La clé a expiré et ne peut pas être utilisée.           |
| d | Vous avez déclaré que cette clé devait être désactivée. |
| c | Il y a des paquets d’auto-signature inconnus.           |

Le champ de possibilités (`%c`) est remplacé par une séquence de deux caractères représentant les possibilités d’une clé. Le premier caractère représente les possibilités de chiffrement de la clé ; un signe moins (-) signifie que la clé ne peut être utilisée pour le chiffrement. Un point (.) signifie qu’elle est marquée en tant que clé de signature dans l’un des identifiants utilisateurs, mais qu’elle peut également servir au chiffrement. La lettre e indique que la clé peut être utilisée pour le chiffrement.

Le second caractère indique les possibilités de signature de la clé. Ici encore, un “-” implique “pas pour signer”, “.” implique que la clé est marquée comme clé de chiffrement pour l’un des identifiants utilisateurs, et “s” caractérise une clé qui peut être utilisée pour signer.

Finalement, le champ de validité (%t) indique le niveau de certification d'un identifiant. Un point d'interrogation (?) indique une validité inconnue, un signe moins (-) caractérise une association non fiable, une espace signifie une association partiellement fiable, et un signe plus (+) indique une validité totale.

### 2.4.3 Envoyer des messages anonymes par l'intermédiaire de mixmaster.

Il se peut également que vous ayez configuré Mutt pour qu'il coopère avec Mixmaster, un réexpéditeur (remailer) anonyme. Mixmaster vous permet d'envoyer vos messages de manière anonyme en utilisant une chaîne de réexpéditeurs. Mutt supporte Mixmaster dans ses version 2.04 (beta 45 semble être la dernière) et 2.03, mais pas les versions précédentes ni la soit-disant version 3 beta, pour laquelle la dernière semble être la 2.9b23.

Pour l'utiliser, vous devrez obéir à certaines restrictions. Plus important, vous ne pouvez utiliser les champs d'en-têtes Cc (copie carbone) et Bcc (copie aveugle). Pour demander à Mutt d'utiliser mixmaster, vous devez sélectionner une chaîne de réexpéditeurs, en utilisant la fonction mix du menu compose.

L'écran de sélection de chaîne est divisée en deux parties. Dans la partie supérieure (la plus importante), figure la liste des réexpéditeurs que vous pouvez utiliser. Dans la partie inférieure, vous voyez la chaîne actuellement sélectionnée de réexpéditeurs.

Vous pouvez naviguer à l'intérieur de la chaîne en utilisant les fonctions `chain-prev` (précédent) et `chain-next` (suivant), dont les raccourcis par défaut sont les flèches gauche et droite ainsi que les touches `h` et `l` (pensez aux raccourcis de vi). Pour insérer un réexpéditeur à la position de chaîne courante, utilisez la fonction `insert`. Pour ajouter un réexpéditeur à la suite de la position courante, utilisez `select-entry` ou `append`. Vous pouvez également supprimer des entrées de la chaîne, en utilisant la fonction idoine. Finalement, pour abandonner vos modifications, quittez le menu, ou **acceptez** les en appuyant sur (par défaut) la touche `Retour`.

Notez que différents réexpéditeurs possèdent différentes capacités, indiquées par l'entrée %c des lignes du menu "remailer" (voir 6.3.106 (\$mix\_entry\_format)). Le plus important est la capacité "intermédiaire" ("middleman") indiquée par un "M" majuscule : elle signifie que le réexpéditeur en question ne peut pas être utilisé en tant qu'élément final d'une chaîne, mais se contentera de faire suivre le message à d'autres réexpéditeurs mixmaster. Pour plus de détails sur les autres capacités, veuillez jeter un oeil à la documentation de Mixmaster.

## 2.5 Faire suivre ou réexpédier un message

Ceci vous permet d'envoyer un message existant aux destinataires spécifiés. La réexpédition d'un message utilise la commande 6.3.198 (sendmail) pour envoyer une copie à d'autres adresses comme si elles étaient les destinataires originelles du message. D'un autre côté, faire suivre un message vous autorise à le modifier avant envoi (par exemple en y ajoutant vos propres commentaires).

Les raccourcis suivants sont disponibles par défaut :

|                |                      |                         |
|----------------|----------------------|-------------------------|
| <code>f</code> | <code>forward</code> | faire suivre un message |
| <code>b</code> | <code>bounce</code>  | réexpédier un message   |

Il est possible de faire suivre en incluant le message original dans le nouveau corps de message (délimité par des lignes indicatrices) ou en tant qu'attachement MIME, selon la valeur de la variable 6.3.103 (\$mime\_forward). Comme dans l'afficheur (pager), le décodage des attachements peut être contrôlé respectivement par les variables 6.3.52 (\$forward\_decode) et 6.3.104 (\$mime\_forward\_decode). Le format désiré peut dépendre du contenu, donc \$mime\_forward est une quadri-option qui peut par exemple être positionnée à "ask-no".

L'inclusion des en-têtes est contrôlée par le réglage courant de la variable 6.3.234 (`$weed`), à moins que 6.3.103 (`mime_forward`) ne soit positionnée.

L'édition d'un message à faire suivre se déroule de la même manière que la composition d'une réponse à un courrier.

## 2.6 Ajourner un courrier

Il est parfois souhaitable de reporter l'envoi d'un courrier dont vous avez commencé la composition. Lorsque vous utilisez la fonction `postpone-message` du menu `compose`, le corps et les attachements du message sont sauvés dans la boîte aux lettres spécifiée par la variable 6.3.167 (`$postponed`). Cela signifie que vous avez la possibilité de rappeler le message plus tard même si vous quittez Mutt et le relancez entre temps.

Lorsqu'un message est ajourné, il y a plusieurs moyens de le reprendre. Depuis la ligne de commande, vous pouvez utiliser l'option “-p”, ou si vous *composez* un nouveau message depuis l'*index* ou l'*afficheur* (pager) Mutt vous rappellera l'existence de messages ajournés le cas échéant. Si plusieurs messages sont actuellement ajournés, le menu `postponed` (“ajournés”) apparaîtra pour vous permettre de choisir le message que vous souhaitez rappeler.

**Remarque :** si vous ajournez la réponse à un courrier, son réglage de réponse n'est mis à jour que lorsque vous avez terminé son édition et l'avez envoyé. De même, vous devez être dans le même dossier que le message auquel vous répondez pour que l'état (indicateurs) du message soit mis à jour.

Voir également la quadri-option 6.3.166 (`$postpone`).

# 3 Configuration

Bien que la configuration par défaut (ou “préférences”) soit suffisante pour pouvoir utiliser Mutt aussitôt après installation, il est souvent souhaitable de l'adapter à vos propres goûts. Quand Mutt est lancé, il cherchera à lire votre fichier de configuration “système” (réglages par défaut positionnés par votre administrateur système), à moins que l'option “-n” de la 6.1 (ligne de commande) ne soit spécifiée. Ce fichier est typiquement `/usr/local/share/mutt/Muttrc` ou `/etc/Muttrc`. Mutt recherchera ensuite un fichier nommé `.muttrc` dans votre répertoire personnel. Si ce fichier est inexistant et que votre répertoire personnel contient un sous-répertoire nommé `.mutt`, Mutt tentera de charger un fichier nommé `.mutt/muttrc`.

`.muttrc` est le fichier dans lequel vous placez normalement vos 6.2 (commandes) destinées à configurer Mutt.

De plus, Mutt peut utiliser des fichiers de configuration spécifiques à une version donnée à la place des fichiers décrits plus haut. Par exemple, si votre répertoire de configuration système contient le fichier `Muttrc-0.88` et que vous utilisez une version 0.88 de Mutt, c'est ce fichier qui sera utilisé au lieu du fichier `Muttrc`. Cela s'applique de la même manière au fichier de configuration utilisateur, si vous avez un fichier `.muttrc-0.88.6` dans votre répertoire personnel et que vous utilisez Mutt 0.88.6, c'est ce fichier qui sera lu à la place du fichier par défaut `.muttrc`. Le numéro de version est celui rendu par l'option “-v” de la 6.1 (ligne de commande) ou la fonction `show-version` (touche par défaut : V) du menu `index`.

## 3.1 Syntaxe des fichiers d'initialisation

Un fichier d'initialisation consiste en une série de 6.2 (commandes). Chaque ligne du fichier est susceptible de contenir une ou plusieurs commandes. Lorsque plusieurs commandes sont utilisées, elles doivent être séparées par un point-virgule (;).

```
set realname='Mutt user' ; ignore x-
```

Le signe dièse (“#”) sert de caractère de commentaire. Vous pouvez l’utiliser pour annoter votre fichier d’initialisation. Tout texte suivant ce caractère de commentaire jusqu’à la fin de la ligne est ignoré. Par exemple,

```
my_hdr X-Disclaimer: Pourquoi me lisez vous ? # Ceci est un commentaire
```

Les apostrophes (’) et les doubles guillemets (") peuvent être utilisés pour “citer” les chaînes contenant des espaces ou autres caractères spéciaux. La différence entre les deux est similaire à celle qui existe dans la plupart des shells, c’est-à-dire que l’apostrophe sert à spécifier une chaîne littérale (une chaîne non interprétée dans les variables shell ou précédée d’un backslash (\) [voir paragraphe suivant]), alors que les doubles guillemets indiquent une chaîne devant être évaluée. Par exemple, les apostrophes inversées (‘) sont évaluées à l’intérieur de doubles guillemets, mais **pas** si elles sont entourées d’apostrophes.

\ protège le prochain caractère, comme avec des shells tels que bash ou zsh. Par exemple, si vous souhaitez mettre un double guillemet “” dans une chaîne, vous pouvez utiliser “\” pour forcer le prochain caractère à être littéral plutôt qu’interprété.

```
set realname="Michael \"MuttDude\" Elkins"
```

“\” permet d’insérer un “\” littéral. “\n” et “\r” ont le sens C habituel de “linefeed” et “retour-chariot”, respectivement.

Un \ à la fin d’une ligne peut être utilisé pour découper des commandes sur plusieurs lignes, du moment que les découpages ne se fassent pas au milieu d’un nom de commande.

Il est également possible de substituer la sortie d’une commande Unix dans un fichier d’initialisation. Cela est accompli en entourant la commande d’apostrophes inversées (‘). Par exemple,

```
my_hdr X-Operating-System: ‘uname -a’
```

La sortie de la commande Unix “uname -a” est substituée avant que la ligne ne soit analysée. Notez que les fichiers d’initialisation étant lus ligne par ligne, seule la première ligne de la sortie d’une commande Unix sera remplacée.

Les variables d’environnement peuvent être accédées de la même manière que dans des shells comme sh et bash : préfixez le nom de la variable d’environnement par un “\$”. Par exemple,

```
set record+=sent_on_${HOSTNAME}
```

Les commandes reconnues par Mutt sont expliquées dans le paragraphe suivant. Pour une liste complète, voyez la [6.2](#) (référence des commandes).

### 3.2 Définir/Utiliser des alias

Utilisation : `alias clé adresse [ , adresse, ... ]`

Il est généralement très compliqué de se souvenir ou de taper l’adresse de quelqu’un avec qui vous correspondez. Mutt vous permet de créer des “alias” qui font correspondre une courte chaîne à une adresse complète.

**Remarque :** si vous voulez créer un alias pour un groupe (en spécifiant plus d’une adresse), vous **devez** séparer les adresses par une virgule (“,”).

Pour supprimer un ou plusieurs alias (“\*” signifie tous les alias) :

```
unalias [ * | clé ... ]
```

```
alias muttdude me@cs.hmc.edu (Michael Elkins)
alias lesgens annie, marc, jacques
```

Contrairement à d'autres logiciels de courrier, Mutt ne requiert pas que les alias soient déclarés dans un fichier spécial. La commande `alias` peut apparaître n'importe où dans un fichier de configuration, du moment que ce fichier est [3.24](#) (lu). Par conséquent, vous pouvez avoir plusieurs fichiers d'alias, ou tous vos alias définis dans votre `muttrc`.

D'un autre côté, la fonction [2.3.4](#) (`create-alias`) ne peut utiliser qu'un seul fichier, celui désigné par la variable [6.3.3](#) (`$alias_file`) (qui est `~/muttrc` par défaut). Ce fichier n'est pas non plus spécial, dans le sens que Mutt se fera une joie d'ajouter des alias à n'importe quel fichier, mais pour que les nouveaux alias puissent être utilisés vous devez explicitement [3.24](#) (lire) également ce fichier.

Par exemple :

```
source /usr/local/share/Mutt.aliases
source ~/.mail_aliases
set alias_file=~/.mail_aliases
```

Vous pouvez plus ou moins utiliser les alias à n'importe quel endroit où Mutt vous réclame des adresses, comme aux invites *To:* ou *Cc:*. Vous pouvez aussi entrer des alias dans votre éditeur pour les champs d'en-têtes appropriés si vous avez positionné la variable [6.3.40](#) (`$edit_headers`).

De plus, aux diverses invites d'adresses, vous pouvez utiliser le caractère tab (tabulation) pour compléter un alias partiel en un alias complet. S'il y a plusieurs possibilités de complétion, Mutt vous affichera un menu présentant les alias correspondant. Pour voir la liste complète des alias, vous devez taper tab sans alias partiel, par exemple au début d'une invite ou après une virgule indiquant plusieurs adresses.

Dans le menu alias, vous pouvez sélectionner autant d'alias que vous le désirez à l'aide de la touche *select-entry* (défaut : RET), et utiliser la touche *exit* (défaut : q) pour retourner à l'invite d'adresses.

### 3.3 Modifier les raccourcis clavier

Utilisation : `bind menu touche fonction`

Cette commande vous permet de changer les réglages par défaut des raccourcis clavier (opérations exécutées lorsque vous tapez sur une touche).

*menu* spécifie à quel menu appartient le raccourci. Les contextes actuellement définis sont :

- generic
- alias
- attach
- browser
- editor
- index
- compose
- pager
- pgp

- postpone

*touche* est la touche (ou séquence de touches) que vous souhaitez affecter. Pour spécifier un caractère de contrôle, utilisez la séquence `\Cx`, où *x* est la lettre du caractère de contrôle (par exemple, pour spécifier control-A utilisez “`\Ca`”). Notez que la casse de *x* et de `\C` sont ignorées, de telle sorte que `\CA`, `\Ca`, `\cA` et `\ca` sont équivalents. Une méthode alternative est de spécifier la touche par sa valeur octale à trois chiffres, préfixée par un “`\`” (par exemple, `\177` est équivalent à `\c?`).

De plus, *touche* peut être :

|                                |                           |
|--------------------------------|---------------------------|
| <code>\t</code>                | tabulation                |
| <code>&lt;tab&gt;</code>       | tabulation                |
| <code>\r</code>                | retour chariot            |
| <code>\n</code>                | nouvelle ligne (linefeed) |
| <code>\e</code>                | échappement (escape)      |
| <code>&lt;esc&gt;</code>       | échappement (escape)      |
| <code>&lt;up&gt;</code>        | flèche haut               |
| <code>&lt;down&gt;</code>      | flèche bas                |
| <code>&lt;left&gt;</code>      | flèche gauche             |
| <code>&lt;right&gt;</code>     | flèche droite             |
| <code>&lt;pageup&gt;</code>    | page précédente           |
| <code>&lt;pagedown&gt;</code>  | page suivante             |
| <code>&lt;backspace&gt;</code> | effacement arrière        |
| <code>&lt;delete&gt;</code>    | effacement                |
| <code>&lt;insert&gt;</code>    | insertion                 |
| <code>&lt;enter&gt;</code>     | entrée                    |
| <code>&lt;return&gt;</code>    | retour                    |
| <code>&lt;home&gt;</code>      | début (home)              |
| <code>&lt;end&gt;</code>       | fin (end)                 |
| <code>&lt;space&gt;</code>     | barre espace              |
| <code>&lt;f1&gt;</code>        | touche de fonction 1      |
| <code>&lt;f10&gt;</code>       | touche de fonction 10     |

Il n'est pas nécessaire d'entourer *touche* de double guillemets à moins qu'il n'y ait une espace (“ ”).

*fonction* spécifie quelle action effectuer quand *touche* est pressée. Pour une liste complète des fonctions, voir [6.4](#) (référence). La fonction spéciale `noop` supprime le raccourci clavier spécifié.

### 3.4 Définir des alias pour les jeux de caractères

Utilisation : `charset-hook alias jeu`

Utilisation : `iconv-hook jeu jeu-local`

La commande `charset-hook` définit un alias pour un jeu de caractères. Ceci est utile pour afficher correctement des messages dont le jeu de caractères annoncé est inconnu de Mutt.

La commande `iconv-hook` définit un nom pour un jeu de caractères spécifique au système. Cela est utile quand la librairie de conversion de caractère de votre système insiste pour utiliser des noms bizarres, spécifiques au système, pour les jeux de caractères.

### 3.5 Positionner des variables selon la boîte aux lettres

Utilisation : `folder-hook [!]regex commande`

Il est souvent souhaitable de modifier la configuration en fonction de la boîte aux lettres que vous lisez. La commande `folder-hook` fournit une méthode qui vous permet d'effectuer n'importe quelle commande de configuration. *regex* est une expression rationnelle indiquant dans quelle boîte exécuter *commande* avant son ouverture. Si une boîte correspond à plusieurs `folder-hooks`, ceux-ci sont exécutés dans leur ordre de définition dans le `muttrc`.

**Remarque :** si vous utilisez le raccourci “!” pour 6.3.213 (`$spoolfile`) au début d'un motif, vous devez le placer entre double guillemets pour pouvoir le distinguer de l'opérateur logique *non*.

Notez que les réglages ne sont *pas* restaurés quand vous quittez la boîte aux lettres. Un exemple d'action est de changer la méthode de tri en fonction de la boîte lue :

```
folder-hook mutt set sort=threads
```

Cependant, la précédente méthode de tri n'est pas restaurée si vous lisez une autre boîte. Pour spécifier une commande par défaut, utilisez le motif “.” :

```
folder-hook . set sort=date-sent
```

### 3.6 Macros clavier

Utilisation : `macro menu touche séquence [ description ]`

Les macros sont pratiques lorsque vous souhaitez qu'une seule touche déclenche une série d'actions. Lorsque vous pressez *touche* dans le menu *menu*, Mutt se comportera comme si vous aviez tapé *séquence*. Donc si vous exécutez fréquemment une certaine séquence de commandes, vous pouvez créer une macro pour exécuter toutes ces commandes d'une simple touche.

*touche* et *séquence* sont développées en suivant les mêmes règles que pour les 3.3 (raccourcis clavier). Mais il y a tout de même des règles supplémentaires. La première est que les caractères de contrôle dans *séquence* peuvent aussi être de la forme  $\hat{x}$ . Deuxièmement, pour spécifier une certaine touche comme *flèche haut* ou pour appeler directement une fonction, vous pouvez utiliser le format `<nom de la touche>` et `<nom de la fonction>`. Pour une liste des noms des touches, reportez-vous à la section sur les 3.3 (raccourcis claviers). Les fonctions sont listées dans la 6.4 (référence des fonctions).

L'avantage d'utiliser les noms des fonctions directement est que les macros fonctionneront quels que soient les raccourcis claviers configurés, et elles ne nécessitent pas que l'utilisateur aie défini ses touches d'une façon particulière. Cela rend les macros plus portables et plus robustes, et cela facilite l'insertion de macros dans des fichiers utilisés par différents utilisateurs (comme le fichier `Muttrc` du système).

Optionnellement, vous pouvez ajouter une description de la macro après *séquence*, elle sera affichée dans l'aide.

**Note :** les définitions de macros listées (s'il y en a) dans l'aide sont tronquées à la largeur de l'écran, sans retour à la ligne.

### 3.7 Utiliser les attributs vidéos de couleur (color) et mono

Utilisation : `color objet premier plan arrière-plan [ expression régulière ]`

Utilisation : `color index premier plan arrière-plan [ motif ]`

Utilisation : `uncolor index motif [ motif ... ]`

Si votre terminal supporte la couleur, vous pouvez pimenter la présentation de Mutt en créant votre propre jeu de couleurs. Pour définir la couleur d'un objet (type d'information), vous avez à spécifier les couleurs de

premier plan (en général, la couleur du texte) **et** d'arrière-plan (il n'est pas possible de ne spécifier qu'une seule des deux couleurs).

*objet* peut prendre une des valeurs suivantes :

- attachment
- body (recherche, “matche”, cette *expression rationnelle* dans le corps des messages)
- bold (mettre en sur-brillance le texte en gras dans le corps des messages)
- error (les messages d'erreur affichés par Mutt)
- header (recherche, “matche”, cette *expression rationnelle* dans les en-têtes du message)
- hdrdefault (couleur par défaut des en-têtes du message dans l'afficheur)
- index (recherche, “matche”, ce *motif* dans l'index des messages)
- indicator (flèche ou barre utilisée pour indiquer l'élément actif dans un menu)
- markers (les marqueurs “+” au début des lignes qui ont subit un retour automatique à ligne dans l'afficheur)
- message (messages d'information)
- normal
- quoted (recherche, “matche”, [6.3.176](#) (\$quote\_regexp) dans le corps des messages)
- quoted1, quoted2, ..., quotedN (niveaux supérieurs de citation de texte)
- search (sur-brillance des mots recherchés dans l'afficheur)
- signature
- status (lignes d'information et de mode des boîtes ou des messages)
- tilde (le “~” utilisé dans l'afficheur pour les lignes vierges à la fin d'un message)
- tree (arborescence des messages d'une enfilade dans l'index des messages et dans le menu des attachements)
- underline (sur-brillance du texte souligné dans le corps des messages)

*premier plan* et *arrière-plan* peuvent prendre une des valeurs suivantes :

- white (blanc)
- black (noir)
- green (vert)
- magenta
- blue (bleu)
- cyan
- yellow (jaune)
- red (rouge)

- default (couleur par défaut)
- color*x* (couleur numéro *x* du terminal)

*premier plan* peut être préfixé du mot-clé **bright** (brillant) pour mettre la couleur du premier plan en gras (par exemple **brightred**).

Si votre terminal le permet, le mot-clé *default* peut être utilisé comme couleur de transparence. La valeur *brightdefault* est également permise. Si Mutt a été compilé avec la bibliothèque *S-Lang*, vous devez également définir la variable d'environnement *COLORFGBG* avec les couleurs par défaut de votre terminal pour que cela fonctionne ; voici un exemple (pour les shells de type Bourne) :

```
set COLORFGBG="green;black"
export COLORFGBG
```

**Note :** La bibliothèque *S-Lang* nécessite l'emploi des mots-clés *lightgray* (gris clair) et *brown* (marron) au lieu de *white* (blanc) et *yellow* (jaune) pour définir cette variable.

**Note :** La commande *uncolor* ne peut être appliquée qu'à l'objet *index*. Elle supprime des entrées de la liste. Vous **devez** spécifier le même motif que celui de la commande *color* pour qu'il soit supprimé. L'utilisation du motif spécial "\*" permet de vider la liste des couleurs à appliquer à l'index de toutes ses entrées.

Mutt reconnaît également les mots-clés *color0*, *color1*, ..., *colorN-1* (**N** étant le nombre de couleurs supporté par votre terminal). C'est particulièrement utile lorsque vous redéfinissez les couleurs affichées par votre terminal (par exemple en changeant la couleur associée à *color2* dans votre *xterm*), puisque les noms des couleurs risquent de ne plus correspondre à leurs définitions.

Si votre terminal ne supporte pas la couleur, il est tout de même possible de changer les attributs de l'affichage en utilisant la commande "mono" :

Utilisation : `mono <objet> <attribut> [ expression régulière ]`

Utilisation : `mono index attribut motif`

Utilisation : `unmono index motif [ motif ... ]`

ou *attribut* peut prendre une des valeurs suivantes :

- none (aucun)
- bold (gras)
- underline (souligné)
- reverse (vidéo inverse)
- standout (fait ressortir le texte)

### 3.8 Ignorer (cacher) des en-têtes de messages

Utilisation : `[un]ignore motif [ motif ... ]`

Les messages ont souvent de nombreux en-têtes qui ont été ajoutés par divers systèmes de traitement, ou qu'il n'est simplement pas utile d'afficher à l'écran. Cette commande vous permet de spécifier quels en-têtes vous ne souhaitez normalement pas voir.

Vous n'avez pas besoin de spécifier le nom complet de l'en-tête. Par exemple, "ignore content-" ignorera tous les en-têtes qui commencent par le motif "content-".

Pour supprimer un token de la liste, utilisez la commande “unignore”. Notez que si vous faites “ignore x-”, il n’est pas possible de faire “unignore x-mailer”, par exemple. La commande “unignore” ne permet **pas** à Mutt d’afficher les en-têtes qui correspondent au motif donné.

“unignore \*” supprime tous les tokens de la liste des en-têtes à ignorer.

Par exemple :

```
# Le draconien défrichage d'en-têtes de Sven
ignore *
unignore from date subject to cc
unignore organization organisation x-mailer: x-newsreader: x-mailing-list:
unignore posted-to:
```

### 3.9 Listes de diffusion

Utilisation : `[un]lists adresse [ adresse ... ]`

Utilisation : `[un]subscribe adresse [ adresse ... ]`

Mutt dispose de quelques fonctionnalités bien pratiques pour 4.8 (gérer les listes de diffusion). Pour en tirer parti, vous devez spécifier quelles adresses sont celles de listes de diffusion, et à quelles listes vous êtes abonné. Une fois cela fait, la fonction 2.3.4 (list-reply) sera disponible pour toutes les listes connues de Mutt. De plus, lorsque vous envoyez un message à une liste à laquelle vous êtes abonné, Mutt ajoutera un en-tête Mail-Followup-To (suivi à) pour annoncer aux logiciels de messagerie des autres abonnés de ne pas envoyer une copie de leurs réponses à votre adresse personnelle. Notez que l’en-tête Mail-Followup-To est une extension non-standardisée et n’est pas supportée par tous les logiciels de messagerie. Ajouter cet en-tête ne constitue donc pas une protection absolue contre la réception de CCs de messages de listes de diffusion. Notez également que l’ajout de l’en-tête Mail-Followup-To est contrôlé par la variable de configuration 6.3.50 (followup\_to).

Plus précisément, Mutt se sert de deux listes distinctes : celle des listes de diffusion connues, et celle des listes auxquelles vous êtes abonné. Toute liste de diffusion à laquelle vous déclarez être abonné est connue. Pour marquer une liste de diffusion comme connue, utilisez la commande “lists”. Pour la marquer comme une liste à laquelle vous êtes abonné, utilisez “subscribe”.

Précisez l’adresse jusqu’à lever toute ambiguïté. Par exemple, si vous vous êtes inscrit à la liste des utilisateurs francophones de Mutt, vous recevrez des messages adressés à `mutt-users-fr@mutt.org`. Donc, pour préciser à Mutt que c’est une liste de diffusion, vous pourriez ajouter “lists mutt-users-fr” à votre fichier d’initialisation. Mais pour dire à Mutt que vous y êtes abonné, utilisez plutôt la commande “subscribe mutt-users-fr”. S’il vous arrive de recevoir des messages de quelqu’un dont l’adresse est `mutt-users-fr@example.com`, vous pourriez utiliser “lists mutt-users-fr@mutt.org” ou “subscribe mutt-users-fr@mutt.org” pour ne parler que des messages de la liste.

La commande “unlists” est utilisée pour supprimer une entrée des listes de diffusions connues ou de celles auxquelles vous êtes abonné. Pour supprimer toutes les entrées, utilisez “unlists \*”.

Pour ne retirer une liste de diffusion que des listes de diffusion auxquelles vous êtes abonné, en la conservant dans les listes de diffusion connues, utilisez “unsubscribe”.

### 3.10 Utiliser plusieurs boîtes de type spoule

Utilisation : `mbox-hook [!]motif boîte`

Cette commande est utilisée pour déplacer les messages lus d’une certaine boîte à une autre boîte automatiquement lorsque vous quittez ou changez de dossier. *motif* est une expression régulière précisant la boîte à

considérer comme un “spoule” et *boîte* spécifique où les messages devraient être déplacés lorsqu'ils ont été lus. Contrairement à certaines autres commandes de type *hook*, seul le premier motif qui correspond est utilisé, les motifs qui suivent sont ignorés (il n'est pas possible de sauver les messages lus dans plus d'une boîte).

### 3.11 Définir les boîtes qui reçoivent du courrier

Utilisation : `mailboxes` [!] *chemin* [ *chemin* ... ]

Cette commande spécifie les dossiers qui peuvent recevoir du courrier et dans lesquels de nouveaux messages seront recherchés. La barre d'état principale affiche combien de ces dossiers ont de nouveaux messages.

Lorsque vous changez de dossier, presser *espace* parcourera la liste des dossiers ayant de nouveaux messages.

Appuyer sur TAB dans le sélecteur de dossiers affichera un menu présentant les fichiers spécifiés par la commande `mailboxes` et indiquera ceux qui contiennent de nouveaux messages. Mutt entrera directement dans ce mode si vous utilisez l'option `-y` depuis la ligne de commande.

**Note** : les nouveaux messages sont détectés en comparant la date de dernière modification avec la date du dernier accès. Les utilitaires tels que `biff` ou `frm` ou tout autre programme qui accède aux boîtes aux lettres peuvent empêcher Mutt de détecter l'arrivée de nouveaux messages dans ces boîtes à lettres si la date du dernier accès n'est pas correctement restaurée. Les outils de sauvegarde sont également une cause très répandue de date de dernier accès modifiée.

**Note** : les chemins dans la commande `mailboxes` sont résolus lorsque la commande est exécutée, donc si ces noms contiennent des 4.7 (caractères de raccourci) (comme “=” et “!”), toute les définitions de variables qui affectent ces caractères (comme 6.3.48 (`$folder`) et 6.3.213 (`$spoolfile`)) doivent être exécutées avant la commande `mailboxes`.

### 3.12 User defined headers

Usage:

`my_hdr` *string*

`unmy_hdr` *field* [ *field* ... ]

The “`my_hdr`” command allows you to create your own header fields which will be added to every message you send.

For example, if you would like to add an “Organization:” header field to all of your outgoing messages, you can put the command

```
my_hdr Organization: A Really Big Company, Anytown, USA
```

in your `.muttrc`.

**Note**: space characters are *not* allowed between the keyword and the colon (“:”). The standard for electronic mail (RFC822) says that space is illegal there, so Mutt enforces the rule.

If you would like to add a header field to a single message, you should either set the 6.3.40 (`edit_headers`) variable, or use the *edit-headers* function (default: “E”) in the send-menu so that you can edit the header of your message along with the body.

To remove user defined header fields, use the “`unmy_hdr`” command. You may specify an asterisk (“\*”) to remove all header fields, or the fields to remove. For example, to remove all “To” and “Cc” header fields, you could use:

```
unmy_hdr to cc
```

### 3.13 Defining the order of headers when viewing messages

Usage: `hdr_order header1 header2 header3`

With this command, you can specify an order in which mutt will attempt to present headers to you when viewing messages.

“`unhdr_order *`” will clear all previous headers from the order list, thus removing the header order effects set by the system-wide startup file.

```
hdr_order From Date: From: To: Cc: Subject:
```

### 3.14 Specify default save filename

Usage: `save-hook [!]pattern filename`

This command is used to override the default filename used when saving messages. *filename* will be used as the default filename if the message is *From:* an address matching *regexp* or if you are the author and the message is addressed *to:* something matching *regexp*.

See 4.4.1 (Message Matching in Hooks) for information on the exact format of *pattern*.

Examples:

```
save-hook me@(turing\\.?)?cs\\.hmc\\.edu$ +elkins
save-hook aol\\.com$ +spam
```

Also see the 3.16 (`fcc-save-hook`) command.

### 3.15 Specify default Fcc: mailbox when composing

Usage: `fcc-hook [!]pattern mailbox`

This command is used to save outgoing mail in a mailbox other than 6.3.181 (`$record`). Mutt searches the initial list of message recipients for the first matching *regexp* and uses *mailbox* as the default Fcc: mailbox. If no match is found the message will be saved to 6.3.181 (`$record`) mailbox.

See 4.4.1 (Message Matching in Hooks) for information on the exact format of *pattern*.

Example: `fcc-hook aol.com$ +spammers`

The above will save a copy of all messages going to the aol.com domain to the ‘+spammers’ mailbox by default. Also see the 3.16 (`fcc-save-hook`) command.

### 3.16 Specify default save filename and default Fcc: mailbox at once

Usage: `fcc-save-hook [!]pattern mailbox`

This command is a shortcut, equivalent to doing both a 3.15 (`fcc-hook`) and a 3.14 (`save-hook`) with its arguments.

### 3.17 Change settings based upon message recipients

Usage: `send-hook [!]pattern command`

This command can be used to execute arbitrary configuration commands based upon recipients of the message. *pattern* is a regular expression matching the desired address. *command* is executed when *regex* matches recipients of the message. When multiple matches occur, commands are executed in the order they are specified in the muttrc.

See 4.4.1 (Message Matching in Hooks) for information on the exact format of *pattern*.

Example: `send-hook mutt "set mime_forward signature=''"`

Another typical use for this command is to change the values of the 6.3.15 (\$attribution), 6.3.203 (\$signature) and 6.3.86 (\$locale) variables in order to change the language of the attributions and signatures based upon the recipients.

**Note:** the send-hook's are only executed ONCE after getting the initial list of recipients. Adding a recipient after replying or editing the message will NOT cause any send-hook to be executed. Also note that my\_hdr commands which modify recipient headers, or the message's subject, don't have any effect on the current message when executed from a send-hook.

### 3.18 Change settings before formatting a message

Usage: `message-hook [!]pattern command`

This command can be used to execute arbitrary configuration commands before viewing or formatting a message based upon information about the message. *command* is executed if the *pattern* matches the message to be displayed. When multiple matches occur, commands are executed in the order they are specified in the muttrc.

See 4.4.1 (Message Matching in Hooks) for information on the exact format of *pattern*.

Example:

```
message-hook ~A 'set pager=builtin'
message-hook '~f freshmeat-news' 'set pager="less \"+/^ subject: .*\"'
```

### 3.19 Choosing the PGP key of the recipient

Usage: `pgp-hook pattern keyid`

When encrypting messages with PGP, you may want to associate a certain PGP key with a given e-mail address automatically, either because the recipient's public key can't be deduced from the destination address, or because, for some reasons, you need to override the key Mutt would normally use. The `pgp-hook` command provides a method by which you can specify the ID of the public key to be used when encrypting messages to a certain recipient.

### 3.20 Adding key sequences to the keyboard buffer

Usage: `push string`

This command adds the named string to the keyboard buffer. The string may contain control characters, key names and function names like the sequence string in the 3.6 (macro) command. You may use it to automatically run a sequence of commands at startup, or when entering certain folders.

### 3.21 Executing functions

Usage: `exec function [ function ... ]`

This command can be used to execute any function. Functions are listed in the 6.4 (function reference). “exec function” is equivalent to “push <function>”.

### 3.22 Message Scoring

Usage: `score pattern value`

Usage: `unscore pattern [ pattern ... ]`

The `score` commands adds *value* to a message’s score if *pattern* matches it. *pattern* is a string in the format described in the 4.2 (patterns) section (note: For efficiency reasons, patterns which scan information not available in the index, such as `~b`, `~B` or `~h`, may not be used). *value* is a positive or negative integer. A message’s final score is the sum total of all matching `score` entries. However, you may optionally prefix *value* with an equal sign (=) to cause evaluation to stop at a particular entry if there is a match. Negative final scores are rounded up to 0.

The `unscore` command removes score entries from the list. You **must** specify the same pattern specified in the `score` command for it to be removed. The pattern “\*” is a special token which means to clear the list of all score entries.

### 3.23 Setting variables

Usage: `set [no|inv]variable[=value] [ variable ... ]`

Usage: `toggle variable [variable ... ]`

Usage: `unset variable [variable ... ]`

Usage: `reset variable [variable ... ]`

This command is used to set (and unset) 6.3 (configuration variables). There are four basic types of variables: boolean, number, string and quadoption. *boolean* variables can be *set* (true) or *unset* (false). *number* variables can be assigned a positive integer value.

*string* variables consist of any number of printable characters. *strings* must be enclosed in quotes if they contain spaces or tabs. You may also use the “C” escape sequences `\n` and `\t` for newline and tab, respectively.

*quadoption* variables are used to control whether or not to be prompted for certain actions, or to specify a default action. A value of *yes* will cause the action to be carried out automatically as if you had answered yes to the question. Similarly, a value of *no* will cause the the action to be carried out as if you had answered “no.” A value of *ask-yes* will cause a prompt with a default answer of “yes” and *ask-no* will provide a default answer of “no.”

Prefixing a variable with “no” will unset it. Example: `set noaskbcc`.

For *boolean* variables, you may optionally prefix the variable name with `inv` to toggle the value (on or off). This is useful when writing macros. Example: `set invsmart.wrap`.

The `toggle` command automatically prepends the `inv` prefix to all specified variables.

The `unset` command automatically prepends the `no` prefix to all specified variables.

Using the enter-command function in the *index* menu, you can query the value of a variable by prefixing the name of the variable with a question mark:

```
set ?allow_8bit
```

The question mark is actually only required for boolean and quadoption variables.

The `reset` command resets all given variables to the compile time defaults (hopefully mentioned in this manual). If you use the command `set` and prefix the variable with “&” this has the same behavior as the `reset` command.

With the `reset` command there exists the special variable “all”, which allows you to reset all variables to their system defaults.

### 3.24 Reading initialization commands from another file

Usage: `source filename`

This command allows the inclusion of initialization commands from other files. For example, I place all of my aliases in `~/.mail_aliases` so that I can make my `~/.muttrc` readable and keep my aliases private.

If the filename begins with a tilde (“~”), it will be expanded to the path of your home directory.

If the filename ends with a vertical bar (`|`), then *filename* is considered to be an executable program from which to read input (eg. `source ~bin/myscript|/`).

### 3.25 Removing hooks

Usage: `unhook [ * | hook-type ]`

This command permits you to flush hooks you have previously defined. You can either remove all hooks by giving the “\*” character as an argument, or you can remove all hooks of a specific type by saying something like `unhook send-hook`.

## 4 Utilisation avancée

### 4.1 Expressions Rationnelles

Toutes les chaînes de motifs de Mutt incluant des 4.2 (motifs) plus complexes doivent être indiqués en utilisant des expressions rationnelles (regex) de la norme “POSIX étendue” (qui est plus ou moins la syntaxe employée par `grep` et `GNU awk`). Pour votre convenance, nous avons inclus ci-dessous une courte description de cette syntaxe.

La recherche est sensible à la casse si le motif contient au moins une lettre majuscule, sinon elle y est insensible. Notez que le “\” doit être protégé s’il est utilisé dans une expression rationnelle et dans une commande d’initialisation : “\\”.

Une expression rationnelle est un motif qui décrit un ensemble de chaînes de caractères. Les expressions rationnelles sont construites de façon analogue aux expressions arithmétiques, en employant divers opérateurs pour combiner de plus petites expressions.

Notez qu’une expression rationnelle peut être incluse/délimitée par “ ” ou ‘ ’ qui est utile si l’expression rationnelle inclut un espace. Voir la 3.1 (syntaxe des dossiers d’initialisation) pour plus d’information sur le traitement du délimiteur “ ” et ‘ ’. Pour qu’une expression rationnelle reconnaisse un “ ” ou ‘ ’ vous devez l’échapper avec “\ ” (antislash).

Les modules fondamentaux sont les expressions rationnelles qui font correspondre un unique caractère. La plupart des caractères, y compris toutes les lettres et les chiffres, sont des expressions rationnelles qui se reconnaissent elle mêmes. N’importe quel méta-caractère avec une signification spéciale peut être échappé en le précédant d’un antislash. Le caractère “.” reconnaît n’importe quel caractère simple. L’accent circonflexe

“^” et le symbole dollar “\$” sont des méta-caractères qui font correspondre respectivement la chaîne vide au commencement et à la fin d’une ligne.

Une liste de caractères délimitée par “[” et “]” reconnaît n’importe quel caractère dans cette liste ; si le premier caractère de la liste est un accent circonflexe “^” alors il fait correspondre n’importe quel caractère qui n’est pas dans cette liste. Par exemple, l’expression rationnelle [0123456789] reconnaît n’importe quel chiffre. Une gamme de caractères ASCII peut être indiquée en donnant le premier et le dernier caractère, séparées par un trait d’union “-”. La plupart des méta-caractères perdent leur signification spéciale à l’intérieur des listes. Pour inclure un “]” placez-le en premier dans la liste. De même, pour inclure un “^” placez-le n’importe où mais pas en premier. En conclusion, pour inclure un trait d’union “-” placez-le dernier.

Certaines classes de caractères sont prédéfinies. Les classes de caractères se composent de “[:”, un mot-clé dénotant la classe, et “:]”. Les classes suivantes sont définies par la norme POSIX :

**[:alnum:]**

Caractères alphanumériques.

**[:alpha:]**

Caractères alphabétiques.

**[:blank:]**

Espace ou tabulation.

**[:cntrl:]**

Caractères de contrôle.

**[:digit:]**

Chiffres.

**[:graph:]**

Caractères qui sont imprimables et visibles. (l’espace est imprimable, mais non visible, alors qu’un “a” est visible et imprimable.)

**[:lower:]**

Lettres minuscules.

**[:print:]**

Caractères imprimables (caractères qui ne sont pas les caractères de contrôle.)

**[:punct:]**

Caractères de ponctuation (caractères qui ne sont pas des lettres, des chiffres, des caractères de contrôle, ou des caractères d’espacement).

**[:space:]**

Caractères d’espacement (tels que l’espace, la tabulation, le chargement de page,...).

**[:upper:]**

Lettres majuscules.

**[:xdigit:]**

Caractères hexadécimaux.

Une classe de caractères est valide seulement dans une expression rationnelle à l'intérieur des parenthèses d'une liste de caractères. Notez que les parenthèses dans ces noms de classe font partie des noms symboliques, et doivent être inclus en plus des parenthèses délimitant cette liste. Par exemple `[:digit:]` est équivalent à `[0-9]`.

Deux séquences supplémentaires peuvent apparaître dans les listes de caractères. Elles s'appliquent aux jeux de caractères non ASCII pouvant avoir des symboles simples (appelés éléments de collation, c'est-à-dire des ensembles de lettres formant un tout indissociable pour le classement alphabétique) qui sont représentés par plus d'un caractère, de même qu'à plusieurs caractères équivalents du point de vue du tri et de la comparaison :

### Symboles de collation

Un symbole de collation est un élément de collation multi-caractères enfermé dans des “[.” et “.]”. Par exemple, si “ch” est un élément de collation, alors `[.ch.]` est une expression rationnelle qui correspond à cet élément, alors que `[ch]` est une expression rationnelle qui correspond à “c” ou à “h”.

### Classes d'équivalence

Une classe d'équivalence est le nom (localement spécifique) d'une liste de caractères équivalents. Le nom est délimité par “[=” et “=]”. Par exemple, le “e” pourrait être employé pour représenter tous les “è”, “é” et “e”. Dans ce cas-ci, `[=e=]` est une regexp qui représente n'importe quel caractère “è”, “é” et “e”.

Une expression rationnelle reconnaissant un unique caractère peut être suivie d'un ou de plusieurs opérateurs de répétition :

?

Le caractère précédent est facultatif et est reconnu tout au plus une fois.

\*

Le caractère précédent sera reconnu zéro fois ou plus.

+

Le caractère précédent sera reconnu une ou plusieurs fois.

{n}

Le caractère précédent est reconnu exactement  $n$  fois.

{n,}

Le caractère précédent est reconnu  $n$  fois ou plus.

{,m}

Le caractère précédent est reconnu entre 0 et  $n$  fois.

{n,m}

Le caractère précédent est reconnu au moins  $n$  fois et au plus  $m$  fois.

Deux expressions rationnelles peuvent être concaténées ; l'expression rationnelle résultante reconnaît n'importe quelle chaîne constituée en concaténant les deux sous-chaînes qui reconnaissent respectivement les sous-expressions concaténées.

Deux expressions rationnelles peuvent être jointes par le symbole “[|” ; l'expression rationnelle résultante reconnaît n'importe quelle chaîne reconnaissant l'une ou l'autre des deux sous-expressions.

Les répétitions ont priorité sur la concaténation, qui ont priorité sur l'alternance. Une sous-expression entière peut être délimitée entre parenthèses pour changer ces priorités.

**Note :** Si vous compilez Mutt avec le paquet GNU *rx*, les opérateurs suivants peuvent également être employés dans des expressions rationnelles :

`\\y`

Reconnaît une chaîne vide au début ou à la fin d'un mot.

`\\B`

Reconnaît une chaîne vide dans un mot.

`\\<`

Reconnaît une chaîne vide au début d'un mot.

`\\>`

Reconnaît une chaîne vide à la fin d'un mot.

`\\w`

Reconnaît n'importe quel caractère pouvant être contenu dans un mot (lettre, chiffre, ou tiret bas (underscore)).

`\\W`

Reconnaît n'importe quel caractère qui ne fait pas parti d'un mot.

`\\‘`

Reconnaît une chaîne vide au début d'un tampon (chaîne).

`\\’`

Reconnaît une chaîne vide à la fin d'un tampon.

Notez cependant que ces opérateurs ne sont pas définis par POSIX, ainsi ils peuvent être disponibles ou pas dans les bibliothèques courantes sur de divers systèmes.

## 4.2 Motifs

Plusieurs des commandes de Mutt vous permettent d'indiquer un motif pour sélectionner des messages (`limit`, `tag-pattern`, `delete-pattern`, etc.). Il y a plusieurs manières de choisir des messages :

|                             |   |
|-----------------------------|---|
| <code>~A</code>             | tous les messages                                       |
| <code>~b EXPR</code>        | messages contenant EXPR dans leur corps                 |
| <code>~B EXPR</code>        | messages contenant EXPR (intégralité du message)        |
| <code>~c USER</code>        | messages avec une copie carbone (cc:) pour USER         |
| <code>~C EXPR</code>        | message est destiné à (to:) ou en copie (cc:) pour EXPR |
| <code>~D</code>             | messages effacés  |
| <code>~d [MIN]-[MAX]</code> | messages avec ‘‘date-d-envoi’’ dans l'intervalle Date   |
| <code>~E</code>             | messages arrivés à expiration                           |
| <code>~e EXPR</code>        | message contenant EXPR dans le champ ‘‘Sender’’         |
| <code>~F</code>             | messages marqués important                              |
| <code>~f USER</code>        | messages provenant de USER                              |
| <code>~g</code>             | messages signés (PGP)                                   |
| <code>~G</code>             | messages chiffrés (PGP)                                 |

|   |   |
|---|---|
| <code>~h</code> <code>EXPR</code>                       | messages contenant <code>EXPR</code> dans leurs en-têtes                          |
| <code>~k</code>   | message contenant une clé PGP   |
| <code>~i</code> <code>ID</code>                         | message avec <code>ID</code> dans le champ ‘‘Message-ID’’                         |
| <code>~L</code> <code>EXPR</code>                       | message provenant de ou reçu par <code>EXPR</code>                                |
| <code>~l</code>   | message adressé à une liste de diffusion connue                                   |
| <code>~m</code> <code>[MIN]</code> - <code>[MAX]</code> | message dans l’intervalle <code>MIN</code> à <code>MAX</code> *)                  |
| <code>~n</code> <code>[MIN]</code> - <code>[MAX]</code> | messages avec un score dans l’intervalle <code>MIN</code> à <code>MAX</code> *)   |
| <code>~N</code>   | nouveaux messages   |
| <code>~O</code>   | anciens messages  |
| <code>~p</code>   | message qui vous est adressé (voir <code>\$alternates</code> )                    |
| <code>~P</code>   | message que vous avez envoyé (voir <code>\$alternates</code> )                    |
| <code>~Q</code>   | messages auxquels vous avez répondu   |
| <code>~R</code>   | messages lus  |
| <code>~r</code> <code>[MIN]</code> - <code>[MAX]</code> | messages avec ‘‘date-reception’’ dans l’intervalle <code>Date</code>              |
| <code>~S</code>   | messages remplacés (supersede)  |
| <code>~s</code> <code>SUBJECT</code>                    | messages ayant <code>SUBJECT</code> dans l’objet (champ ‘‘Subject’’).             |
| <code>~T</code>   | messages marqués (tag)  |
| <code>~t</code> <code>USER</code>                       | messages adressés à <code>USER</code>   |
| <code>~U</code>   | messages non lus  |
| <code>~v</code>   | message faisant partie d’une enfilade repliée                                     |
| <code>~x</code> <code>EXPR</code>                       | messages contenant <code>EXPR</code> dans le champ ‘References’                   |
| <code>~y</code> <code>EXPR</code>                       | messages contenant <code>EXPR</code> dans le champ ‘X-Label’                      |
| <code>~z</code> <code>[MIN]</code> - <code>[MAX]</code> | messages avec une taille dans l’intervalle <code>MIN</code> à <code>MAX</code> *) |
| <code>~=</code>   | doublons (voir <code>\$duplicate_threads</code> )                                 |

Où `EXPR`, `USER`, `ID`, et `SUBJECT` sont des 4.1 (expressions rationnelles).

\*) Les expressions `<[MAX]`, `>[MIN]`, `[MIN]`- et `-[MAX]` sont aussi autorisées.

#### 4.2.1 Modificateur de motifs

Notons que les motifs qui reconnaissent des ‘listes’ d’adresses (notamment `c`, `C`, `p`, `P` et `t`) les reconnaissent s’il y en a au moins une qui correspond dans la liste entière. Si vous voulez vous assurer que tous les éléments de cette liste correspondent, vous devez mettre en tête de votre motif le caractère ‘‘^’’. Cet exemple reconnaît tous les courriers qui sont seulement destinés à l’Allemagne.

```
^^C \.de$
```

#### 4.2.2 Motifs Complexes

L’expression logique AND est appelée en indiquant plusieurs critères. Par exemple :

```
~t mutt ~f elkins
```

choisit les messages qui contiennent le mot ‘‘mutt’’ dans liste des destinataires **et** qui contiennent le mot ‘‘elkins’’ dans le champ ‘‘From’’ de l’en-tête.

Mutt reconnaît également les opérateurs suivants pour créer des motifs de recherche plus complexes :

- `!` – opérateur NON
- `|` – opérateur OU
- `()` – groupement d’opérateurs

Voici un exemple illustrant un motif complexe de recherche. Ce motif va sélectionner tous les messages qui ne contiennent pas mutt dans les champs “To” ou “Cc” et qui proviennent de “elkins”.

```
!(~t mutt|~c mutt) ~f elkins
```

Voici un exemple utilisant des espaces dans les expressions rationnelles. (notez les délimiteurs ‘ et “). Pour que ceci corresponde, le sujet du courrier doit reconnaître “^Junk +From +Me\$” et doit être en provenance de “Jim +Somebody” ou de “Ed +SomeoneElse” :

```
'~s "^Junk +From +Me$" ~f ("Jim +Somebody"|"Ed +SomeoneElse")'
```

### 4.2.3 Recherche par date

Mutt supporte deux types de dates, les dates *absolues* et les dates *relatives*.

**Absolue.** Les dates doivent être de la forme JJ/MM/AA (le mois et l’année sont facultatifs, par défaut seront pris en compte le mois et l’année en cours). Voici un exemple d’intervalle de dates :

```
Limite les messages correspondant à : ~d 20/1/95-31/10
```

Si vous omettez la (première) date minimum, et que vous spécifiez juste “-DD/MM/YY”, tous les messages *avant* la date donnée seront sélectionnés. Si vous omettez la (seconde) date maximum, et que vous spécifiez “DD/MM/YY-”, tous les messages *après* la date donnée seront sélectionnés. Si vous indiquez une seule date sans le tiret (“-”), seuls les messages envoyés à la date donnée seront sélectionnés.

**Marges d’erreur.** Vous pouvez ajouter des marges d’erreur aux dates absolues. Une marge d’erreur est un signe (+ ou -), suivi d’un chiffre, suivi d’une des unités suivantes :

|   |          |
|---|----------|
| y | années   |
| m | mois     |
| w | semaines |
| d | jours    |

Vous pouvez remplacer le signe par le caractère “\*”, qui est équivalent à renseigner plus et moins dans les marges d’erreur.

Exemple : Pour choisir n’importe quel message deux semaines aux alentours du 15 Janvier 2001, vous pouvez utiliser le motif suivant :

```
Limite les messages correspondant à : ~d 15/1/2001*2w
```

**Relatif.** Ce type de date est relatif à la date du jour, et peut être spécifié comme suit :

- >*décalage* (messages plus vieux que le *décalage*)
- <*décalage* (messages plus récents que le *décalage*)
- =*décalage* (messages qui correspondent exactement au *décalage*)

Les *décalages* sont indiqués avec des nombres positifs suivis d’une des unités suivantes :

|   |          |
|---|----------|
| y | années   |
| m | mois     |
| w | semaines |
| d | jours    |

Exemple : Pour sélectionner les messages de moins d'un mois, vous pouvez utiliser :

```
Limite les messages correspondant à : ~d <1m
```

**Note** : toutes les dates employées quand on fait une recherche sont relatives au fuseau horaire **local**, à moins que vous ne changiez la définition de votre [6.3.83](#) (`$index_format`) pour inclure un autre format du type `%[...]` (ce **ne sont pas** les dates affichées dans l'index principal).

### 4.3 Using Tags

Sometimes it is desirable to perform an operation on a group of messages all at once rather than one at a time. An example might be to save messages to a mailing list to a separate folder, or to delete all messages with a given subject. To tag all messages matching a pattern, use the `tag-pattern` function, which is bound to “shift-T” by default. Or you can select individual messages by hand using the “tag-message” function, which is bound to “t” by default. See [4.2](#) (patterns) for Mutt’s pattern matching syntax.

Once you have tagged the desired messages, you can use the “tag-prefix” operator, which is the “;” (semicolon) key by default. When the “tag-prefix” operator is used, the **next** operation will be applied to all tagged messages if that operation can be used in that manner. If the [6.3.17](#) (`$auto_tag`) variable is set, the next operation applies to the tagged messages automatically, without requiring the “tag-prefix”.

### 4.4 Using Hooks

A *hook* is a concept borrowed from the EMACS editor which allows you to execute arbitrary commands before performing some operation. For example, you may wish to tailor your configuration based upon which mailbox you are reading, or to whom you are sending mail. In the Mutt world, a *hook* consists of a [4.1](#) (regular expression) or [4.2](#) (pattern) along with a configuration option/command. See

- [3.5](#) (folder-hook)
- [3.17](#) (send-hook)
- [3.18](#) (message-hook)
- [3.14](#) (save-hook)
- [3.10](#) (mbox-hook)
- [3.15](#) (fcc-hook)
- [3.16](#) (fcc-save-hook)

for specific details on each type of *hook* available.

**Note:** if a hook changes configuration settings, these changes remain effective until the end of the current mutt session. As this is generally not desired, a default hook needs to be added before all other hooks to restore configuration defaults. Here is an example with `send-hook` and the `my_hdr` directive:

```
send-hook . 'unmy_hdr From:'  
send-hook ~Cb@b.b my_hdr from: c@c.c
```

#### 4.4.1 Message Matching in Hooks

Hooks that act upon messages (`send-hook`, `save-hook`, `fcc-hook`, `message-hook`) are evaluated in a slightly different manner. For the other types of hooks, a 4.1 (regular expression) is sufficient. But in dealing with messages a finer grain of control is needed for matching since for different purposes you want to match different criteria.

Mutt allows the use of the 4.2 (search pattern) language for matching messages in hook commands. This works in exactly the same way as it would when *limiting* or *searching* the mailbox, except that you are restricted to those operators which match information mutt extracts from the header of the message (i.e. from, to, cc, date, subject, etc.).

For example, if you wanted to set your return address based upon sending mail to a specific address, you could do something like:

```
send-hook '~t ^me@cs\.hmc\.edu$' 'my_hdr From: Mutt User <user@host>'
```

which would execute the given command when sending mail to `me@cs.hmc.edu`.

However, it is not required that you write the pattern to match using the full searching language. You can still specify a simple *regular expression* like the other hooks, in which case Mutt will translate your pattern into the full language, using the translation specified by the 6.3.31 (`$default_hook`) variable. The pattern is translated at the time the hook is declared, so the value of 6.3.31 (`$default_hook`) that is in effect at that time will be used.

#### 4.5 External Address Queries

Mutt supports connecting to external directory databases such as LDAP, ph/qi, bddb, or NIS through a wrapper script which connects to mutt using a simple interface. Using the 6.3.174 (`$query_command`) variable, you specify the wrapper command to use. For example:

```
set query_command = "mutt_ldap_query.pl '%s'"
```

The wrapper script should accept the query on the command-line. It should return a one line message, then each matching response on a single line, each line containing a tab separated address then name then some other optional information. On error, or if there are no matching addresses, return a non-zero exit code and a one line error message.

An example multiple response output:

```
Searching database ... 20 entries ... 3 matching:
me@cs.hmc.edu      Michael Elkins  mutt dude
blong@fiction.net  Brandon Long   mutt and more
roessler@guug.de   Thomas Roessler mutt pgp
```

There are two mechanisms for accessing the query function of mutt. One is to do a query from the index menu using the query function (default: Q). This will prompt for a query, then bring up the query menu which will list the matching responses. From the query menu, you can select addresses to create aliases, or to mail. You can tag multiple messages to mail, start a new query, or have a new query appended to the current responses.

The other mechanism for accessing the query function is for address completion, similar to the alias completion. In any prompt for address entry, you can use the complete-query function (default: ^T) to run a query based on the current address you have typed. Like aliases, mutt will look for what you have typed back to

the last space or comma. If there is a single response for that query, mutt will expand the address in place. If there are multiple responses, mutt will activate the query menu. At the query menu, you can select one or more addresses to be added to the prompt.

## 4.6 Mailbox Formats

Mutt supports reading and writing of four different mailbox formats: mbox, MMDF, MH and Maildir. The mailbox type is autodetected, so there is no need to use a flag for different mailbox types. When creating new mailboxes, Mutt uses the default specified with the [6.3.95](#) (`$mbox_type`) variable.

**mbox.** This is the most widely used mailbox format for UNIX. All messages are stored in a single file. Each message has a line of the form:

```
From me@cs.hmc.edu Fri, 11 Apr 1997 11:44:56 PST
```

to denote the start of a new message (this is often referred to as the “From.” line).

**MMDF.** This is a variant of the *mbox* format. Each message is surrounded by lines containing “^A^A^A^A” (four control-A’s).

**MH.** A radical departure from *mbox* and *MMDF*, a mailbox consists of a directory and each message is stored in a separate file. The filename indicates the message number (however, this may not correspond to the message number Mutt displays). Deleted messages are renamed with a comma (,) prepended to the filename. **Note:** Mutt detects this type of mailbox by looking for either `.mh_sequences` or `.xmhcache` (needed to distinguish normal directories from MH mailboxes).

**Maildir.** The newest of the mailbox formats, used by the Qmail MTA (a replacement for sendmail). Similar to *MH*, except that it adds three subdirectories of the mailbox: *tmp*, *new* and *cur*. Filenames for the messages are chosen in such a way they are unique, even when two programs are writing the mailbox over NFS, which means that no file locking is needed.

## 4.7 Mailbox Shortcuts

There are a number of built in shortcuts which refer to specific mailboxes. These shortcuts can be used anywhere you are prompted for a file or mailbox path.

- `!` – refers to your [6.3.213](#) (`$spoolfile`) (incoming) mailbox
- `>` – refers to your [6.3.94](#) (`$mbox`) file
- `<` – refers to your [6.3.181](#) (`$record`) file
- `-` or `!!` – refers to the file you’ve last visited
- `~` – refers to your home directory
- `=` or `+` – refers to your [6.3.48](#) (`$folder`) directory
- `@alias` – refers to the [3.14](#) (default save folder) as determined by the address of the alias

## 4.8 Handling Mailing Lists

Mutt has a few configuration options that make dealing with large amounts of mail easier. The first thing you must do is to let Mutt know what addresses you consider to be mailing lists (technically this does not have to be a mailing list, but that is what it is most often used for), and what lists you are subscribed to. This is accomplished through the use of the 3.9 (`lists` and `subscribe`) commands in your `muttrc`.

Now that Mutt knows what your mailing lists are, it can do several things, the first of which is the ability to show the name of a list through which you received a message (i.e., of a subscribed list) in the *index* menu display. This is useful to distinguish between personal and list mail in the same mailbox. In the 6.3.83 (`$index_format`) variable, the escape “%L” will return the string “To <list>” when “list” appears in the “To” field, and “Cc <list>” when it appears in the “Cc” field (otherwise it returns the name of the author).

Often times the “To” and “Cc” fields in mailing list messages tend to get quite large. Most people do not bother to remove the author of the message they are reply to from the list, resulting in two or more copies being sent to that person. The “list-reply” function, which by default is bound to “L” in the *index* menu and *pager*, helps reduce the clutter by only replying to the known mailing list addresses instead of all recipients (except as specified by `Mail-Followup-To`, see below).

Mutt also supports the `Mail-Followup-To` header. When you send a message to a list of recipients which includes one or several subscribed mailing lists, and if the 6.3.50 (`$followup_to`) option is set, mutt will generate a `Mail-Followup-To` header which contains all the recipients to whom you send this message, but not your address. This indicates that group-replies or list-replies (also known as “followups”) to this message should only be sent to the original recipients of the message, and not separately to you - you’ll receive your copy through one of the mailing lists you are subscribed to.

Conversely, when group-replying or list-replying to a message which has a `Mail-Followup-To` header, mutt will respect this header if the 6.3.66 (`$honor_followup_to`) configuration variable is set. Using list-reply will in this case also make sure that the reply goes to the mailing list, even if it’s not specified in the list of recipients in the `Mail-Followup-To`.

Note that, when header editing is enabled, you can create a `Mail-Followup-To` header manually. Mutt will only auto-generate this header if it doesn’t exist when you send the message.

The other method some mailing list admins use is to generate a “Reply-To” field which points back to the mailing list address rather than the author of the message. This can create problems when trying to reply directly to the author in private, since most mail clients will automatically reply to the address given in the “Reply-To” field. Mutt uses the 6.3.184 (`$reply_to`) variable to help decide which address to use. If set, you will be prompted as to whether or not you would like to use the address given in the “Reply-To” field, or reply directly to the address given in the “From” field. When unset, the “Reply-To” field will be used when present.

The “X-Label:” header field can be used to further identify mailing lists or list subject matter (or just to annotate messages individually). The 6.3.83 (`$index_format`) variable’s “%y” and “%Y” escapes can be used to expand “X-Label:” fields in the index, and Mutt’s pattern-matcher can match regular expressions to “X-Label:” fields with the “y” selector. “X-Label:” is not a standard message header field, but it can easily be inserted by procmail and other mail filtering agents.

Lastly, Mutt has the ability to 6.3.208 (`sort`) the mailbox into 2.3.3 (threads). A thread is a group of messages which all relate to the same subject. This is usually organized into a tree-like structure where a message and all of its replies are represented graphically. If you’ve ever used a threaded news client, this is the same concept. It makes dealing with large volume mailing lists easier because you can easily delete uninteresting threads and quickly find topics of value.

## 4.9 Support de DSN (notification de statut d’acheminement)

La RFC 1894 (DSN : Delivery Status Notification) définit un jeu de types MIME destinés à relayer l’état des courriers électroniques. Ceux-ci peuvent être vus comme des “accusés de réception”. Berkeley sendmail 8.8.x possède des options de ligne de commande permettant au logiciel de messagerie client de choisir le type de messages d’état devant être retourné.

Deux variables sont disponibles pour gérer ceci. La variable 6.3.37 (`$dsn_notify`) sert à demander des accusés pour différents résultats (comme envoi non réussi, message délivré, etc.). La variable 6.3.38 (`$dsn_return`) contrôle la quantité de votre message devant être retournée avec l’accusé (en-têtes ou message complet). Veuillez vous référer au manuel de sendmail pour plus de détails sur DSN.

## 4.10 Support de POP3 (OPTIONNEL)

Si Mutt a été compilé avec le support du protocole POP3 (avec l’option `-enable-pop` du script *configure*), il est alors capable de travailler avec des boîtes aux lettres situées sur un serveur POP3 distant et de rapatrier des courriers pour consultation locale.

Vous pouvez accéder à la boîte POP3 distante en sélectionnant le dossier `pop://serveurpop/`

Vous pouvez préciser le numéro de port à utiliser sur le serveur avec la syntaxe suivante : `pop://serveurpop:port/`

Vous pouvez également spécifier un nom d’utilisateur différent pour chaque dossier de cette manière : `pop://utilisateur@serveurpop[:port]/`

Il est plus coûteux de vérifier la présence de nouveaux courriers par POP3 que localement. Pour cette raison, il est possible de contrôler la fréquence à laquelle Mutt va interroger le serveur distant à l’aide de la variable 6.3.158 (`$pop_checkinterval`) (par défaut toutes les 60 secondes).

Si Mutt a été compilé avec le support de SSL (avec l’option `-with-ssl` du script *configure*), les connexions aux serveurs POP3 peuvent être chiffrées. Cela requiert naturellement que le serveur supporte les connexions chiffrées par SSL. Pour accéder à un dossier avec POP3/SSL, utilisez le préfixe `pops:`, ie : `pops://[utilisateur@]serveurpop[:port]/`

Un autre moyen d’accéder à vos courriers par POP3 est la fonction *fetch-mail* (raccourci par défaut : `G`), qui permet de se connecter à un serveur POP (défini par 6.3.160 (`pop_host`)) et de rapatrier tous vos nouveaux courrier vers le spoule local (6.3.213 (`spoolfile`)). Ensuite, Mutt fonctionne exactement comme si le courrier avait toujours été local.

**Remarque :** si votre seul besoin est de rapatrier tous vos messages vers des boîtes locales, vous devriez considérer l’utilisation de programmes spécifiquement conçus pour cette tâche, comme *fetchmail*.

## 4.11 Support d’IMAP (OPTIONNEL)

Si Mutt a été compilé avec le support d’IMAP (avec l’option `-enable-imap` du script *configure*), il est alors capable d’accéder à des dossiers situés sur un serveur IMAP distant.

Vous pouvez accéder à la boîte aux lettres distante en sélectionnant le dossier `imap://serveurimap/INBOX`, où `serveurimap` est le nom du serveur IMAP et `INBOX` est le nom particulier de votre boîte aux lettre courante (votre spoule) sur le serveur IMAP. Si vous souhaitez accéder un autre dossier sur le serveur IMAP, utilisez `imap://serveurimap/chemin/vers/dossier` où `chemin/vers/dossier` est l’emplacement du dossier auquel vous voulez accéder.

Vous pouvez forcer le numéro de port en le précisant à la suite du nom du serveur, ie : `imap://serveurimap:port/INBOX`

Vous pouvez également spécifier un nom d'utilisateur différent pour chaque dossier, comme ceci :  
`imap://utilisateur@serveurimap[:port]/INBOX`

Si Mutt a été compilé avec le support de SSL (avec l'option `-with-ssl` du script `configure`), les connexions IMAP peuvent être chiffrées, à la condition naturelle que les serveurs IMAP supportent les connexions chiffrées par SSL. L'accès à un dossier par IMAP/SSL se fait en utilisant la syntaxe suivante :  
`imaps://[utilisateur@]serveurimap[:port]/chemin/vers/dossier.`

Une syntaxe compatible avec le logiciel de messagerie Pine peut également être utilisée, ie :  
`{[utilisateur@]serveurimap[:port] [/ssl]}chemin/vers/dossier`

Notez que tous les serveurs n'utilisent pas / comme séparateur hiérarchique. Mutt parvient normalement à détecter le séparateur en usage sur le serveur et à convertir les chemins en conséquence.

Lorsque vous naviguez entre les dossiers d'un serveur IMAP, vous pouvez contrôler si vous voulez afficher tous les dossiers ou seulement ceux auxquels vous êtes inscrits, à l'aide de la commande `toggle-subscribed`. Voir également la variable 6.3.74 (`$imap_list_subscribed`).

L'interrogation d'un serveur IMAP pour vérifier la présence de nouveaux courriers peut causer des délais perceptibles. Vous souhaitez donc paramétrer minutieusement les variables 6.3.87 (`$mail_check`) et 6.3.223 (`$timeout`). Personnellement, avec

```
set mail_check=90
set timeout=15
```

j'obtiens de relativement bons résultats sur ma lente connexion RTC.

Notez que si le format de stockage mbox est utilisé sur un serveur UW antérieur à la version v12.250, il a été signalé que ce serveur déconnectait le client de messagerie si un autre client sélectionne le même dossier.

#### 4.11.1 Le sélecteur de dossiers

Depuis la version 1.2, Mutt supporte la navigation entre les boîtes aux lettres d'un serveur IMAP. C'est dans les grandes lignes similaire au sélecteur de dossiers locaux, avec les différences suivantes :

- À la place des droits de fichiers, Mutt affiche la chaîne de caractères "IMAP", éventuellement suivie du symbole "+" indiquant que l'entrée contient à la fois des courriers et des sous-dossiers. C'est souvent le cas sur des serveurs de type Cyrus.
- Dans le cas où l'entrée peut contenir à la fois des courriers et des sous-dossiers, la touche de sélection (`entrée` par défaut) fera descendre dans la vue des sous-dossiers. Si vous souhaitez voir les messages du dossier, vous devez employer `view-file` (touche `espace` du clavier par défaut) à la place.
- Vous pouvez supprimer des boîtes aux lettres avec la commande `delete-mailbox` (touche par défaut : `d`). Vous pouvez également vous inscrire (`subscribe`) et vous désinscrire (`unsubscribe`) à des boîtes aux lettres (en principe ces commandes sont liées respectivement aux touches `s` et `u`).

#### 4.11.2 Authentification

Mutt supporte quatre méthode d'authentification auprès des serveurs IMAP : SASL, GSSAPI, CRAM-MD5, et LOGIN (il existe un patch de Grant Edwards ajoutant le support de l'authentification NTLM pour vous autres, malheureux utilisateurs d'Exchange, mais il reste à intégrer dans la distribution officielle de Mutt). Il y a également un support pour le pseudo-protocole ANONYMOUS, qui vous permet de vous connecter à un serveur IMAP public sans disposer d'un compte sur celui-ci. Pour utiliser ANONYMOUS, utilisez simplement le nom d'utilisateur "anonymous" ou bien pas de nom du tout.

SASL est un "super-authentifiant" spécial, qui choisit parmi plusieurs protocoles (dont GSSAPI, CRAM-MD5, ANONYMOUS et DIGEST-MD5) le plus sécurisé disponible à la fois sur le serveur et votre machine cliente. En utilisant certaines de ces méthodes (comme DIGEST-MD5 et peut-être GSSAPI), l'intégralité de votre session sera chiffrée et invisible à ces foisonnants sniffeurs de réseaux. C'est la meilleure option, si vous en disposez. Pour l'utiliser, vous devez avoir la bibliothèque Cyrus SASL installée sur votre système, et Mutt doit avoir été compilé avec l'option `-with-sasl`.

Mutt essaiera tour à tour les méthodes compilées dans Mutt et disponibles sur le serveur dans l'ordre suivant : SASL, ANONYMOUS, GSSAPI, CRAM-MD5, LOGIN.

Quelques variables sont disponibles pour contrôler l'authentification :

- **6.3.79** (`$imap_user`) - spécifie le nom d'utilisateur sous lequel vous vous authentifierez auprès du serveur IMAP. Si un nom d'utilisateur est explicitement mentionné dans le chemin de la boîte aux lettres (ie une boîte de la forme `{utilisateur@serveur}`), c'est ce dernier qui sera utilisé à la place.
- **6.3.75** (`$imap_pass`) - un mot de passe que vous pouvez indiquer, et qui sera utilisé par toutes les méthodes d'authentification en ayant besoin.
- **6.3.69** (`$imap_authenticators`) - une liste d'éléments séparés par deux-points (":") des méthodes d'authentification à essayer dans l'ordre indiqué. Si cette variable est spécifiée, cela remplace le comportement par défaut de Mutt (tout essayer dans l'ordre sus-mentionné).

#### 4.12 Gérer plusieurs comptes IMAP/POP (OPTIONNEL)

S'il se trouve que vous possédez des comptes sur plusieurs serveurs IMAP et/ou POP, il se peut que vous trouviez la gestion des paramètres d'authentification peu pratique et sujette à erreurs. La commande `account-hook` peut dans ce cas s'avérer utile. Ce "hook" (comportement contextuel) fonctionne comme un `folder-hook`, mais est invoqué chaque fois que vous accédez à une boîte aux lettres distante (également à l'intérieur du sélecteur de dossiers), et non seulement lorsque vous ouvrez la boîte.

Quelques exemples :

```
account-hook . 'unset imap_user; unset imap_pass; unset tunnel'
account-hook imap://serveur1/ 'set imap_user=moi1 imap_pass=foo'
account-hook imap://serveur2/ 'set tunnel="ssh host2 /usr/libexec/imapd"'
```

#### 4.13 Démarrer un navigateur web sur les URLs (EXTERNE)

Si un courrier contient des URLs (*unified resource locator* = adresse sur le web comme `http://www.mutt.org/`), il est pratique de disposer d'un menu présentant toutes ces URLs et permettant de démarrer un navigateur web sur l'une d'elles. Cette fonctionnalité est proposée par le programme externe `urlview` qui peut être trouvé à l'adresse <ftp://ftp.guug.de/pub/mutt/contrib/>, et par la configuration suivante :

```
macro index \cb |urlview\n
macro pager \cb |urlview\n
```

## 5 Mutt's MIME Support

Quite a bit of effort has been made to make Mutt the premier text-mode MIME MUA. Every effort has been made to provide the functionality that the discerning MIME user requires, and the conformance to the

standards wherever possible. When configuring Mutt for MIME, there are two extra types of configuration files which Mutt uses. One is the `mime.types` file, which contains the mapping of file extensions to IANA MIME types. The other is the `mailcap` file, which specifies the external commands to use for handling specific MIME types.

## 5.1 Using MIME in Mutt

There are three areas/menus in Mutt which deal with MIME, they are the pager (while viewing a message), the attachment menu and the compose menu.

### 5.1.1 Viewing MIME messages in the pager

When you select a message from the index and view it in the pager, Mutt decodes the message to a text representation. Mutt internally supports a number of MIME types, including `text/plain`, `text/enriched`, `message/rfc822`, and `message/news`. In addition, the export controlled version of Mutt recognizes a variety of PGP MIME types, including PGP/MIME and `application/pgp`.

Mutt will denote attachments with a couple lines describing them. These lines are of the form:

```
[-- Attachment #1: Description --]
[-- Type: text/plain, Encoding: 7bit, Size: 10000 --]
```

Where the `Description` is the description or filename given for the attachment, and the `Encoding` is one of `7bit/8bit/quoted-printable/base64/binary`.

If Mutt cannot deal with a MIME type, it will display a message like:

```
[-- image/gif is unsupported (use 'v' to view this part) --]
```

### 5.1.2 The Attachment Menu

The default binding for `view-attachments` is `'v'`, which displays the attachment menu for a message. The attachment menu displays a list of the attachments in a message. From the attachment menu, you can save, print, pipe, delete, and view attachments. You can apply these operations to a group of attachments at once, by tagging the attachments and by using the “tag-prefix” operator. You can also reply to the current message from this menu, and only the current attachment (or the attachments tagged) will be quoted in your reply. You can view attachments as text, or view them using the mailcap viewer definition.

Finally, you can apply the usual message-related functions (like [2.3.4](#) (resend-message), and the reply and forward functions) to attachments of type `message/rfc822`.

See the help on the attachment menu for more information.

### 5.1.3 The Compose Menu

The compose menu is the menu you see before you send a message. It allows you to edit the recipient list, the subject, and other aspects of your message. It also contains a list of the attachments of your message, including the main body. From this menu, you can print, copy, filter, pipe, edit, compose, review, and rename an attachment or a list of tagged attachments. You can also modifying the attachment information, notably the type, encoding and description.

Attachments appear as follows:

```
- 1 [text/plain, 7bit, 1K]           /tmp/mutt-euler-8082-0 <no description>
  2 [application/x-gzip, base64, 422K] ~/src/mutt-0.85.tar.gz <no description>
```

The `'-'` denotes that Mutt will delete the file after sending (or postponing, or cancelling) the message. It can be toggled with the `toggle-unlink` command (default: `u`). The next field is the MIME content-type, and can be changed with the `edit-type` command (default: `^T`). The next field is the encoding for the attachment, which allows a binary message to be encoded for transmission on 7bit links. It can be changed with the `edit-encoding` command (default: `^E`). The next field is the size of the attachment, rounded to kilobytes or megabytes. The next field is the filename, which can be changed with the `rename-file` command (default: `R`). The final field is the description of the attachment, and can be changed with the `edit-description` command (default: `d`).

## 5.2 MIME Type configuration with `mime.types`

When you add an attachment to your mail message, Mutt searches your personal `mime.types` file at `${HOME}/.mime.types`, and then the system `mime.types` file at `/usr/local/share/mutt/mime.types` or `/etc/mime.types`

The `mime.types` file consist of lines containing a MIME type and a space separated list of extensions. For example:

```
application/postscript      ps eps
application/pgp             pgp
audio/x-aiff                aif aifc aiff
```

A sample `mime.types` file comes with the Mutt distribution, and should contain most of the MIME types you are likely to use.

If Mutt can not determine the mime type by the extension of the file you attach, it will look at the file. If the file is free of binary information, Mutt will assume that the file is plain text, and mark it as `text/plain`. If the file contains binary information, then Mutt will mark it as `application/octet-stream`. You can change the MIME type that Mutt assigns to an attachment by using the `edit-type` command from the compose menu (default: `^T`). The MIME type is actually a major mime type followed by the sub-type, separated by a `'/'`. 6 major types: `application`, `text`, `image`, `video`, `audio`, and `model` have been approved after various internet discussions. Mutt recognises all of these if the appropriate entry is found in the `mime.types` file. It also recognises other major mime types, such as the chemical type that is widely used in the molecular modelling community to pass molecular data in various forms to various molecular viewers. Non-recognised mime types should only be used if the recipient of the message is likely to be expecting such attachments.

## 5.3 MIME Viewer configuration with `mailcap`

Mutt supports RFC 1524 MIME Configuration, in particular the Unix specific format specified in Appendix A of RFC 1524. This file format is commonly referred to as the `mailcap` format. Many MIME compliant programs utilize the `mailcap` format, allowing you to specify handling for all MIME types in one place for all programs. Programs known to use this format include Netscape, XMOsaic, lynx and metmail.

In order to handle various MIME types that Mutt can not handle internally, Mutt parses a series of external configuration files to find an external handler. The default search string for these files is a colon delimited list set to

```
${HOME}/.mailcap:/usr/local/share/mutt/mailcap:/etc/mailcap:/etc/mailcap:/usr/etc/mailcap:/usr/local/etc/m
```

where `$HOME` is your home directory.

In particular, the metacmail distribution will install a mailcap file, usually as `/usr/local/etc/mailcap`, which contains some baseline entries.

### 5.3.1 The Basics of the mailcap file

A mailcap file consists of a series of lines which are comments, blank, or definitions.

A comment line consists of a `#` character followed by anything you want.

A blank line is blank.

A definition line consists of a content type, a view command, and any number of optional fields. Each field of a definition line is divided by a semicolon `;` character.

The content type is specified in the MIME standard type/subtype method. For example, `text/plain`, `text/html`, `image/gif`, etc. In addition, the mailcap format includes two formats for wildcards, one using the special `'*'` subtype, the other is the implicit wild, where you only include the major type. For example, `image/*`, or `video`, will match all image types and video types, respectively.

The view command is a Unix command for viewing the type specified. There are two different types of commands supported. The default is to send the body of the MIME message to the command on stdin. You can change this behaviour by using `%s` as a parameter to your view command. This will cause Mutt to save the body of the MIME message to a temporary file, and then call the view command with the `%s` replaced by the name of the temporary file. In both cases, Mutt will turn over the terminal to the view program until the program quits, at which time Mutt will remove the temporary file if it exists.

So, in the simplest form, you can send a `text/plain` message to the external pager `more` on stdin:

```
text/plain; more
```

Or, you could send the message as a file:

```
text/plain; more %s
```

Perhaps you would like to use `lynx` to interactively view a `text/html` message:

```
text/html; lynx %s
```

In this case, `lynx` does not support viewing a file from stdin, so you must use the `%s` syntax. **Note:** *Some older versions of lynx contain a bug where they will check the mailcap file for a viewer for text/html. They will find the line which calls lynx, and run it. This causes lynx to continuously spawn itself to view the object.*

On the other hand, maybe you don't want to use `lynx` interactively, you just want to have it convert the `text/html` to `text/plain`, then you can use:

```
text/html; lynx -dump %s | more
```

Perhaps you wish to use `lynx` to view `text/html` files, and a pager on all other text formats, then you would use the following:

```
text/html; lynx %s
text/*; more
```

This is the simplest form of a mailcap file.

### 5.3.2 Secure use of mailcap

The interpretation of shell meta-characters embedded in MIME parameters can lead to security problems in general. Mutt tries to quote parameters in expansion of %s syntaxes properly, and avoids risky characters by substituting them, see the 6.3.89 (mailcap\_sanitize) variable.

Although mutt's procedures to invoke programs with mailcap seem to be safe, there are other applications parsing mailcap, maybe taking less care of it. Therefore you should pay attention to the following rules:

*Keep the %-expandos away from shell quoting.* Don't quote them with single or double quotes. Mutt does this for you, the right way, as should any other program which interprets mailcap. Don't put them into backtick expansions. Be highly careful with eval statements, and avoid them if possible at all. Trying to fix broken behaviour with quotes introduces new leaks - there is no alternative to correct quoting in the first place.

If you have to use the %-expandos' values in context where you need quoting or backtick expansions, put that value into a shell variable and reference the shell variable where necessary, as in the following example (using \$charset inside the backtick expansion is safe, since it is not itself subject to any further expansion):

```
text/test-mailcap-bug; cat %s; copiousoutput; test=charset=%{charset} \
  && test " `echo $charset | tr ' [A-Z]' '[a-z]' " != iso-8859-1
```

### 5.3.3 Advanced mailcap Usage

**Optional Fields** In addition to the required content-type and view command fields, you can add semi-colon ';' separated fields to set flags and other options. Mutt recognizes the following optional fields:

#### copiousoutput

This flag tells Mutt that the command passes possibly large amounts of text on stdout. This causes Mutt to invoke a pager (either the internal pager or the external pager defined by the pager variable) on the output of the view command. Without this flag, Mutt assumes that the command is interactive. One could use this to replace the pipe to more in the lynx -dump example in the Basic section:

```
text/html; lynx -dump %s ; copiousoutput
```

This will cause lynx to format the text/html output as text/plain and Mutt will use your standard pager to display the results.

#### needsterminal

Mutt uses this flag when viewing attachments with 5.4 (autoview), in order to decide whether it should honor the setting of the 6.3.233 (\$wait\_key) variable or not. When an attachment is viewed using an interactive program, and the corresponding mailcap entry has a needsterminal flag, Mutt will use 6.3.233 (\$wait\_key) and the exit status of the program to decide if it will ask you to press a key after the external program has exited. In all other situations it will not prompt you for a key.

#### compose=<command>

This flag specifies the command to use to create a new attachment of a specific MIME type. Mutt supports this from the compose menu.

#### composetyped=<command>

This flag specifies the command to use to create a new attachment of a specific MIME type. This command differs from the compose command in that mutt will expect standard MIME headers on the data. This can be used to specify parameters, filename, description, etc. for a new attachment. Mutt supports this from the compose menu.

**print=<command>**

This flag specifies the command to use to print a specific MIME type. Mutt supports this from the attachment and compose menus.

**edit=<command>**

This flag specifies the command to use to edit a specific MIME type. Mutt supports this from the compose menu, and also uses it to compose new attachments. Mutt will default to the defined editor for text attachments.

**nametemplate=<template>**

This field specifies the format for the file denoted by %s in the command fields. Certain programs will require a certain file extension, for instance, to correctly view a file. For instance, lynx will only interpret a file as `text/html` if the file ends in `.html`. So, you would specify lynx as a `text/html` viewer with a line in the mailcap file like:

```
text/html; lynx %s; nametemplate=%s.html
```

**test=<command>**

This field specifies a command to run to test whether this mailcap entry should be used. The command is defined with the command expansion rules defined in the next section. If the command returns 0, then the test passed, and Mutt uses this entry. If the command returns non-zero, then the test failed, and Mutt continues searching for the right entry. **Note:** *the content-type must match before Mutt performs the test.* For example:

```
text/html; netscape -remote 'openURL(%s)' ; test=RunningX
text/html; lynx %s
```

In this example, Mutt will run the program RunningX which will return 0 if the X Window manager is running, and non-zero if it isn't. If RunningX returns 0, then Mutt will call netscape to display the `text/html` object. If RunningX doesn't return 0, then Mutt will go on to the next entry and use lynx to display the `text/html` object.

**Search Order** When searching for an entry in the mailcap file, Mutt will search for the most useful entry for its purpose. For instance, if you are attempting to print an `image/gif`, and you have the following entries in your mailcap file, Mutt will search for an entry with the print command:

```
image/*;          xv %s
image/gif;        ; print= anytopnm %s | pnmtops | lpr; \
                  nametemplate=%s.gif
```

Mutt will skip the `image/*` entry and use the `image/gif` entry with the print command.

In addition, you can use this with 5.4 (Autoview) to denote two commands for viewing an attachment, one to be viewed automatically, the other to be viewed interactively from the attachment menu. In addition, you can then use the test feature to determine which viewer to use interactively depending on your environment.

```
text/html;        netscape -remote 'openURL(%s)' ; test=RunningX
text/html;        lynx %s; nametemplate=%s.html
text/html;        lynx -dump %s; nametemplate=%s.html; copiousoutput
```

For 5.4 (Autoview), Mutt will choose the third entry because of the `copiousoutput` tag. For interactive viewing, Mutt will run the program RunningX to determine if it should use the first entry. If the program returns non-zero, Mutt will use the second entry for interactive viewing.

**Command Expansion** The various commands defined in the mailcap files are passed to the `/bin/sh` shell using the `system()` function. Before the command is passed to `/bin/sh -c`, it is parsed to expand various special parameters with information from Mutt. The keywords Mutt expands are:

`%s`

As seen in the basic mailcap section, this variable is expanded to a filename specified by the calling program. This file contains the body of the message to view/print/edit or where the composing program should place the results of composition. In addition, the use of this keyword causes Mutt to not pass the body of the message to the view/print/edit program on stdin.

`%t`

Mutt will expand `%t` to the text representation of the content type of the message in the same form as the first parameter of the mailcap definition line, ie `text/html` or `image/gif`.

`%{<parameter>}`

Mutt will expand this to the value of the specified parameter from the Content-Type: line of the mail message. For instance, if Your mail message contains:

```
Content-Type: text/plain; charset=iso-8859-1
```

then Mutt will expand `%{charset}` to `iso-8859-1`. The default metamail mailcap file uses this feature to test the charset to spawn an xterm using the right charset to view the message.

`\%`

This will be replaced by a `%`

Mutt does not currently support the `%F` and `%n` keywords specified in RFC 1524. The main purpose of these parameters is for multipart messages, which is handled internally by Mutt.

#### 5.3.4 Example mailcap files

This mailcap file is fairly simple and standard:

---

```
# I'm always running X :)
video/*;      xanim %s > /dev/null
image/*;      xv %s > /dev/null

# I'm always running netscape (if my computer had more memory, maybe)
text/html;    netscape -remote 'openURL(%s)'
```

---

This mailcap file shows quite a number of examples:

---

```
# Use xanim to view all videos  Xanim produces a header on startup,
# send that to /dev/null so I don't see it
video/*;      xanim %s > /dev/null

# Send html to a running netscape by remote
text/html;    netscape -remote 'openURL(%s)'; test=RunningNetscape

# If I'm not running netscape but I am running X, start netscape on the
# object
```

```

text/html;      netscape %s; test=RunningX

# Else use lynx to view it as text
text/html;      lynx %s

# This version would convert the text/html to text/plain
text/html;      lynx -dump %s; copiousoutput

# enriched.sh converts text/enriched to text/html and then uses
# lynx -dump to convert it to text/plain
text/enriched;  enriched.sh ; copiousoutput

# I use enscript to print text in two columns to a page
text/*;         more %s; print=enscript -2Gr %s

# Netscape adds a flag to tell itself to view jpegs internally
image/jpeg;xv %s; x-mozilla-flags=internal

# Use xv to view images if I'm running X
# In addition, this uses the \ to extend the line and set my editor
# for images
image/*;xv %s; test=RunningX; \
    edit=xpaint %s

# Convert images to text using the netpbm tools
image/*; (anytopnm %s | pnmscale -ysize 80 46 | ppmtopgm | pgmentopbm |
pbmtoascii -lx2 ) 2>&1 ; copiousoutput

# Send excel spreadsheets to my NT box
application/ms-excel; open.pl %s

```

---

## 5.4 MIME Autoview

In addition to explicitly telling Mutt to view an attachment with the MIME viewer defined in the mailcap file, Mutt has support for automatically viewing MIME attachments while in the pager.

To work, you must define a viewer in the mailcap file which uses the `copiousoutput` option to denote that it is non-interactive. Usually, you also use the entry to convert the attachment to a text representation which you can view in the pager.

You then use the `auto_view` muttrc command to list the content-types that you wish to view automatically.

For instance, if you set `auto_view` to:

```
auto_view text/html text/enriched application/x-gunzip application/postscript image/gif application/x-tar-
```

Mutt could use the following mailcap entries to automatically view attachments of these types.

```

text/html;      lynx -dump %s; copiousoutput; nametemplate=%s.html
text/enriched;  enriched.sh ; copiousoutput
image/*;        anytopnm %s | pnmscale -xsize 80 -ysize 50 | ppmtopgm | pgmentopbm | pbmtoascii ; copiousout
application/x-gunzip;  gzcat; copiousoutput

```

```
application/x-tar-gz; gunzip -c %s | tar -tf - ; copiousoutput
application/postscript; ps2ascii %s; copiousoutput
```

## 5.5 MIME Multipart/Alternative

Mutt has some heuristics for determining which attachment of a multipart/alternative type to display. First, mutt will check the `alternative_order` list to determine if one of the available types is preferred. The `alternative_order` list consists of a number of mimetypes in order, including support for implicit and explicit wildcards, for example:

```
alternative_order text/enriched text/plain text application/postscript image/*
```

Next, mutt will check if any of the types have a defined [5.4](#) (`auto_view`), and use that. Failing that, Mutt will look for any text type. As a last attempt, mutt will look for any type it knows how to handle.

# 6 Référence

## 6.1 Options de la ligne de commande

Si Mutt est lancé sans arguments, il tentera de lire votre spoule de courriers. Il est cependant également possible de lire d'autres boîtes et d'envoyer des messages à partir de la ligne de commande.

```
-a      attache un fichier à un message
-c      spécifie une adresse de copie carbone (Cc)
-e      spécifie une commande de configuration à exécuter après la lecture des fichiers d'initialisation
-F      spécifie un autre fichier pour les commandes d'initialisation
-f      spécifie une boîte aux lettres à charger
-h      affiche l'aide sur les options de ligne de commande
-H      spécifie un fichier brouillon/modèle dans lequel lire des en-têtes et un corps de message
-i      spécifie un fichier à inclure dans la rédaction d'un message
-n      ne pas lire le fichier de configuration globale Muttrc
-m      spécifie un format de boîte à lettre par défaut
-p      rappelle un message non envoyé
-R      ouvre une boîte à lettre en lecture seule
-s      spécifie un objet (le mettre entre guillemets s'il contient des espaces)
-v      affiche le numéro de version ainsi que les options de compilation
-x      simule le mode de fonctionnement de mailx(1)
-y      affiche un menu composé des fichiers spécifiés par la commande mailboxes

-z      quitte immédiatement s'il n'y a pas de nouveaux message dans la boîte
-Z      ouvre la première boîte contenant de nouveaux messages, quitte immédiatement s'il n'y en a pas
```

Pour lire des messages dans une boîte aux lettres

```
mutt [-nz] [-F muttrc] [-m type] [-f boîte]
```

Pour composer un nouveau message

```
mutt [-n] [-F muttrc] [-a fichier] [-c adresse] [-i fichier] [-s objet] adresse [ adresse ... ]
```

Mutt possède également un mode "batch" permettant d'envoyer des messages préparés. Il suffit de rediriger l'entrée à partir du fichier que vous souhaitez envoyer. Par exemple,

```
mutt -s "données de l'exécution #2" professeur@universite.fr < ~/run2.dat
```

Cette commande enverra un message à “professeur@universite.fr” avec l’objet “données de l’exécution #2”. Le corps du message sera constitué du contenu du fichier “~/run2.dat”.

## 6.2 Commandes de configuration

Ci-dessous figurent les commandes reconnues par mutt.

- 4.12 (account-hook) *motif commande*
- 3.2 (alias) *touche adresse [ , adresse, ... ]*
- 3.2 (unalias) [*\* | touche ...*]
- 5.5 (alternative\_order) *typemime [ typemime ... ]*
- 5.4 (auto\_view) *typemime [ typemime ... ]*
- 3.3 (bind) *jeu touche fonction*
- 3.4 (charset-hook) *alias jeu-de-caractères*
- 3.7 (color) *objet premier-plan arrière-plan [ expr-rationnelle ]*
- 3.7 (uncolor) *index motif [ motif ... ]*
- 3.21 (exec) *fonction [ fonction ... ]*
- 3.15 (fcc-hook) *motif boîte*
- 3.16 (fcc-save-hook) *motif boîte*
- 3.5 (folder-hook) *motif commande*
- 3.13 (hdr\_order) *en-tête [ en-tête ... ]*
- 3.13 (unhdr\_order) *en-tête [ en-tête ... ]*
- 3.4 (iconv-hook) *jeu-de-caractères jeu-de-caractères-local*
- 3.8 (ignore) *motif [ motif ... ]*
- 3.8 (unignore) *motif [ motif ... ]*
- 3.9 (lists) *adresse [ adresse ... ]*
- 3.9 (unlists) *adresse [ adresse ... ]*
- 3.6 (macro) *menu touche séquence [ description ]*
- 3.11 (mailboxes) *fichier [ fichier ... ]*
- 3.10 (mbox-hook) *motif boîte*
- 3.18 (message-hook) *motif commande*
- 3.7 (mono) *objet attribut [ expr-rationnelle ]*
- 3.7 (unmono) *index motif [ motif ... ]*
- 3.12 (my\_hdr) *chaîne*
- 3.12 (unmy\_hdr) *champ [ champ ... ]*

- [3.19](#) (pgp-hook) *motif id-de-clé*
- [3.20](#) (push) *chaîne*
- [3.23](#) (reset) *variable [variable ... ]*
- [3.14](#) (save-hook) *expr-rationnelle fichier*
- [3.22](#) (score) *motif valeur*
- [3.22](#) (unscore) *motif [ motif ... ]*
- [3.17](#) (send-hook) *expr-rationnelle commande*
- [3.23](#) (set) [no|inv]*variable[=valeur] [ variable ... ]*
- [3.23](#) (unset) *variable [variable ... ]*
- [3.24](#) (source) *fichier*
- [3.9](#) (subscribe) *adresse [ adresse ... ]*
- [3.9](#) (unsubscribe) *adresse [ adresse ... ]*
- [3.23](#) (toggle) *variable [variable ... ]*
- [3.25](#) (unhook) *type-hook*

## 6.3 Configuration variables

### 6.3.1 abort\_nosubject

Type: quadri-option

Valeur par défaut: ask-yes

Quand cette variable est positionnée à *yes*, la composition du message sera abandonnée si aucun objet n'est donné à l'invite. La composition du message ne sera en revanche jamais abandonnée (même si aucun objet n'a été spécifié) si cette variable est positionnée à *no*.

### 6.3.2 abort\_unmodified

Type: quadri-option

Valeur par défaut: yes

Quand cette variable est positionnée à *yes*, la composition est automatiquement abandonnée après l'édition du corps du message si aucun changement n'a été apporté au fichier. (cette vérification n'est effectuée qu'après la *première* édition du fichier). Si cette variable est positionnée à *no*, la composition n'est jamais abandonnée.

### 6.3.3 alias\_file

Type: chemin

Valeur par défaut: "~/muttrc"

Le fichier par défaut dans lequel sont sauves les alias créés par la fonction "[2.3.4](#) (create-alias)".

**Remarque:** Mutt ne lira pas ce fichier automatiquement; vous devez utiliser explicitement la commande "[3.24](#) (source)" pour cela.

### 6.3.4 alias\_format

Type: chaîne

Valeur par défaut: "%4n %2f %t %-10a %r"

Spécifie le format des données affichées dans le menu 'alias'. Les séquences de format de style printf(3) suivantes sont disponibles :

**%a**

alias

**%f**

drapeaux - actuellement, uniquement "d" (alias marqué pour suppression)

**%n**

numéro dans l'index

**%r**

adresse correspondant à l'alias

**%t**

caractère indiquant si l'alias est marqué pour inclusion

### 6.3.5 allow\_8bit

Type: booléen

Valeur par défaut: yes

Contrôle si les données 8 bits sont converties en données 7 bits à l'aide de l'encodage Quoted-Printable ou Base64 lors de l'envoi d'un courrier.

### 6.3.6 allow\_ansi

Type: booléen

Valeur par défaut: no

Contrôle si les codes de couleur ANSI figurant dans les messages (et les balises de couleur pour les messages en texte enrichi) doivent être interprétés. Les messages contenant ces codes sont rares, mais si cette option est positionnée, leur texte sera coloré en conséquence. Notez que ceci peut outrepasser vos propres choix de couleur, et même occasionner un problème de sécurité, étant donné qu'un message pourrait inclure une ligne telle que "[ - La sortie PGP suit ..." et lui donner la même couleur que votre couleur pour les attachements.

### 6.3.7 alternates

Type: expression rationnelle

Valeur par défaut: ""

Une expression rationnelle permettant de spécifier les *différentes* adresses auxquelles vous recevez des courriers. Cela a des répercussions sur l'idée que Mutt se fait des messages que vous envoyez et recevez.

### 6.3.8 arrow\_cursor

Type: booléen

Valeur par défaut: no

Quand cette variable est positionnée, une flèche (“->”) est utilisée pour désigner l’entrée courante dans les menus au lieu d’inverser la ligne entière. Cela augmente la réactivité pour des connections réseau lentes ou par modem, car il y a moins à ré-afficher lors d’un déplacement vers l’entrée suivante ou précédente du menu.

### 6.3.9 ascii\_chars

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, Mutt utilisera uniquement des caractères ASCII pour afficher les fils de discussion et la représentation arborescente des attachements, à la place des caractères *ACS* utilisés par défaut.

### 6.3.10 askbcc

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, Mutt vous demandera une liste de destinataires à mettre en copie aveugle (Bcc) avant la composition du corps d’un message.

### 6.3.11 askcc

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, Mutt vous demandera une liste de destinataires à mettre en copie carbone (Cc) avant la composition du corps d’un message.

### 6.3.12 attach\_format

Type: chaîne

Valeur par défaut: "%u%D%I %t%4n %T%.40d%> [%.7m/%.10M, %.6e%?C?, %C?, %s] "

Cette variable spécifie le format du menu ‘attachement’. Les séquences de format de style printf() suivantes sont reconnues :

%D

marqué pour suppression

%d

description

%e

encodage (content-transfer-encoding) MIME

%f

nom de fichier

**%I**

disposition (I=en ligne, A=comme attachement)

**%m**

type MIME principal

**%M**

type MIME secondaire

**%n**

numéro d'attachement

**%s**

taille

**%t**

marqué

**%u**

fichier marqué pour suppression

**%>X**

justifier à droite le reste de la chaîne et remplir l'intervalle avec le caractère "X"

**%|X**

remplir jusqu'en fin de ligne avec le caractère "X"

### 6.3.13 `attach_sep`

Type: chaîne

Valeur par défaut: "\n"

Le séparateur à ajouter entre les attachements lors d'une opération (sauvegarde, passage à une commande, etc) sur une liste d'attachements marqués.

### 6.3.14 `attach_split`

Type: booléen

Valeur par défaut: yes

Lors d'une opération (sauvegarde, passage à une commande, etc) sur une liste d'attachements marqués, si cette variable est positionnée, Mutt opérera successivement sur chaque attachement. Autrement, il concatènera les attachements et les traitera d'un seul bloc, comme s'il s'agissait d'un unique attachement. Dans ce cas, le séparateur "6.3.13 (`$attach_sep`)" est ajouté à la suite de chaque attachement.

### 6.3.15 `attribution`

Type: chaîne

Valeur par défaut: "On %d, %n wrote:"

### 6.3.16 autoedit

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée en même temps que “6.3.40 (\$edit.headers)”, Mutt sautera le menu d’envoi initial, permettant ainsi de composer immédiatement le corps de votre message. Il est possible d’accéder au menu d’envoi après avoir composé votre message.

Voir également “6.3.45 (\$fast\_reply)”.

### 6.3.17 auto\_tag

Type: booléen

Valeur par défaut: no

Quand cette variable est positionnée, les fonctions du menu *index* opérant sur un message seront appliquées à chaque message marqué (s’il y en a). Sinon, vous devez d’abord utiliser la fonction tag-prefix (touche par défaut : “;”) pour que la prochaine fonction s’applique aux messages marqués.

### 6.3.18 beep

Type: booléen

Valeur par défaut: yes

Quand cette variable est positionnée, Mutt émet un "bip" si une erreur se produit.

### 6.3.19 beep\_new

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, Mutt émettra un "bip" pour chaque annonce qu’un nouveau courrier a été reçu. Ceci est indépendant de l’état de la variable “6.3.18 (\$beep)”.

### 6.3.20 bounce\_delivered

Type: booléen

Valeur par défaut: yes

Quand cette variable est positionnée, Mutt inclura l’en-tête *Delivered-To* lors du renvoi d’un message ("bounce"). Cette variable est susceptible d’intéresser les utilisateurs de Postfix.

### 6.3.21 charset

Type: chaîne

Valeur par défaut: ""

Le jeu de caractères utilisé pour afficher et entrer du texte.

### 6.3.22 check\_new

Type: booléen

Valeur par défaut: yes

**Note:** cette option ne concerne que les boîtes de types *maildir* et *MH*.

Quand cette variable est positionnée, Mutt vérifie si de nouveaux courriers sont arrivés alors que la boîte aux lettres est ouverte. Cette vérification peut prendre un certain temps, particulièrement avec des boîtes de type MH, car cela implique de regarder le contenu du répertoire et de regarder pour chaque fichier s'il n'a pas déjà été lu. Si *check\_new* n'est *pas* positionné, aucune vérification de l'arrivée de nouveaux courriers ne sera effectuée tant que la boîte est ouverte.

### 6.3.23 collapse\_unread

Type: booléen

Valeur par défaut: yes

Si cette variable n'est *pas* positionnée, Mutt ne repliera pas le fil d'une discussion contenant des messages non lus.

### 6.3.24 uncollapse\_jump

Type: booléen

Valeur par défaut: no

Si cette variable *est* positionnée, Mutt ira au prochain message non lu (s'il y en a un) quand le fil de discussion courant n'est *pas* replié.

### 6.3.25 compose\_format

Type: chaîne

Valeur par défaut: "-- Mutt: Compose [Approx. msg size: %l Atts: %a]%>-"

Contrôle le format de la ligne d'état affichée dans le menu *Compose*\fp. Cette chaîne est similaire à "[6.3.215](#) (*\$status\_format*)", mais possède son propre jeu de séquence de format de type *printf()* :

**%h**

*nom de l'hôte local*

**%l**

*taille approximative (en octets) du message courant*

**%v**

*chaîne de numéro de version de Mutt Voir le texte décrivant l'option "[6.3.215](#) (*\$status\_format*)" pour plus d'information sur la manière de configurer "[6.3.25](#) (*\$compose\_format*)".*

### 6.3.26 confirmappend

Type: booléen

Valeur par défaut: yes

Si cette variable est positionnée, Mutt attend confirmation avant d'ajouter des messages à une boîte aux lettres existante.

### 6.3.27 confirmcreate

Type: booléen

Valeur par défaut: yes

Si cette variable est positionnée, Mutt attend confirmation avant de créer une boîte à lettre inexistante pour y sauver des messages.

### 6.3.28 connect\_timeout

Type: nombre

Valeur par défaut: 30

Mutt arrêtera sa tentative d'ouvrir une connexion réseau (pour IMAP et POP) si au bout cette durée (en secondes) la connexion ne parvient pas à s'établir. Avec une valeur négative, Mutt attendra indéfiniment que la connexion s'établisse.

### 6.3.29 copy

Type: quadri-option

Valeur par défaut: yes

Cette variable contrôle si une copie des messages que vous envoyez doit être sauvée pour consultation ultérieure. Voir également “6.3.181 (\$record)”, “6.3.192 (\$save\_name)”, “6.3.51 (\$force\_name)” et “3.15 (fcc-hook)”.

### 6.3.30 date\_format

Type: chaîne

Valeur par défaut: "%a, %b %d, %Y at %I:%M:%S%p %Z"

Cette variable contrôle le format d'affichage de la date par la séquence “%d”, présente dans “6.3.83 (\$index\_format)”. La date est générée en passant cette chaîne en argument à la fonction *strftime*. Consulter la page de manuel de *strftime(3)* pour connaître la syntaxe appropriée.

Le mois et le jour de la semaine sont produits en accord avec les paramètres de localisation définis par la variable “6.3.86 (\$locale)”, à moins que le premier caractère de “6.3.30 (\$date\_format)” soit un point d'exclamation. Dans ce dernier cas, le point d'exclamation est omis et le mois ainsi que le jour de la semaine sont produits suivant la “locale” *C* (c'est-à-dire anglais US) dans le reste de la chaîne.

### 6.3.31 default\_hook

Type: chaîne

Valeur par défaut: "~f %s !~P | (~P ~C %s)"

Cette variable contrôle la manière dont les send-hooks, message-hooks, save-hooks et fcc-hooks doivent être interprétés au cas où une simple expression régulière est spécifiée à la place d'un motif. Les hooks sont compilés (transformés en motif compréhensible par Mutt) à leur déclaration; ainsi un hook sera interprété d'après sa valeur au moment où il a été défini. La valeur par défaut correspond si le message provient d'un auteur correspondant à l'expression rationnelle fournie, ou de vous-même (s'il y a correspondance entre l'adresse et “6.3.7 (\$alternates)”) et est à destination (To) ou copie carbone (Cc) d'une personne correspondant à l'expression rationnelle.

### 6.3.32 delete

Type: quadri-option

Valeur par défaut: ask-yes

Cette variable contrôle si les messages sont réellement effacés lors de la fermeture ou la synchronisation d'une boîte aux lettres. Si elle *est* positionnée, les messages marqués pour suppression seront automatiquement effacés sans demande de confirmation. Si elle n'est *pas* positionnée, les messages marqués pour suppression seront gardés dans la boîte aux lettres.

### 6.3.33 delete\_untag

Type: booléen

Valeur par défaut: yes

Si cette option *est* positionnée, Mutt enlèvera le marquage ("tag") de messages qu'il marque pour suppression. Cela a cours lorsque vous effacez explicitement un message, ou quand vous le sauvez dans un autre dossier.

### 6.3.34 digest\_collapse

Type: booléen

Valeur par défaut: yes

Si cette option est positionnée, les sous-parties de messages d'un "digest" ne seront pas affichées dans le menu attachements. Pour voir ces sous-parties, taper 'v' dans ce menu.

### 6.3.35 display\_filter

Type: chemin

Valeur par défaut: ""

Quand cette variable est positionnée, elle indique une commande qui sera utilisée comme un filtre sur les messages. Quand un message est visualisé, son contenu est dirigé vers l'entrée standard de 6.3.35 (`$display_filter`), et le message ainsi filtré est lu sur la sortie standard.

### 6.3.36 dotlock\_program

Type: chemin

Valeur par défaut: "BINDIR/mutt\_dotlock"

Contient l'emplacement du programme `mutt_dotlock` (8) utilisé par Mutt.

### 6.3.37 dsn\_notify

Type: chaîne

Valeur par défaut: ""

**Remarque :** cette variable ne devrait pas être positionnée à moins que vous n'utilisiez Sendmail (à partir de la version 8.8.x).

Cette variable contrôle le moment où la demande d'accusé de réception est renvoyée. Cette chaîne de caractères consiste en une liste d'une ou plusieurs des valeurs suivantes, séparées par des virgules (pas d'espaces !) : *never*, pour ne jamais demander d'accusé de réception, *failure*, pour demander un accusé en

cas d'échec de transmission, *delay*, pour être averti d'un retard du message, *success*, pour être informé d'une de la réussite de la transmission.

Exemple : `set dsn_notify="failure,delay"`

### 6.3.38 `dsn_return`

Type: chaîne

Valeur par défaut: ""

**Remarque :** cette variable ne devrait pas être positionnée à moins que vous n'utilisiez Sendmail (à partir de la version 8.8.x).

Cette variable contrôle comment votre courrier est renvoyé dans les messages DSN (accusés de réception). Elle peut être positionnée soit à *hdrs* pour ne renvoyer que les en-têtes, ou à  pour retourner l'intégralité du message.

Exemple : `set dsn_return=hdrs`

### 6.3.39 `duplicate_threads`

Type: booléen

Valeur par défaut: `yes`

Cette variable contrôle si Mutt, lorsqu'il affiche les fils de discussion graphiquement, doit regrouper les messages comportant le même champ d'en-tête Message-Id. Si elle est positionnée, les doublons seront indiqués par un signe égal dans l'arborescence des messages.

### 6.3.40 `edit_headers`

Type: booléen

Valeur par défaut: `no`

Cette option vous donne la possibilité d'éditer les en-têtes alors que vous rédigez le corps de votre message.

### 6.3.41 `editor`

Type: chemin

Valeur par défaut: ""

Cette variable spécifie l'éditeur de texte utilisé par Mutt. Elle prend par défaut la valeur de la variable d'environnement EDITOR ou VISUAL, ou bien la chaîne "vi".

### 6.3.42 `encode_from`

Type: booléen

Valeur par défaut: `no`

Si cette variable *est* positionnée, Mutt encodera les messages contenant la chaîne "From" en début de ligne au format "quoted- printable". Utile pour éviter les altérations des messages qu'ont tendance à produire certains logiciels d'acheminement et de transfert de courrier.

### 6.3.43 `envelope_from`

Type: booléen

Valeur par défaut: no

Quand cette variable *est* positionnée, Mutt tentera de déduire l'*enveloppe* d'expéditeur ("From ") du message d'après le champ d'en-têtes "From:". Remarquez que cette information est passée à sendmail via l'option de ligne de commande "-f". Par conséquent, ne positionnez pas 6.3.43 (`$envelope_from`) si vous utilisez déjà l'option "-f" dans 6.3.198 (`$sendmail`) ou si le sendmail de votre machine ne reconnaît pas cette option.

### 6.3.44 `escape`

Type: chaîne

Valeur par défaut: "~"

Caractère d'échappement à utiliser pour les fonctions de l'éditeur intégré.

### 6.3.45 `fast_reply`

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, la demande initiale des destinataires et de l'objet du courrier sont sautées lors d'une réponse à des messages, et la demande de l'objet est sautée si vous faites suivre des messages.

**Remarque :** cette variable n'a aucun effet si la variable "6.3.16 (`$autoedit`)" est positionnée.

### 6.3.46 `fcc_attach`

Type: booléen

Valeur par défaut: yes

Cette variable contrôle si les attachements des messages que vous envoyez doivent ou non être sauvegardés avec le corps du message.

### 6.3.47 `fcc_clear`

Type: booléen

Valeur par défaut: no

Quand cette variable est positionnée, les messages sortant (FCC) seront sauvés en clair, même si ce message est envoyé chiffré et/ou signé.

### 6.3.48 `folder`

Type: chemin

Valeur par défaut: "~/Mail"

Spécifie l'emplacement par défaut de vos boîtes aux lettres. Un '+' ou un '=' en début de chemin sera interprété d'après le contenu de cette variable. Si vous modifiez cette variable par rapport à sa valeur par défaut, assurez vous que cette affectation a bien lieu *avant* l'utilisation de '+' ou '=' dans une autre variable, étant donné que le remplacement a lieu à l'exécution de la commande 'set'.

### 6.3.49 folder\_format

Type: chaîne

Valeur par défaut: "%2C %t %N %F %l %-8.8u %-8.8g %8s %d %f"

Cette variable vous permet de personnaliser l’affichage du sélecteur de fichier suivant vos goûts. Cette chaîne est similaire à “6.3.83 (\$index\_format)”, mais possède son propre jeu de séquences de format de type printf() :

**%C**

numéro du fichier courant

**%d**

date/heure de dernière modification de la boîte

**%f**

nom du fichier

**%F**

permissions du fichiers

**%g**

nom du groupe (ou GID numérique, si le nom est absent)

**%l**

nombre de liens durs

**%N**

N si la boîte a de nouveaux courriers, sinon un blanc

**%s**

taille en octets

**%t**

\* si le fichier est marqué, sinon un blanc

**%u**

nom du propriétaire (ou UID numérique, si pas de nom)

**%>X**

justifier à droite le reste de la chaîne et remplir l’intervalle avec le caractère "X"

**%|X**

remplir la fin de la ligne avec le caractère "X"

### 6.3.50 followup\_to

Type: booléen

Valeur par défaut: yes

Contrôle si le champ d’en-têtes *Mail-Followup-To* est généré à l’envoi de courrier. Quand cette variable est positionnée, Mutt générera ce champ si vous répondez à une liste de diffusion connue, spécifiée à l’aide des commandes “subscribe” ou “lists”.

Ce champ a deux intérêts. Premièrement, vous évitez de recevoir des doublons de réponses que vous envoyez à des listes de diffusion. Le second but est de s'assurer que vous recevrez bien une réponse séparée pour chaque message envoyé à une liste connue mais à laquelle vous n'êtes pas abonné. Le champ d'en-têtes contiendra uniquement l'adresse de la liste si vous y êtes abonné, l'adresse de la liste ainsi que votre propre adresse dans le cas contraire. Sans ce champ, une "réponse à tous les expéditeurs" sera envoyée à la fois à la liste et à votre adresse, occasionnant pour vous la réception de deux copies du même courrier.

#### 6.3.51 `force_name`

Type: booléen

Valeur par défaut: no

Cette variable est similaire à "6.3.192 (`$save_name`)", à l'exception du fait que Mutt sauvera une copie de votre message sortant d'après le nom de destinataire (partie à gauche du "@" de l'adresse) auquel vous l'expédiez, même si cette boîte aux lettres n'existe pas.

Voir également la variable "6.3.181 (`$record`)".

#### 6.3.52 `forward_decode`

Type: booléen

Valeur par défaut: yes

Contrôle le décodage de messages MIME complexes en texte simple (text/plain) lorsque vous faites suivre un courrier. Les en-têtes du message sont également décodés d'après la RFC 2047. Cette variable n'est utilisée que si "6.3.103 (`$mime_forward`)" n'est *pas* positionnée, sinon c'est "6.3.104 (`$mime_forward_decode`)" qui est utilisée à la place.

#### 6.3.53 `forward_format`

Type: chaîne

Valeur par défaut: "[%a: %s]"

Cette variable contrôle l'objet utilisé par défaut lorsque vous faites suivre un message. Elle utilise les mêmes séquences que la variable "6.3.83 (`$index_format`)".

#### 6.3.54 `forward_quote`

Type: booléen

Valeur par défaut: no

Si cette variable *est* positionnée, les messages que vous faites suivre inclus dans le corps du courrier (quand "6.3.103 (`$mime_forward`)" n'est *pas* positionnée) seront cités en utilisant la chaîne "6.3.82 (`$indent_string`)".

#### 6.3.55 `from`

Type: adresse électronique

Valeur par défaut: ""

Quand elle est positionnée, cette variable contient une adresse d'expéditeur ("From:") à utiliser par défaut. Il est possible de passer outre cette variable à l'aide de la commande `my_hdr` (utilisée dans des `send-hooks`) et de la variable "6.3.187 (`$reverse_name`)".

Par défaut, le contenu de la variable d'environnement `EMAIL` est utilisé.

### 6.3.56 gecos\_mask

Type: expression rationnelle

Valeur par défaut: "`^[^,]*`"

Une expression rationnelle utilisée par Mutt pour analyser le champ GECOS d'une entrée de passwd lors de l'expansion d'un alias. Par défaut, elle est positionnée à "`^[^,]*`" ce qui renvoie la chaîne jusqu'à la première virgule rencontrée. Si le champ GECOS contient une chaîne du type "nom, prénom", vous devriez positionner "[6.3.56](#) (`$gecos_mask`)" à "`.*`".

Cela peut se révéler utile si vous rencontrez la situation suivante : vous envoyez un courriel à l'utilisateur d'identifiant (ID) pauld dont le nom est Paul Dupont. Si Mutt remplace pauld par "Dupont", alors vous devriez positionner "[6.3.56](#) (`$gecos_mask`)" à une expression rationnelle qui fera la correspondance avec le nom entier, de telle sorte que "Dupont" devienne "Dupont, Paul".

### 6.3.57 hdrs

Type: booléen

Valeur par défaut: yes Quand cette variable n'est pas positionnée, les champs d'en-têtes normalement ajoutés par la commande "[3.12](#) (`my_hdr`)" ne sont pas créés. Cette variable ne doit *pas* être positionnée *avant* la composition d'un nouveau courriel ou d'une réponse pour avoir un effet. Sinon, les champs d'en-têtes définis par l'utilisateur seront ajoutés à chaque nouveau message.

### 6.3.58 header

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, Mutt inclus les en-têtes du message auquel vous répondez dans le tampon d'édition. La configuration "[6.3.234](#) (`$weed`)" est appliquées.

### 6.3.59 help

Type: booléen

Valeur par défaut: yes

Quand cette variable est positionnée, un rappel des raccourcis claviers pour les principales fonctions est affiché sur la première ligne de l'écran.

**Remarque :** le raccourci sera affiché de manière incorrecte s'il consiste en une séquence de touches plutôt qu'une seule. De même, la ligne d'aide peut ne pas être mise à jour si un raccourci est modifié en cours d'exécution. Comme cette variable est principalement destinée aux nouveaux utilisateurs, rien de ceci ne devrait représenter un problème crucial.

### 6.3.60 hidden\_host

Type: booléen

Valeur par défaut: no

Quand cette variable est positionnée, Mutt omettra la partie nom d'hôte de la variable "[6.3.67](#) (`$hostname`)". Cette variable n'a pas d'influence sur la génération des identifiants de courriel (Message-ID), et ne conduit pas à la suppression des domaines de premier niveau.

### 6.3.61 `hide_limited`

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, Mutt ne montrera pas dans l'arborescence des discussions si des messages sont manquants.

### 6.3.62 `hide_missing`

Type: booléen

Valeur par défaut: yes

Si cette variable est positionnée, Mutt ne montrera pas dans l'arborescence des discussions si des messages sont cachés par l'usage d'un affichage sélectif par motif.

### 6.3.63 `hide_top_limited`

Type: booléen

Valeur par défaut: no

Quand cette variable est positionnée, Mutt ne montrera pas dans l'arborescence si les premiers messages de discussions sont absent. Notez que cette option est sans effet si [6.3.61](#) (`$hide_limited`) est positionnée.

### 6.3.64 `hide_top_missing`

Type: booléen

Valeur par défaut: yes

Si cette variable est positionnée, Mutt ne montrera pas dans l'arborescence des discussions si les premiers messages de discussions sont cachés par l'usage d'un affichage sélectif par motif.

### 6.3.65 `history`

Type: nombre

Valeur par défaut: 10

Cette variable contrôle la taille (en nombre de chaînes mémorisées) de l'historique des chaînes entrées. L'historique est vidé quand cette variable est modifiée.

### 6.3.66 `honor_followup_to`

Type: quadri-option

Valeur par défaut: yes

Cette variable contrôle si l'en-tête Mail-Followup-To est respecté lors d'une réponse à tous les destinataires.

### 6.3.67 `hostname`

Type: chaîne

Valeur par défaut: ""

Spécifie le nom d'hôte à utiliser après le "@" dans les adresses électroniques locales. Cette variable est prioritaire par rapport à la définition obtenue depuis le fichier `/etc/rseolv.conf` au moment de la compilation.

### 6.3.68 ignore\_list\_reply\_to

Type: booléen

Valeur par défaut: no

Affecte le comportement de la fonction *reply* (répondre) lors de réponses à des courriers provenant de listes de diffusion. Quand cette variable est positionnée, si le champ d'en-têtes "Reply-To:" possède la même valeur que le champ "To:", Mutt considérera que le champ "Reply-To:" a été positionné par le gestionnaire de liste afin d'automatiser les réponses vers la liste, et ignorera ce champ. Pour expédier une réponse vers la liste alors que cette option est positionnée, utilisez la fonction *list-reply* (répondre à la liste); la fonction *group-reply* (répondre à tous les destinataires) répondra à la fois à l'expéditeur et à la liste.

### 6.3.69 imap\_authenticators

Type: chaîne

Valeur par défaut: ""

Cela consiste en une liste de méthodes d'authentification séparées par deux-points (":") que Mutt est susceptible d'utiliser pour se connecter à un serveur IMAP, classées par ordre à essayer. Les méthodes d'authentification sont soit 'login' ou la partie droite d'une chaîne de possibilités IMAP 'AUTH=xxx', par exemple 'digest-md5' 'gssapi' ou 'cram-md5'. Ce paramètre n'est pas sensible à la casse. Si ce paramètre n'est pas positionné (cas par défaut) Mutt tentera toutes les méthodes disponibles, de la plus sécurisée à la moins sécurisée.

Exemple : set imap\_authenticators="gssapi:cram-md5:login"

**Remarque :** Mutt n'aura recours à des méthodes alternatives que si les précédentes sont indisponibles. Si une méthode est disponible mais que l'authentification échoue, Mutt ne se connectera pas au serveur IMAP.

### 6.3.70 imap\_delim\_chars

Type: chaîne

Valeur par défaut: "/." Cette variable contient la liste de caractères que vous souhaitez utiliser comme séparateurs pour afficher les chemins IMAP. Cela sert en particulier à l'usage du raccourci '=' pour votre variable *folder*.

### 6.3.71 imap\_force\_ssl

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, Mutt utilisera toujours SSL pour se connecter à un serveur IMAP.

### 6.3.72 imap\_home\_namespace

Type: chaîne

Valeur par défaut: ""

Généralement, vous souhaitez voir vos dossiers personnels en même temps que votre INBOX (boîte aux lettres entrante) dans la sélecteur de boîtes IMAP. Si vous voyez autre chose, vous pouvez positionner cette variable sur le chemin IMAP vers vos dossiers.

### 6.3.73 `imap_keepalive`

Type: nombre

Valeur par défaut: 900

Cette variable spécifie le temps d'attente maximum avant que Mutt n'interroge les connexions IMAP ouvertes, afin d'éviter que le serveur ne les ferme alors que Mutt n'en a pas fini avec elles. La valeur par défaut bien inférieure au temps spécifié par la RFC (30 minutes) pour qu'un serveur agisse ainsi, mais en pratique la RFC est très souvent enfreinte. Réduisez cette valeur si votre connexion au serveur IMAP est coupée pour cause d'inactivité.

### 6.3.74 `imap_list_subscribed`

Type: booléen

Valeur par défaut: no

Cette variable spécifie si la navigation entre les dossiers IMAP doit se préoccuper de tous les dossiers ou seulement ceux auxquels vous êtes abonnés. Ce paramètre peut être inversé dans le sélecteur IMAP à l'aide de la commande *toggle-subscribed*.

### 6.3.75 `imap_pass`

Type: chaîne

Valeur par défaut: ""

Spécifie le mot de passe pour votre compte IMAP. Si cette variable n'est pas positionnée, Mutt vous demandera votre mot de passe à l'invocation de la fonction de récupération de courrier. **Attention** : vous devriez n'utiliser cette option que sur une machine relativement bien sécurisée, car le super-utilisateur (root) peut lire votre fichier de configuration muttrc même si vous êtes le seul à en avoir le droit.

### 6.3.76 `imap_passive`

Type: booléen

Valeur par défaut: yes

Quand cette variable est positionnée, Mutt n'ouvrira pas de nouvelles connexions IMAP pour vérifier la présence de nouveaux courriers. Mutt ne fera cette interrogation que par l'intermédiaire de connexions IMAP déjà ouvertes. Ceci est utile si vous ne souhaitez pas que Mutt vous demande un nom d'utilisateur et un mot de passe au démarrage, ou si l'ouverture d'une connexion est lente.

### 6.3.77 `imap_peek`

Type: booléen

Valeur par défaut: yes

Si cette variable est positionnée, Mutt évitera de marquer implicitement votre courrier comme ayant été lu dès que vous récupérez un message sur le serveur. C'est généralement une bonne chose, mais cela peut ralentir sensiblement la fermeture d'un dossier IMAP. Cette option est là pour satisfaire les fans de la rapidité.

### 6.3.78 `imap_servernoise`

Type: booléen

Valeur par défaut: yes

Si cette option est positionnée, Mutt affichera les avertissements des serveurs IMAP sous la forme de messages d'erreur. Comme ces messages sont la plupart du temps inoffensifs, ou dûs à des problèmes de configuration sur le serveur dont la résolution est hors de portée de l'utilisateur, vous pouvez souhaiter les supprimer.

### 6.3.79 `imap_user`

Type: chaîne

Valeur par défaut: ""

Votre "login" (nom d'utilisateur) sur le serveur IMAP.

Par défaut, cette variable est égale à votre nom d'utilisateur sur la machine locale.

### 6.3.80 `implicit_autoview`

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, Mutt recherchera une entrée mailcap comportant l'indicateur `copiousoutput` pour *chaque* attachement MIME pour lequel il n'a aucun afficheur interne défini. Si une telle entrée est trouvée, Mutt utilisera l'afficheur qu'elle définit pour convertir le corps de l'attachement sous forme de texte.

### 6.3.81 `include`

Type: quadri-option

Valeur par défaut: ask-yes

Contrôle si une copie du ou des messages auxquels vous répondez doit être incluse dans votre réponse.

### 6.3.82 `indent_string`

Type: chaîne

Valeur par défaut: "> "

Spécifie la chaîne de caractères à ajouter en tête de chaque ligne de texte cité auquel vous répondez. Il est très fortement conseillé de ne pas modifier cette variable, qui a tendance à agiter les plus fanatiques gardiens de la netiquette.

### 6.3.83 `index_format`

Type: chaîne

Valeur par défaut: "%4C %Z %{%b %d} %-15.15L (%4l) %s"

Cette variable vous permet de personnaliser à votre convenance le format d'une ligne de l'index des messages.

Les "chaînes de format" sont similaires aux chaînes utilisées dans la fonction "C" printf pour formater les sorties (voir la page de manuel pour plus de détails). Les séquences de format suivantes sont reconnues par Mutt :

**%a**

adresse de l'expéditeur

**%b**

nom du dossier originel (penser à `_b` oïte aux lettres)

**%B**

la liste à laquelle le courrier a été envoyé, sinon le nom du dossier (%b).

**%c**

nombre de caractères (en octets) dans le message

**%C**

numéro du message courant

**%d**

date et heure du message dans le format spécifié par “date\_format” et converties dans la zone horaire de l’expéditeur

**%D**

date et heure du message dans le format spécifié par “date\_format” et converties dans la zone horaire locale

**%e**

numéro du message courant dans la discussion

**%E**

nombre de messages dans la discussion courante

**%f**

champ d’en-têtes From: complet (adresse + nom réel)

**%F**

nom de l’expéditeur, ou du destinataire si vous êtes l’auteur du message

**%i**

identifiant du message courant (“message-id”)

**%l**

nombre de lignes du message

**%L**

Si une adresse dans les champs d’en-tête To ou CC correspond à une adresse définie par la commande “lists”, cela affiche "To <nom-de-la-liste>" ; autrement, identique à %F.

**%m**

nombre total de courriers dans la boîte aux lettres

**%M**

nombre de messages cachés si la discussion est “repliée”

**%N**

score du message

**%n**

nom réel de l’auteur (ou son adresse si le nom est absent)

**%O**

(dossier de sauvegarde `_O_riginel`) L'endroit où Mutt aurait originellement sauvé le message : nom de liste, ou à défaut de liste, nom de destinataire

**%s**

objet du message

**%S**

statut du message (N/D/d/!/r/\*)

**%t**

champ d'en-têtes 'To:' (destinataires)

**%T**

Le caractère approprié, selon la variable 6.3.225 (`$to_chars`)

**%u**

nom d'utilisateur (login) de l'auteur

**%v**

prénom de l'auteur, ou destinataire si vous êtes l'auteur du message

**%y**

champ d'en-têtes 'x-label:', s'il est présent

**%Y**

champ d'en-têtes 'x-label:', s'il est présent, et (1) le message ne fait pas partie de l'aborescence d'une discussion, (2) est le premier message d'une enfilade, ou (3) le champ 'x-label' est différent du champ 'x-label' du message précédent.

**%Z**

indicateurs d'état du message

**%{fmt}**

la date et l'heure du message sont converties selon la zone horaire de l'expéditeur, et "fmt" est développée par la fonction de bibliothèque "strftime"; si cette séquence est précédée d'un point d'exclamation, l'usage des locales est désactivé

**%[fmt]**

la date et l'heure du message sont converties selon la zone horaire locale, et "fmt" est développée par la fonction de bibliothèque "strftime"; si cette séquence est précédée d'un point d'exclamation, l'usage des locales est désactivé

**%(fmt)**

la date et l'heure à laquelle le message a été reçu. "fmt" est développée par la fonction de bibliothèque "strftime"; si cette séquence est précédée d'un point d'exclamation, l'usage des locales est désactivé

**%<fmt>**

l'heure locale courante. "fmt" est développée par la fonction de bibliothèque "strftime"; si cette séquence est précédée d'un point d'exclamation, l'usage des locales est désactivé

**%>X**

justifier à droite le reste de la chaîne en remplissant l'intervalle avec le caractère "X"

%|X

remplir la fin de la ligne avec le caractère "X"

Voir également : "6.3.225 (\$to\_chars)".

### 6.3.84 ispell

Type: chemin

Valeur par défaut: "ISPELL"

Comment appeler ispell (logiciel de correction orthographique GNU)

### 6.3.85 keep\_flagged

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, les messages marqués comme important (voir la fonction *flag-message*) ne seront pas déplacés vers votre boîte aux lettres "6.3.94 (\$mbox)", ou selon une directive "3.10 (mbox-hook)".

### 6.3.86 locale

Type: chaîne

Valeur par défaut: "C"

La locale (informations de localisation) utilisée par *strftime(3)* pour formater les dates. Les valeurs autorisées sont celles acceptées par votre système pour la variable de locale *LC\_TIME*.

### 6.3.87 mail\_check

Type: nombre

Valeur par défaut: 5

Cette variable configure l'intervalle (en secondes) entre deux vérifications de la présence de nouveaux courriers par Mutt.

### 6.3.88 mailcap\_path

Type: chaîne

Valeur par défaut: ""

Cette variable spécifie quels fichiers consulter lors d'une tentative d'affichage d'un corps de message MIME dont le type n'est pas directement supporté par Mutt.

### 6.3.89 mailcap\_sanitize

Type: booléen

Valeur par défaut: yes

Si cette variable est positionnée, Mutt restreindra les séquences de caractères mailcap développables ("%") à un jeu de caractères bien défini ne comportant pas de risques. Ceci est un réglage sûr, mais nous ne pouvons pas affirmer qu'il ne bloquera pas le fonctionnement de quelques mécanismes MIME sophistiqués.

**NE MODIFIEZ PAS CETTE VARIABLE À MOINS D'ÊTRE VRAIMENT SÛR DE CE QUE VOUS FAITES !**

### 6.3.90 maildir\_trash

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, les messages marqués pour suppression seront gardés avec l'indicateur (T)rashed (supprimé) au lieu d'être effacés. **REMARQUE** : cela ne s'applique qu'aux boîtes aux lettres de format maildir. Positionner cette variable n'a aucun effet sur les autres types de boîte.

### 6.3.91 mark\_old

Type: booléen

Valeur par défaut: yes

Contrôle si Mutt fait la distinction entre les *nouveaux* ("new") messages et les *anciens* messages **non lus** ("old"). Par défaut, Mutt positionnera l'indicateur "old" aux nouveaux messages si vous fermez une boîte aux lettres sans les avoir lus. Au prochain lancement de Mutt, ces messages seront affichés avec l'indicateur "O", montrant qu'ils sont anciens. Pour que Mutt considère tous les messages non lus comme nouveaux, ne positionnez pas cette variable.

### 6.3.92 markers

Type: booléen

Valeur par défaut: yes

Contrôle l'affichage des lignes plus larges que l'afficheur interne. Si cette variable est positionnée, un marqueur "+" est affiché au début des lignes coupées. Voir également la variable "6.3.205 (\$smart\_wrap)".

### 6.3.93 mask

Type: expression rationnelle

Valeur par défaut: "!\.^[^.]"

Une expression rationnelle utilisée dans le sélecteur de fichiers, pouvant être précédée de l'opérateur de *négation* "!". Seuls les fichiers dont le nom correspond à ce masque seront affichés. La correspondance est toujours sensible à la casse.

### 6.3.94 mbox

Type: chemin

Valeur par défaut: "~/mbox"

Cette variable spécifie le dossier dans lequel les messages lus de votre spoule ("6.3.213 (\$spoolfile)") seront déplacés.

### 6.3.95 mbox\_type

Type: type de boîte

Valeur par défaut: mbox

Le format de boîte aux lettres utilisés par défaut lors de la création de nouveaux dossiers. Il peut être d'un des types suivants : mbox, MMDF, MH et Maildir.

### 6.3.96 metoo

Type: booléen

Valeur par défaut: no

Si cette variable n'est pas positionnée, Mutt ôtera votre adresse de la liste des destinataires lors d'une réponse à un message.

### 6.3.97 menu\_scroll

Type: booléen

Valeur par défaut: no

Quand cette variable *est* positionnée, les menus défileront d'une ligne vers le haut ou le bas si vous essayez d'accéder à une entrée en dehors des limites de l'écran. Si elle n'est *pas* positionnée, l'écran est effacé et la page de menu suivante ou précédente est affichée (utile pour les connections lentes, car cela évite de nombreux réaffichages)

### 6.3.98 meta\_key

Type: booléen

Valeur par défaut: no

Cette variable, si elle est positionnée, force Mutt à interpréter les frappes au clavier comportant le bit de poids fort (bit 8) comme si l'utilisateur avait pressé la touche ESC et la touche correspondante après suppression du bit de poids fort. Par exemple, si la touche pressée a une valeur ASCII de 0xf4, alors elle est traitée comme si l'utilisateur avait pressé la touche "x". Cela résulte du fait qu'ôter le poids fort de "0xf4" donne "0x74", ce qui correspond au caractère ASCII "x".

### 6.3.99 mh\_purge

Type: booléen

Valeur par défaut: no

Quand cette variable n'est pas positionnée, Mutt adoptera un comportement similaire à mh en renommant les messages marqués pour suppression en *,<ancien nom de fichier>* au lieu de les effacer réellement. Si elle est positionnée, les fichiers seront simplement effacés.

### 6.3.100 mh\_seq\_flagged

Type: chaîne

Valeur par défaut: "flagged"

Le nom de la séquence MH utilisée pour marquer les messages avec l'indicateur "important".

### 6.3.101 mh\_seq\_replied

Type: chaîne

Valeur par défaut: "replied"

Le nom de la séquence MH utilisée pour marquer les messages auxquels vous avez répondu.

### 6.3.102 `mh_seq_unseen`

Type: chaîne

Valeur par défaut: "unseen"

Le nom de la séquence MH utilisée pour les messages non vus.

### 6.3.103 `mime_forward`

Type: quadri-option

Valeur par défaut: no

Quand cette variable est positionnée, le message que vous faites suivre sera attaché en tant que pièce jointe MIME au lieu d'être incluse dans le corps du courrier. Cela est utile pour que le destinataire puisse correctement voir le message que vous lui faites suivre tel que vous l'avez reçu. Si vous souhaitez pouvoir changer votre manière de faire suivre des messages d'un courrier à l'autre, positionnez cette variable à ask-no ou ask-yes.

Voir également "[6.3.52](#) (`$forward_decode`)" et "[6.3.104](#) (`$mime_forward_decode`)".

### 6.3.104 `mime_forward_decode`

Type: booléen

Valeur par défaut: no

Contrôle le décodage de messages MIME complexes vers du texte simple (text/plain) lorsque vous faites suivre un message alors que "[6.3.103](#) (`$mime_forward`)" est positionnée. Autrement, c'est "[6.3.52](#) (`$forward_decode`)" qui est utilisée.

### 6.3.105 `mime_forward_rest`

Type: quadri-option

Valeur par défaut: yes

Lorsque vous faites suivre plusieurs attachements d'un message MIME à partir du menu attachements, les attachements qui ne peuvent être décodés de manière raisonnable seront attachés au message nouvellement composé si cette option est positionnée.

### 6.3.106 `mix_entry_format`

Type: chaîne

Valeur par défaut: "%4n %c %-16s %a"

Cette variable décrit le format de la ligne de renvoi (remailer line) sur l'écran de sélection de chaîne Mix-master. Les séquences de format de type printf suivante sont possibles :

**%n**

Le numéro en cours sur le menu.

**%c**

Capacités du renvoyeur (remailer).

**%s**

Nom court du renvoyeur (remailer).

**%a**

Adresse courriel du renvoyeur (remailer).

### 6.3.107 **mixmaster**

Type: chemin

Valeur par défaut: "MIXMASTER"

Cette variable contient le chemin vers l'exécutable Mixmaster de votre système. Il est utilisé avec un ensemble de paramètres pour récupérer la liste des renvoyeurs (remailers) connus, et finalement envoyer un message à travers la chaîne Mixmaster.

### 6.3.108 **move**

Type: quadri-option

Valeur par défaut: ask-no

Détermine s'il vous est demandé confirmation avant de déplacer des messages lus depuis votre spoule vers votre boîte aux lettres "6.3.94 (\$mbox)", ou après une commande "3.10 (mbox-hook)".

### 6.3.109 **message\_format**

Type: chaîne

Valeur par défaut: "%s"

La chaîne de caractères affichée dans le menu "attachements" pour les attachements de type "message/rfc822". Voir "6.3.83 (\$index\_format)" pour la liste complète des séquences de format reconnues.

### 6.3.110 **pager**

Type: chemin

Valeur par défaut: "builtin"

Cette variable détermine quel afficheur utiliser pour voir les messages. "builtin" signifie que vous voulez utiliser l'afficheur interne. Sinon, cette variable doit préciser le chemin d'accès à l'afficheur que vous voulez utiliser.

Utiliser un afficheur externe peut avoir des inconvénients : des frappes de touches supplémentaires sont nécessaires puisque vous ne pouvez appeler directement les fonctions de Mutt depuis l'afficheur, et le redimensionnement de l'écran fera que les lignes plus longues que la largeur de l'écran seront mal formatées dans le menu d'aide.

### 6.3.111 **pager\_context**

Type: nombre

Valeur par défaut: 0

Cette variable détermine le nombre de lignes de contexte qui est donné lors de l'affichage la page suivant ou précédente dans l'afficheur interne. Par défaut, Mutt affiche en tête de la page suivant la dernière affichée sur la page en cours (aucune ligne de contexte).

### 6.3.112 pager\_format

Type: chaîne

Valeur par défaut: "-%Z- %C/%m: %-20.20n %s"

Cette variable détermine le format du message d'état (status) en une ligne, affiché avant chaque message dans l'afficheur interne ou dans un afficheur externe. Les séquences de format valides sont listées dans la section "6.3.83 (\$index\_format)".

### 6.3.113 pager\_index\_lines

Type: nombre

Valeur par défaut: 0

Spécifie le nombre de lignes de mini-index qui sont affichées quand on est dans l'afficheur. Le message courant, sauf s'il est dans les premiers ou les derniers messages du dossier, est affiché aux alentours du premier tiers du mini-index (en partant du haut), ce qui donne au lecteur le contexte de quelques messages précédents et suivants. Cela est par exemple utile pour voir combien de message il reste à lire dans le fil de discussion courant. Une des lignes est réservée pour la barre d'état de l'index, et donc un *pager\_index\_lines* de 6 ne montrer effectivement que 5 lignes d'index. Une valeur de 0 entraîne qu'aucun index ne sera affiché. Si le nombre de messages dans la dossier courant est inférieur à *pager\_index\_lines*, alors l'index n'utilisera que le minimum de lignes nécessaires.

### 6.3.114 pager\_stop

Type: booléen

Valeur par défaut: no

Quand cette variable est positionnée, l'afficheur interne ne se placera **pas** au prochain message lorsque vous êtes à la fin d'un message et que vous demandez la fonction *next-page*.

### 6.3.115 pgp\_autosign

Type: booléen

Valeur par défaut: no

Positionner cette variable impose à Mutt de toujours essayer de signer en PGP/MIME les messages sortants. Cela peut être remplacé par l'utilisation du *pgp-menu*, quand le chiffrement n'est pas requis ou alors que la signature est demandée.

### 6.3.116 pgp\_autoencrypt

Type: booléen

Valeur par défaut: no

Positionner cette variable impose à Mutt de toujours essayer de chiffrer en PGP/MIME les messages sortants. Cela n'a probablement d'intérêt qu'en association avec la commande *send-hook*. Cela peut être remplacé par l'utilisation du *pgp-menu*, quand le chiffrement n'est pas requis ou alors que la signature est demandée.

### 6.3.117 pgp\_ignore\_subkeys

Type: booléen

Valeur par défaut: yes

Positionner cette variable impose à Mutt d'ignorer le sous-clés OpenPGP. Au lieu de quoi, la clé principale hérite des capacités des sous-clés. Enlevez le positionnement de cette variable si vous voulez jouer aux jeu intéressant de sélection de clé.

### 6.3.118 `pgp_entry_format`

Type: chaîne

Valeur par défaut: "%4n %t%f %4l/0x%k %-4a %2c %u"

Cette variable permet de configurer le menu de sélection de clé PGP à votre goût. Cette chaîne de caractère est similaire à celle de "[6.3.83](#) (`$index_format`)", mais elle a son propre ensemble de séquences de format de type printf :

`%n`

nombre

`%k`

id de clé

`%u`

id d'utilisateur

`%a`

algorithme

`%l`

longueur de clé

`%f`

drapeaux

`%c`

capacités

`%t`

confiance/validité de l'association clé-uid

`%[<s>]`

date de la clé, où `<s>` est une expression `strftime(3)`

### 6.3.119 `pgp_good_sign`

Type: expression rationnelle

Valeur par défaut: ""

Si vous assignez un texte à cette variable, une signature PGP n'est considéré comme valide que si la sortie de [6.3.134](#) (`$pgp_verify_command`) contient ce texte. Utilisez cette variable si le code de retour de la commande est 0 même pour les mauvaises signatures.

### 6.3.120 `pgp_long_ids`

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, Mutt utilise les IDs de clé PGP en 64 bits. Sinon, il utilise les IDs de clé classiques en 32 bits.

### 6.3.121 `pgp_replyencrypt`

Type: booléen

Valeur par défaut: yes

Si cette variable est positionnée, les réponses aux messages chiffrés sont automatiquement chiffrées.

### 6.3.122 `pgp_replysign`

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, les réponses aux messages signés sont automatiquement signées en PGP.

**Note:** cela ne marche pas avec les messages chiffrés et signés !

### 6.3.123 `pgp_replysignencrypted`

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, les réponses aux messages chiffrés en PGP sont automatiquement signées. Cela est pertinent en combinaison avec “6.3.121 (`$pgp_replyencrypt`)”, parce que cela permet de signer tous les messages qui sont automatiquement chiffrés. Cette méthode permet de résoudre le problème décrit dans “6.3.122 (`$pgp_replysign`)”, à savoir que Mutt n’est pas capable de distinguer si un message chiffré est aussi signé.

### 6.3.124 `pgp_retainable_sigs`

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, les messages chiffrés et signés sont composés de parties de corps multipart/signed et multipart/encrypted imbriquées.

Cela est utile pour les applications comme les listes de diffusion chiffrées et signées, où la couche externe (multipart/encrypted) peut facilement être retirée, alors que la partie multipart/signed interne est conservée.

### 6.3.125 `pgp_show_unusable`

Type: booléen

Valeur par défaut: yes

Si cette variable est positionnée, Mutt affiche les clés non utilisables dans le menu de sélection des clés PGP. Cela inclut les clés qui ont été révoquées, qui ont expiré ou qui ont été marquées comme désactivées (“disabled”) par l’utilisateur.

**6.3.126** `pgp_sign_as`

Type: chaîne

Valeur par défaut: ""

SSi vous avez plus d'une paire de clés, cette option vous permet de spécifier laquelle de vos clés privées il faut utiliser. Il est recommandé d'utiliser la forme ID de clé (keyid) pour spécifier votre clé (par exemple : "0x00112233").

**6.3.127** `pgp_strict_enc`

Type: booléen

Valeur par défaut: yes

Si cette variable est positionnée, Mutt encode automatiquement les messages signés en PGP/MIME en *quoted-printable*. Veuillez noter que retirer le positionnement de cette variable pourrait entraîner des problèmes avec des signatures PGP non vérifiables, et donc, ne changez cela que si vous savez ce que vous faites.

**6.3.128** `pgp_timeout`

Type: nombre

Valeur par défaut: 300

Cette variable indique la durée en secondes au bout de laquelle la phrase de passe expire si elle n'est pas utilisée.

**6.3.129** `pgp_verify_sig`

Type: quadri-option

Valeur par défaut: yes

Si "yes", Mutt essaye toujours de vérifier les signatures PGP/MIME. Si "ask-yes" ou "ask-no", Mutt demande s'il doit ou non vérifier la signature. Si "no", Mutt n'essaye jamais de vérifier les signatures PGP/MIME.

**6.3.130** `pgp_sort_keys`

Type: ordre de tri

Valeur par défaut: address

Précise comment les entrées du menu des clés PGP ('pgp keys') sont triées. Les valeurs possibles sont :

**address**

tri alphabétique par ID d'utilisateur

**keyid**

tri alphabétique par ID de clé

**date**

tri par date de création des clés

**trust**

tri par confiance en les clés

If you prefer reverse order of the above values, prefix it with 'reverse'.

### 6.3.131 `pgp_create_traditional`

Type: quadri-option

Valeur par défaut: no

Cette option détermine si Mutt génère l'ancien format de chiffrement ou de signature des message sous certaines conditions.

Notez que PGP/MIME est utilisé automatiquement pour les messages qui ont un jeu de caractères différent de us-ascii, ou qui sont composés de plus d'une partie MIME.

Notez également que l'ancien format des messages PGP est **fortement obsolète**.

### 6.3.132 `pgp_decode_command`

Type: chaîne

Valeur par défaut: ""

Cette variable spécifie la commande (et son format) à utiliser pour déchiffrer des attachements application/pgp.

Les formats de commande PGP ont leur propre ensemble de séquences de format de type printf :

**%p**

Est remplacé par PGPPASSFD=0 quand une phrase de passe est nécessaire, ou par une chaîne vide sinon. Note : cela peut être utilisé avec une structure %?.

**%f**

Est remplacé par le nom d'un fichier contenant le message.

**%s**

Est remplacé par le nom d'un fichier contenant la partie signature d'un attachement multipart/signed au moment de sa vérification.

**%a**

Valeur de 6.3.126 (`$pgp_sign_as`).

**%r**

Une ou plusieurs ID de clé.

Pour avoir des exemples sur la manière de configurer ces formats pour les différentes versions de PGP qui existent, voir les fichiers `pgp*.rc` et `gpg.rc` dans le sous-répertoire `samples/` qui a été installé sur votre system en même temps que cette documentation.

### 6.3.133 `pgp_getkeys_command`

Type: chaîne

Valeur par défaut: ""

La commande indiquée par cette variable est appelée à chaque fois que Mutt a besoin d'information sur des clés publiques. `%r` est la seule séquence de format de type printf utilisable.

### 6.3.134 `pgp_verify_command`

Type: chaîne

Valeur par défaut: ""

La commande indiquée par cette variable est utilisée pour vérifier les signatures PGP/MIME.

### 6.3.135 `pgp_decrypt_command`

Type: chaîne

Valeur par défaut: ""

La commande indiquée par cette variable est utilisée pour déchiffrer un message PGP/MIME chiffré.

### 6.3.136 `pgp_clearsign_command`

Type: chaîne

Valeur par défaut: ""

La commande indiquée par cette variable est utilisée pour créer un attachement de signature PGP en clair ancien format ("clearsigned" old-style PGP attachment). Notez que l'utilisation de ce format est **fortement obsolète** (deprecated).

### 6.3.137 `pgp_sign_command`

Type: chaîne

Valeur par défaut: ""

La commande indiquée par cette variable est utilisée pour créer une signature PGP séparée pour un corps de message (body) au format multipart/signed PGP/MIME.

### 6.3.138 `pgp_encrypt_sign_command`

Type: chaîne

Valeur par défaut: ""

La commande indiquée par cette variable est utilisée pour simultanément chiffrer et signer le corps du message (body).

### 6.3.139 `pgp_encrypt_only_command`

Type: chaîne

Valeur par défaut: ""

La commande indiquée par cette variable est utilisée pour chiffrer le corps du message (body) sans le signer.

### 6.3.140 `pgp_import_command`

Type: chaîne

Valeur par défaut: ""

La commande indiquée par cette variable est utilisée pour importer la clé d'un message dans le trousseau de clés de l'utilisateur. This command is used to import a key from a message into the user's public key ring.

**6.3.141** `pgp_export_command`

Type: chaîne

Valeur par défaut: ""

La commande indiquée par cette variable est utilisée pour exporter une clé publique depuis le trousseau de clés de l'utilisateur.

**6.3.142** `pgp_verify_key_command`

Type: chaîne

Valeur par défaut: ""

La commande indiquée par cette variable est utilisée pour vérifier les informations sur une clé depuis le menu de sélection de clé.

**6.3.143** `pgp_list_secring_command`

Type: chaîne

Valeur par défaut: ""

La commande indiquée par cette variable est utilisée pour lister le contenu du trousseau de clés privées. Le format de sortie doit être analogue à celui utilisé par `gpg -list-keys -with-colons`.

Ce format est aussi généré par l'utilitaire `pgpring` qui est distribué avec Mutt.

**6.3.144** `pgp_list_pubring_command`

Type: chaîne

Valeur par défaut: ""

La commande indiquée par cette variable est utilisée pour lister le contenu du trousseau de clés publiques. Le format de sortie doit être analogue à celui utilisé par `gpg -list-keys -with-colons`.

Ce format est aussi généré par l'utilitaire `pgpring` qui est distribué avec Mutt.

**6.3.145** `forward_decrypt`

Type: booléen

Valeur par défaut: yes

Détermine la gestion des messages chiffrés lors d'un renvoi d'un message (forward). Si cette variable est positionnée, l'encapsulation de chiffrement est supprimée. Cette variable n'est utilisée que si "6.3.103 (`$mime_forward`)" est *positionnée* et que "6.3.104 (`$mime_forward_decode`)" ne l'est *pas*.

**6.3.146** `ssl_starttls`

Type: quadri-option

Valeur par défaut: yes

Si cette variable est positionnée, Mutt essaye d'utiliser STARTTLS sur les serveurs qui en annonce la possibilité. Sinon, Mutt n'essaye pas d'utiliser STARTTLS, quelles que soient les possibilités du

**6.3.147 certificate\_file**

Type: chemin

Valeur par défaut: ""

Cette variable désigne le fichier où sont stockés les certificats que vous validez (trust). Quand un certificat inconnu est rencontré, on vous demande si vous l'acceptez ou non. Si vous l'acceptez, le certificat peut aussi être sauvé dans ce fichier, et ainsi les prochaines connexions seront automatiquement acceptées.

Vous pouvez aussi ajouter manuellement des certificats de CA dans ce fichier. Tout certificat de serveur qui est signé par un de ces certificats de CA est alors automatiquement accepté.

Exemple: set certificate\_file=~/.mutt/certificates

**6.3.148 ssl\_usesystemcerts**

Type: booléen

Valeur par défaut: yes

Si cette variable est positionnée, Mutt utilise les certificats de CA fournis par le système lorsqu'il vérifie si le certificat du serveur est signé par un CA valide (trusted).

**6.3.149 entropy\_file**

Type: chemin

Valeur par défaut: ""

Le fichier indiqué par cette variable doit contenir des données aléatoires et est utilisé pour initialiser les fonctions de la bibliothèque SSL.

**6.3.150 ssl\_use\_sslv2**

Type: booléen

Valeur par défaut: yes

Cette variable détermine si Mutt essaye d'utiliser SSLv2 dans la procédure d'authentification SSL.

**6.3.151 ssl\_use\_sslv3**

Type: booléen

Valeur par défaut: yes

**6.3.152 ssl\_use\_tlsv1**

Type: booléen

Valeur par défaut: yes

Cette variable détermine si Mutt essaye d'utiliser TLSv1 dans la procédure d'authentification SSL.

**6.3.153 pipe\_split**

Type: booléen

Valeur par défaut: no

Cette variable est utilisée avec la commande *pipe-message* et l'opérateur "tag-prefix". Si elle n'est pas positionnée, Mutt concatène les messages et les passe comme un répertoire lorsqu'on demande de passer à un tube (pipe) une liste de messages. Si elle est positionnée, Mutt passe les messages un par un. Dans les deux cas, les messages sont passés dans le tube dans l'ordre de tri courant, et le séparateur "[6.3.155](#) (\$pipe\_sep)" est ajouté après chaque message.

#### 6.3.154 pipe\_decode

Type: booléen

Valeur par défaut: no

Cette variable est utilisée avec la commande *pipe-message*. Quand elle n'est pas positionnée, Mutt passe par le tube (pipe) les messages sans aucun pré-traitement (preprocessing). Si cette variable est positionnée, Mutt commence par nettoyer les entêtes et tenter de déchiffrer le codage PGP/MIME avant de passer les messages au tube.

#### 6.3.155 pipe\_sep

Type: chaîne

Valeur par défaut: "\n"

Séparateur à ajouter entre les messages lorsque qu'une liste de messages marqués (tagged) est passée par un tube (pipe) à une commande Unix externe.

#### 6.3.156 pop\_authenticators

Type: chaîne

Valeur par défaut: ""

Cette variable donne la liste ordonnée des méthodes d'authentification que Mutt peut essayer pour se connecter au serveur POP, les éléments étant séparés par des points-virgules. Les méthodes d'authentification peuvent être 'user', 'apop' ou tout mécanisme SASL, comme 'digest-md5', 'gssapi' ou 'cram-md5'. Cette variable n'est pas sensible à la casse. Si cette variable n'est pas positionnée (situation par défaut), Mutt essaye tous les méthodes disponibles, dans l'ordre du plus sécurisé au moins sécurisé.

Exemple : set pop\_authenticators="digest-md5:apop:utilisateur"

#### 6.3.157 pop\_auth\_try\_all

Type: booléen

Valeur par défaut: yes

Si cette variable est positionnée, Mutt tente toutes les méthodes d'authentification de connexion au serveur POP. Sinon, Mutt ne tente une autre méthode d'authentification que si les méthodes précédentes sont indisponibles. Si une méthode est disponible mais que l'authentification échoue, Mutt ne se connectera pas au serveur POP.

#### 6.3.158 pop\_checkinterval

Type: nombre

Valeur par défaut: 60

Cette variable détermine la fréquence (en secondes) d'interrogation du serveur POP pour récupérer les nouveaux messages.

### 6.3.159 pop\_delete

Type: quadri-option

Valeur par défaut: ask-no

Si cette variable est positionnée, Mutt efface les messages téléchargés avec succès depuis le serveur POP lors de la fonction de récupération des courriers (fetch-mail). Sinon, Mutt télécharge les messages, mais ne les supprime pas sur le serveur POP.

### 6.3.160 pop\_host

Type: chaîne

Valeur par défaut: ""

Nom de votre serveur POP pour la fonction de récupération des courriers (fetch-mail). Vous pouvez éventuellement préciser un numéro port, un nom d'utilisateur et un mot de passe, comme suit :

```
[pop[s]://][utilisateur[:motdepasse]@]serveurpop[:port]
```

### 6.3.161 pop\_last

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, Mutt essaye d'utiliser la commande POP "LAST" pour ne récupérer que les messages non lus du serveur POP quand on utilise la fonction de récupération des courriers (fetch-mail).

### 6.3.162 pop\_reconnect

Type: quadri-option

Valeur par défaut: ask-yes

Détermine si Mutt essaye ou non de se reconnecter au serveur POP en cas de perte de connexion.

### 6.3.163 pop\_user

Type: chaîne

Valeur par défaut: ""

Votre nom d'accès (login) sur le serveur POP.

La valeur par défaut de cette variable est votre nom d'utilisateur sur la machine locale.

### 6.3.164 pop\_pass

Type: chaîne

Valeur par défaut: ""

Précise le mot de passe de votre compte POP. Si cette variable n'est pas positionnée, Mutt demande d'entrer le mot de passe quand on ouvre une boîte aux lettres POP. **Attention** : Vous ne devez utiliser cette option

que si vous êtes sur un machine suffisamment sûre, car l'administrateur (root) peut lire votre muttrc même si vous êtes le seul à pouvoir lire le fichier.

### 6.3.165 `post_indent_string`

Type: chaîne

Valeur par défaut: ""

De manière symétrique à la variable “6.3.15 (`$attribution`)”, Mutt ajoute cette chaîne de caractères après l'inclusion du message auquel on est en train de répondre.

### 6.3.166 `postpone`

Type: quadri-option

Valeur par défaut: ask-yes

Détermine si un message est sauvé ou non dans la boîte aux lettres “6.3.167 (`$postponed`)” quand vous décidez de ne pas l'envoyer immédiatement.

### 6.3.167 `postponed`

Type: chemin

Valeur par défaut: "~/postponed"

Mutt permet d'ajourner (6.3.166 (`postpone`)) indéfiniment l'envoi d'un message que vous éditez. Quand vous décidez d'ajourner un message, Mutt le sauve dans la boîte aux lettres que vous spécifiez avec cette variable. Voir aussi la variable “6.3.166 (`$postpone`)”.

### 6.3.168 `preconnect`

Type: chaîne

Valeur par défaut: ""

Cette variable peut être positionnée à la commande shell à exécuter au cas où Mutt n'arrive pas à établir une connexion au serveur. Cela est utile pour mettre en place des connexions sécurisées, comme avec `ssh(1)`. Si la commande retourne un code d'erreur non nul, Mutt abandonne l'ouverture du service. Exemple :

```
preconnect="ssh -f -q -L 1234:serveurmail.net:143 serveurmail.net sleep 20 < /dev/null > /dev/null"
```

La boîte aux lettres 'toto' sur la machine serveurmail.net peut alors être atteinte avec '{localhost:1234}toto'.

NOTE: Pour que cet exemple fonctionne, vous devez pouvoir ouvrir une session (login) sur la machine distante sans avoir à saisir un mot de passe.

### 6.3.169 `print`

Type: quadri-option

Valeur par défaut: ask-no

Détermine si Mutt demande ou non confirmation avant d'imprimer. C'est utile pour les personnes (comme moi) qui ont tendance à appuyer souvent par erreur sur “p”.

### 6.3.170 `print_command`

Type: chemin

Valeur par défaut: "lpr"

Cette variable détermine la commande tube utilisée pour imprimer des messages.

### 6.3.171 `print_decode`

Type: booléen

Valeur par défaut: yes

Utilisé avec la commande print-message. Si cette variable est positionnée, le message est décodé avec d'être transmis à la commande externe précisée par 6.3.170 (`$print_command`). Sinon, aucun traitement n'est effectué au moment de l'impression. Ce dernier réglage est en particulier utile si vous utilisez un filtre d'impression évolué qui est à même de formater correctement les messages pour l'impression.

### 6.3.172 `print_split`

Type: booléen

Valeur par défaut: no

Utilisé avec la commande print-message. Si cette variable est positionnée, la commande indiquée par 6.3.170 (`$print_command`) est exécutée une fois pour chaque message à imprimer. Sinon, cette commande est exécutée une seule fois, avec tous les messages concaténés, les messages étant séparé par un caractère "form feed".

Les utilisateurs du programme d'impression de courriers **enscript**(1) ont sans doute intérêt à positionner cette variable.

### 6.3.173 `prompt_after`

Type: booléen

Valeur par défaut: yes

Si on utilise un afficheur ("[6.3.110](#) (`$pager`)") *externe*, le positionnement de cette variable fait que Mutt présente une invite de commande quand l'afficheur se termine, au lieu de revenir directement sur le menu d'index. Si cette variable n'est pas positionnée, Mutt revient au menu d'index quand l'afficheur externe se termine.

### 6.3.174 `query_command`

Type: chemin

Valeur par défaut: ""

Cette variable détermine la commande que Mutt utilise pour faire des requêtes externes d'adresse. La chaîne de caractères devrait contenir un %s, qui est remplacé par la chaîne de la requête entrée par l'utilisateur. Voir aussi "[4.5](#) (query)" pour davantage de renseignements.

### 6.3.175 `quit`

Type: quadri-option

Valeur par défaut: yes

Cette variable détermine si “quit” et “exit” font réellement quitter Mutt. Si elle est positionnée à yes, ces deux fonctions font quitter Mutt ; si elle est positionnée à no, elles n’ont aucun effet, et si elle est positionnée à ask-yes ou ask-no, il est demandé confirmation au moment de la demande de sortie.

### 6.3.176 quote\_regexp

Type: expression rationnelle

Valeur par défaut: `^( [ \t]*[>:}#] )+`

Une expression rationnelle utilisée dans l’afficheur interne pour distinguer les parties de citation au sein du corps d’un message.

**Note:** Pour pouvoir utiliser les motifs *quotedx* dans l’afficheur interne, cette variable doit être positionnée à une expression rationnelle qui corresponde *exactement* aux caractères de citation au début des lignes citées.

### 6.3.177 read\_inc

Type: nombre

Valeur par défaut: 10

Si cette variable a une valeur positive, Mutt affiche à quel message il se trouve quand il lit une boîte aux lettres. Le message est affiché après que *read\_inc* messages aient été lus. (par exemple, si la valeur est de 25, Mutt affiche un message au moment où il lit le message 25, et puis un autre quand il atteint le message 50, etc.). Cette variable est destinée à indiquer la progression de la lecture des grosses boîtes aux lettres, ce qui peut être long. Si la valeur est de 0, un unique message apparaît avant le début de la lecture de la boîte aux lettres.

Voir également la variable “[6.3.237](#) (\$write\_inc)”.

### 6.3.178 read\_only

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, tous les dossiers sont ouverts en lecture seule.

### 6.3.179 realname

Type: chaîne

Valeur par défaut: ""

Cette variable détermine quel nom “réel” ou “personnel” doit être utilisé pour envoyer des messages.

Par défaut, c’est le champ GECOS du fichier `/etc/passwd`. Notez que cette variable n’est *pas* utilisée si l’utilisateur a défini un vrai nom à l’aide de la variable [6.3.55](#) (\$from).

### 6.3.180 recall

Type: quadri-option

Valeur par défaut: ask-yes

Cette variable détermine s’il vous est proposé ou non de récupérer un message ajourné quand vous commencez la rédaction d’un nouveau message. Voir aussi “[6.3.167](#) (\$postponed)”.

Mettre cette variable à “yes” n’est en général pas utile, et n’est donc pas recommandé.

**6.3.181 record**

Type: chemin

Valeur par défaut: ""

Cette variable détermine le fichier où seront ajoutés vos messages envoyés. (C'est la méthode de base pour garder une copie des messages émis, mais il existe une autre méthode qui consiste à utiliser la commande "3.12 (my\_hdr)" pour créer un champ *Bcc*: qui pointe vers votre propre adresse.)

La valeur de 6.3.181 (*\$record*) est écrasée par les variables "6.3.51 (*\$force\_name*)" et "6.3.192 (*\$save\_name*)", ainsi que par la commande "3.15 (fcc-hook)".

**6.3.182 reply\_regexp**

Type: expression rationnelle

Valeur par défaut: "^ (re([\[0-9\]+\])\*|aw):[ \t]\*"

Une expression rationnelle utilisée pour reconnaître les messages de réponse au moment d'ordonner un fil de discussion et au moment de répondre. La valeur par défaut correspond à l'anglais "Re:" et à l'allemand "Aw:".

**6.3.183 reply\_self**

Type: booléen

Valeur par défaut: no

Si cette variable n'est pas positionnée et que vous répondez à un message dont vous êtes l'auteur, Mutt supposera que vous voulez répondre aux destinataires de ce message plutôt qu'à vous-même.

**6.3.184 reply\_to**

Type: quadri-option

Valeur par défaut: ask-yes

Si cette variable est positionnée, Mutt vous demande si vous voulez utiliser l'adresse trouvée dans le champ d'en-tête Reply-To: pour répondre à un message. Si vous répondez négativement, il utilisera l'adresse du champ d'en-tête From: à la place. Ce comportement est utile dans le cas de listes de diffusion où le champ d'en-tête Reply-To: est rempli avec l'adresse de la liste et que vous vouliez répondre en privé à l'auteur d'un message.

**6.3.185 resolve**

Type: booléen

Valeur par défaut: yes

Quand cette variable est positionnée, le curseur est automatiquement avancé au prochain message (non effacé) à la suite de chaque commande qui modifie le message courant.

**6.3.186 reverse\_alias**

Type: booléen

Valeur par défaut: no

Cette variable détermine si Mutt affiche ou non dans le menu d'index le nom "privé" trouvé dans le fichier d'alias, et cela dans le cas où il trouve un alias correspondant à l'émetteur du message. Par exemple, si vous avez défini l'alias suivant :

```
alias paul abd30425@quelquepart.net (Paul)
```

et que vous recevez un courrier qui contient l'en-tête suivante :

```
From: abd30425@quelquepart.net
```

Dans le menu d'index, Mutt affichera "Paul" au lieu de "abd30425@quelquepart.net". Cela est utile quand les adresses électroniques ne sont pas très conviviales (adresses CompuServe par exemple).

### 6.3.187 reverse\_name

Type: booléen

Valeur par défaut: no

Il arrive parfois que l'on reçoive un courrier sur une machine, qu'on le déplace sur une autre machine, et que l'on veuille répondre à partir de là. Si cette variable est positionnée, la valeur par défaut de la ligne *From:* du message de réponse est construite à partir de l'adresse de réception du message auquel on répond. Si cette variable n'est pas positionnée, la ligne *From:* utilise l'adresse définie pour la machine en cours d'utilisation.

### 6.3.188 reverse\_realname

Type: booléen

Valeur par défaut: yes

Cette variable permet de régler finement le comportement de la fonctionnalité 6.3.187 (`$reverse_name`). Quand elle est positionnée, Mutt utilise l'adresse des messages entrants telle quelle, en incluant éventuellement les noms réels. Sinon, Mutt écrase ces noms réels avec la valeur de la variable 6.3.179 (`$realname`).

### 6.3.189 rfc2047\_parameters

Type: booléen

Valeur par défaut: no

Quand cette variable est positionnée, Mutt décode les paramètres MIME RFC-2047. Vous pouvez vouloir positionner cette variable si Mutt vous propose d'enregistrer vos attachements sous des noms de fichiers tels que `=?iso-8859-1?Q?fichier=5F=E4=5F991116=2Ezip?=`

Quand cette variable est positionnée de manière interactive, les changements ne sont visibles qu'après avoir changé de dossier.

Notez que cette utilisation de l'encodage RFC-2047 est explicitement interdite par le standard, mais néanmoins rencontrée dans la nature. Notez également que quelque soit le positionnement de ce paramètre, Mutt ne *génèrera jamais* ce type d'encodage, mais utilisera exclusivement l'encodage spécifié dans la RFC-2231.

### 6.3.190 save\_address

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, Mutt utilisera l'adresse complète du destinataire pour choisir le dossier par défaut dans lequel sauver le courrier. Si "6.3.192 (\$save\_name)" ou "6.3.51 (\$force\_name)" est aussi positionné, le choix du dossier fcc sera modifié respectivement.

### 6.3.191 save\_empty

Type: booléen

Valeur par défaut: yes

Si cette variable n'est pas définie, les boîtes aux lettres ne contenant aucun message seront supprimées au moment de leur fermeture (Sauf la boîte "6.3.213 (\$spoolfile)" qui n'est jamais supprimée). Si cette variable est positionnée, aucune boîte aux lettres n'est supprimée.

Notez que cela ne s'applique qu'aux dossiers aux formats mbox et MMDF ; Mutt ne supprime pas les répertoires MH et Maildir.

### 6.3.192 save\_name

Type: booléen

Valeur par défaut: no

Cette variable détermine comment les messages sortants sont sauvés. Quand elle est positionnée, une vérification est faite pour voir s'il existe une boîte aux lettres du même nom que le destinataire (cela est fait en recherchant une boîte aux lettres dans le répertoire "6.3.48 (\$folder)", au nom de l'utilisateur (username) de l'adresse e-mail). Si une telle boîte aux lettres existe, le message partant est sauvé dans cette boîte ; sinon le message est sauvé dans la boîte aux lettres définie par "6.3.181 (\$record)".

Voir aussi la variable "6.3.51 (\$force\_name)".

### 6.3.193 score

Type: booléen

Valeur par défaut: yes

Quand cette variable n'est *pas* positionnée, l'attribution de score n'est pas faite. Cela peut être utile pour désactiver l'utilisation des scores de manière sélective pour certains dossiers où l'on ne veut pas de marquage automatique avec

### 6.3.194 score\_threshold\_delete

Type: nombre

Valeur par défaut: -1

Les messages auxquels a été attribué un score inférieur ou égal à la valeur de cette variable sont automatiquement marqués à supprimer par Mutt. Puisque les scores attribués par Mutt sont toujours positifs ou nuls, la valeur par défaut (-1) ne fera jamais marquer aucun message comme étant à supprimer.

### 6.3.195 score\_threshold\_flag

Type: nombre

Valeur par défaut: 9999

Les messages auxquels a été attribué un score supérieur ou égal à la valeur de cette variable sont automatiquement marqués comme importants (flag).

**6.3.196 score\_threshold\_read**

Type: nombre

Valeur par défaut: -1

Les messages auxquels a été attribué un score inférieur ou égal à la valeur de cette variable sont automatiquement marqués comme lus par Mutt. Puisque les scores attribués par Mutt sont toujours positifs ou nuls, la valeur par défaut (-1) ne fera jamais marquer aucun message comme étant lu.

**6.3.197 send\_charset**

Type: chaîne

Valeur par défaut: "us-ascii:iso-8859-1:utf-8"

Une liste de jeux de caractères (character sets) pour les courriers sortants. Mutt utilise le premier jeu de caractères dans lequel le texte peut être correctement encodé. Si votre "6.3.21 (\$charset)" n'est pas iso-8859-1 et que les destinataires ne peuvent interpréter correctement l'UTF-8, il est souhaitable d'inclure dans cette liste un jeu de caractères très commun choisi judicieusement (comme iso-8859-2, koi8-r ou iso-2022-jp), à la place ou à la suite de "iso-8859-1".

**6.3.198 sendmail**

Type: chemin

Valeur par défaut: "SENDMAIL -oem -oi"

Précise le programme et les arguments utilisés pour envoyer un courrier depuis Mutt. Mutt suppose que ce programme interprète les arguments supplémentaires comme des adresses destinataires.

**6.3.199 sendmail\_wait**

Type: nombre

Valeur par défaut: 0

Détermine la durée à attendre (en secondes) pour que le processus "6.3.198 (\$sendmail)" se termine, avant d'abandonner et de passer l'envoi en tâche de fond.

Mutt interprète la valeur de cette variable comme suit :

>0

nombre de secondes à attendre que sendmail se termine avant de c

0

attendre indéfiniment la fin de sendmail

<0

toujours mettre sendmail en tâche de fond sans attendre

Notez que si vous mettez une valeur non nulle, le flot de sortie du processus fils sera mis dans un fichier temporaire. S'il arrive une erreur, vous serez averti de l'endroit où vous pourrez trouver cette sortie.

### 6.3.200 shell

Type: chemin

Valeur par défaut: ""

La commande à utiliser pour lancer un sous-shell. Par défaut on utilise le shell de login de l'utilisateur, trouvé dans le fichier `/etc/passwd`.

### 6.3.201 sig\_dashes

Type: booléen

Valeur par défaut: yes

Si cette variable est positionnée, un ligne contenant “- ” est insérée avant votre “6.3.203 (\$signature)”. Il est **fortement** recommandé de ne pas enlever le positionnement de cette variable, sauf si votre signature ne contient que votre nom. La raison de cela est que beaucoup de logiciels utilise la chaîne “- \n” pour repérer votre signature. Par exemple, Mutt sait mettre en valeur les signatures en les mettant d’une autre couleur dans son afficheur.

### 6.3.202 sig\_on\_top

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, la signature sera incorporée avant tout texte cité ou renvoyé. Il est **fortement** recommandé de ne pas positionner cette variable, sauf si vous connaissez réellement ses implications, et que vous êtes prêt à recevoir des remarques de la part des gardiens de la netiquette.

### 6.3.203 signature

Type: chemin

Valeur par défaut: “~/signature” Détermine le fichier pour votre signature, laquelle est ajoutée à tous vos messages émis. Si le nom du fichier se termine par un tube (“|”), ce fichier est supposé être un shell-script, et la valeur sera lue depuis sa sortie standard (stdout).

### 6.3.204 simple\_search

Type: chaîne

Valeur par défaut: “~f %s | ~s %s”

Détermine comment Mutt doit étendre une simple recherche en un vrai motif de recherche. Une recherche simple est une recherche qui ne contient aucun opérateur ~. Voir aussi “4.2 (patterns)” pour plus de renseignements sur les motifs de recherche.

Par exemple, si vous tapez simplement “jean” sur une invite de recherche ou de limite, Mutt le transformera automatiquement en la valeur déterminée par cette variable, ce qui avec la valeur par défaut donnerait :

```
~f jean | ~s jean
```

### 6.3.205 smart\_wrap

Type: booléen

Valeur par défaut: yes

Détermine l’affichage des lignes plus longues que la largeur de l’écran au sein de l’afficheur interne. Si cette variable est positionnée, ces longues lignes sont coupées entre les mots. Sinon, elles sont simplement coupées au bord de l’écran. Voir aussi la variable “6.3.92 (\$markers)”.

### 6.3.206 smileys

Type: expression rationnelle

Valeur par défaut: "(>From )|(:[:-^]?[[]])(><){|/DP}"

L’*afficheur* (pager) utilise cette variable pour empêcher quelques erreurs fréquente de “6.3.176 (\$quote\_regexp)”, qui sont le plus souvent des émoticons (smileys) en début de ligne.

### 6.3.207 sleep\_time

Type: nombre

Valeur par défaut: 1

Détermine la durée d’attente (en secondes) au moment d’afficher certains messages d’information, lorsque l’on va de dossier en dossier ou que l’on nettoie les message du répertoire courant. La valeur par défaut est une attente d’une seconde, et une valeur nulle supprime toute attente.

### 6.3.208 sort

Type: ordre de tri

Valeur par défaut: date

Détermine comment les messages sont triés dans le menu d’*index*. Les valeurs possibles sont :

```

date ou date-sent (date d’envoi)
date-received (date de réception)
from (émetteur)
mailbox-order (non trié)
score
size (taille)
subject (sujet)
threads (fils de discussion)
to (destinataire)

```

Il est possible d’utiliser le préfixe *reverse-* pour imposer un ordre de tri inversé (par exemple : set sort=*reverse-date-sent*).

### 6.3.209 sort\_alias

Type: ordre de tri

Valeur par défaut: alias

Détermine comment les entrées du menu ‘alias’ sont triées. Les valeurs possibles sont les suivantes :

```

address (tri alphabétique des adresses e-mail)
alias (tri alphabétique des noms d’alias)
unsorted (ordre des alias dans le fichier .muttrc)

```

### 6.3.210 `sort_aux`

Type: ordre de tri

Valeur par défaut: date

Quand les courriers sont triés par fils de discussion, cette variable définit comment les fils de discussions sont triés entr'eux, et comment les fils sont eux-mêmes triés. Elle peut avoir toutes les valeurs que “6.3.208 (`$sort`)” peut avoir, à part la valeur “threads” (discussion) elle-même (dans ce cas, Mutt utilise date-send, i.e. la date d’envoi). Il est aussi possible d’utiliser le préfixe last- en plus du préfixe reverse-, mais last- doit être mis après reverse-. Le préfixe last- fait qu’entre des descendants d’un même message, on affichera en dernier celui qui a le descendant le plus récent, le reste du classement étant géré selon le reste de la directive `sort_aux`. Par exemple, set `sort_aux=last-date-received` signifie que si un nouveau message est reçu dans un fil de discussion, ce fil-là sera la dernière enfilade affichée (ou le premier si on a positionné `sort=reverse-threads`). Notez que pour un ordre “6.3.208 (`$sort`)” inversé, 6.3.210 (`$sort_aux`) est inversé à son tour (ce qui n’est pas forcément judicieux, mais ce comportement est conservé pour assurer la compatibilité avec les configurations existantes).

### 6.3.211 `sort_browser`

Type: ordre de tri

Valeur par défaut: subject

Détermine la méthode de tri dans le sélecteur de dossiers. Par défaut, les entrées sont triées alphabétiquement. Les valeurs possibles sont :

```
alpha (alphabétiquement)
date
size (taille)
unsorted (non trié)
```

Vous pouvez éventuellement utiliser le préfixe reverse- pour indiquer un tri inversé (exemple : set `sort_browser=reverse-date`).

### 6.3.212 `sort_re`

Type: booléen

Valeur par défaut: yes

Cette variable ne sert que dans le cas d’un affichage trié par fil de discussion avec “6.3.217 (`$strict_threads`)” non positionné. Dans ce cas, elle modifie l’heuristique que Mutt utilise pour ordonner la discussion par sujet. Si `sort_re` est positionné, Mutt ne considérera un message comme le fils d’un autre que si le sujet du message fils commence avec une sous-chaîne correspondant à la valeur de “6.3.182 (`$reply_regexp`)”. Si `sort_re` n’est pas positionné, Mutt attachera toujours le message, que la condition précédente soit respectée ou non, du moment que les parties des deux message ne correspondant pas à “6.3.182 (`$reply_regexp`)” soient identiques.

### 6.3.213 `spoolfile`

Type: chemin

Valeur par défaut: ""

Si votre boîte aux lettres spoule est à un endroit non standard que Mutt ne peut pas trouver, vous pouvez spécifier cet emplacement avec cette variable. Si cette variable n’est pas explicitement définie, Mutt la positionne automatiquement à la valeur de la variable d’environnement `$MAIL`.

**6.3.214 status\_chars**

Type: chaîne

Valeur par défaut: "-\*%A"

Détermine les caractères utilisés par l'indicateur "%r" dans la variable "6.3.215 (\$status\_format)". Le premier caractère est utilisé lorsque la boîte aux lettres n'a pas été modifiée. Le deuxième sert lorsque la boîte aux lettres a été modifiée et doit être resynchronisée. Le troisième est utilisé si la boîte aux lettres est en lecture-seule, ou si elle ne sera pas écrite quand on en sortira (vous pouvez décider si la boîte aux lettres sera écrite ou non à sa sortie avec la commande "toggle-write", associée par défaut à la touche "%"). Le quatrième caractère est utilisé pour indiquer que le dossier en cours a été ouvert dans le mode d'attachement de message (certaines opérations, comme écrire un nouveau message, répondre ou faire suivre un message ne sont pas autorisées dans ce mode).

**6.3.215 status\_format**

Type: chaîne

Valeur par défaut: "-%r-Mutt: %f [Msgs:%?M?%M/?%m%?n? New:%n?%?o? Old:%o?%?d? Del:%d?%?F? Flag:%F?%?t? Tag:%t?%?p? Post:%p?%?b? Inc:%b?%?l? %l?](%s/%S)-%>-(%P)—"

Détermine le format de la ligne de statut affichée dans l'*index*. Cette chaîne ressemble à "6.3.83 (\$index\_format)", mais a son propre ensemble de séquences de format de type printf() :

**%b**

nombre de boîtes aux lettres avec de nouveaux courriers \*

**%d**

nombre de messages effacés \*

**%h**

nom de la machine locale

**%f**

chemin complet (pathname) de la boîte aux lettres courante

**%F**

nombre de messages marqués "important" \*

**%l**

taille (en octets) de la boîte aux lettres courante \*

**%L**

taille (en octets) des messages affichés (i.e., ceux qui correspondent à la limite actuelle) \*

**%m**

nombre de messages dans la boîte aux lettres \*

**%M**

nombre de messages affichés (i.e., ceux qui correspondent à la limite actuelle) \*

**%n**

nombre de nouveaux messages dans la boîte aux lettres \*

|               |   |
|---------------|---|
| <b>%o</b>     | nombre de d'anciens messages non lus  |
| <b>%p</b>     | nombre de messages ajournés *   |
| <b>%P</b>     | pourcentage de l'index parcouru   |
| <b>%r</b>     | indicateur modifié/lecture-seule/pas d'écriture/ message-attaché, suivant la variable <a href="#">6.3.214</a> ( <code>\$status_chars</code> ) |
| <b>%s</b>     | méthode de tri courante ( <a href="#">6.3.208</a> ( <code>\$sort</code> ))  |
| <b>%S</b>     | méthode de tri auxiliaire courante ( <a href="#">6.3.210</a> ( <code>\$sort_aux</code> ))   |
| <b>%t</b>     | nombre de messages marqués *  |
| <b>%u</b>     | nombre de messages non lus *  |
| <b>%v</b>     | numéro de version de Mutt   |
| <b>%V</b>     | motif de limitation actuellement actif, s'il y en a *   |
| <b>%&gt;X</b> | justification droite pour le reste de la chaîne, et remplissage de l'intervalle avec le caractère "X"   |
| <b>% X</b>    | remplissage de la fin de la ligne avec le caractère "X"   |

\* = peut être testé (nullité) pour un affichage conditionnel.

Certaines des séquences de format précédentes peuvent être utilisées pour afficher une chaîne selon qu'elles sont nuls ou non. Par exemple, on pourrait ne vouloir voir que les messages marqués uniquement s'il y a de tels messages, puisque n'en voir aucun n'est pas très significatif. Pour afficher une chaîne suivant la valeur d'une des séquences précédentes, il faut utiliser la structure suivante :

```

%?<caract_de_ctrl>?<chaîne_à_afficher>?

```

où *caract\_de\_format* est un des caractères de la table ci-dessus, et *chaîne\_à\_afficher* est la chaîne qui sera affichée si le *caract\_statut* est non nul. *chaîne\_à\_afficher* **peut** contenir d'autres séquences tout autant que du texte standard, mais il n'est pas possible d'imbriquer des chaînes conditionnelles.

Exemple illustrant comment afficher le nombre de nouveaux messages dans la boîte aux lettres : `:%?n?%n nouveaux messages.?`

De plus, il est possible d'afficher suivant la nullité de la séquence l'une de deux chaînes (la première si la valeur est nulle, la seconde si la valeur est non-nulle), en utilisant la structure suivante : `:%?<caract_ctrl>?<chaîne_si>&<chaîne_sinon>?`

Il est aussi possible de mettre en minuscules le résultat de n'importe quel résultat d'une séquence de type `printf()`. Cela se fait en préfixant les séquences de format avec un tiret bas (`_`). Par exemple, si on veut afficher le nom de la machine locale en minuscules, on mettra : `%_h`

### 6.3.216 `status_on_top`

Type: booléen

Valeur par défaut: no

Positionner cette variable fait que la barre d'état est affichée sur la première ligne de l'écran, plutôt que proche du bas.

### 6.3.217 `strict_threads`

Type: booléen

Valeur par défaut: no

Si cette variable est positionnée, l'affichage des fils de discussion ne tiendra compte que des champs d'en-têtes "In-Reply-To" et "References" lorsque vous triez ("[6.3.208](#) (`$sort`)") par discussion. Par défaut, les messages avec les mêmes sujets sont regroupés ensembles dans des "pseudo-discussions". Ce n'est pas forcément souhaitable, comme par exemple dans une boîte aux lettres où vous pouvez avoir plusieurs messages sans rapports entr'eux, avec le même sujet "salut", et qui seraient mis ensembles.

### 6.3.218 `suspend`

Type: booléen

Valeur par défaut: yes

Quand cette variable n'est pas positionnée, Mutt ne s'arrêtera pas lorsque l'utilisateur appuie sur la touche *susp* du terminal, qui est en général "control-Z". Ceci est utile si vous lancez Mutt à l'intérieur d'un `xterm` en utilisant une commande comme `xterm -e mutt`.

### 6.3.219 `text_flowed`

Type: booléen

Valeur par défaut: no

Quand cette variable est positionnée, Mutt générera des fichiers joints "text/plain; format=flowed". Ce format est plus facilement géré par certains logiciels de courrier, et apparaît en général comme un texte ordinaire. Pour pouvoir pleinement utiliser les possibilités de ce format, votre éditeur doit être compatible avec celui-ci.

Notez que [6.3.82](#) (`$indent_string`) n'est pas pris en compte lorsque cette variable est positionnée.

### 6.3.220 `thread_received`

Type: booléen

Valeur par défaut: no

Quand cette variable est positionnée, Mutt utilise la date de réception plutôt que la date d'émission des messages pour afficher l'arborescence d'une discussion lorsqu'elle est triée par sujet.

**6.3.221 thorough\_search**

Type: booléen

Valeur par défaut: no

Modifie le comportement des opérations de recherche avec `~b` et `~h`, comme décrit dans la section “4.2 (patterns)” ci-dessus. Si cette variable est positionnée, les en-têtes et les fichiers joints des messages sont décodés avant la recherche. Sinon, les messages sont parcourus tels qu'ils sont dans le dossier.

**6.3.222 tilde**

Type: booléen

Valeur par défaut: no

Quand cette variable est positionnée, l'afficheur remplit les lignes vierges en bas de l'écran avec des tildes (“~”).

**6.3.223 timeout**

Type: nombre

Valeur par défaut: 600

Cette variable détermine le *nombre de secondes* pendant lequel Mutt attendra la pression sur une touche dans le menu principal avant d'aller par lui-même vérifier la présence de nouveaux messages. Une valeur nulle ou négative interdira à Mutt d'aller faire cette vérification sans pression sur une touche.

**6.3.224 tmpdir**

Type: chemin

Valeur par défaut: ""

Cette variable permet de spécifier où Mutt doit placer ses fichiers temporaires, nécessaires à l'affichage et à la composition des messages.

**6.3.225 to\_chars**

Type: chaîne

Valeur par défaut: "+TCFL"

Détermine le caractère utilisé pour signaler les courriers dont vous êtes le destinataire. Le premier caractère est celui utilisé lorsque le courrier N'est PAS libellé à votre adresse (" " par défaut). Le second est utilisé lorsque vous êtes l'unique destinataire du message ("+" par défaut). Le troisième sert lorsque votre adresse apparaît dans le champ d'en-tête TO, mais que vous n'êtes pas le seul destinataire du message ("T" par défaut). Le quatrième caractère est utilisé lorsque votre adresse apparaît dans le champ d'en-tête CC, mais que vous n'êtes pas le seul destinataire du message. Le cinquième caractère est utilisé pour indiquer un courrier que *vous* avez envoyé. Le sixième et dernier caractère est utilisé pour marquer un courrier qui a été envoyé à une liste de diffusion à laquelle vous êtes abonné ("L" par défaut).

**6.3.226 tunnel**

Type: chaîne

Valeur par défaut: ""

Le positionnement de cette variable indique à Mutt d'ouvrir un tube (pipe) vers une commande au lieu d'une "raw socket". Cela doit permettre de configurer une connexion déjà authentifiée vers votre serveur IMAP/POP3. Exemple :

```
tunnel="ssh -q mailhost.net /usr/local/libexec/imapd"
```

NOTE : Pour que cet exemple fonctionne, vous devez pouvoir vous connecter à la machine distante sans avoir à entrer un mot de passe.

### 6.3.227 use\_8bitmime

Type: booléen

Valeur par défaut: no

*Attention* : ne positionnez cette variable que si vous utilisez une version de sendmail qui supporte le drapeau -B8BITMIME (comme Sendmail 8.8.x), ou bien vous pourriez ne pas pouvoir envoyer de courrier.

Quand cette variable est positionnée, Mutt invoque “6.3.198 (\$sendmail)” avec le drapeau -B8BITMIME pour envoyer des messages sur 8 bits, afin d'engager la négociation ESMTP.

### 6.3.228 use\_domain

Type: booléen

Valeur par défaut: yes

Quand cette variable est positionnée, Mutt qualifie toutes les adresses locales (celles sans la partie @machine) avec la valeur de “6.3.67 (\$hostname)”. Sinon, aucune adresse n'est qualifiée.

### 6.3.229 use\_from

Type: booléen

Valeur par défaut: yes

Quand cette variable est positionnée, Mutt génère le champ d'en-tête 'From:' lorsqu'il envoie un message. Sinon, aucun champ d'en-tête 'From:' n'est généré, sauf si l'utilisateur en positionne explicitement un en utilisant la commande “3.12 (my\_hdr)”.

### 6.3.230 use\_ipv6

Type: booléen

Valeur par défaut: yes

Quand cette variable est positionnée, Mutt recherche les adresses IPv6 des machines qu'il tente de contacter. Sinon, Mutt se limitera aux adresses IPv4. En principe, la valeur par défaut convient.

### 6.3.231 user\_agent

Type: booléen

Valeur par défaut: yes

Quand cette variable est positionnée, Mutt ajoute un en-tête "User-Agent" aux messages émis, indiquant la version de Mutt ayant servi à les mettre en forme.

**6.3.232 visual**

Type: chemin

Valeur par défaut: ""

Spécifie l'éditeur à appeler quand on invoque la commande `~v` dans l'éditeur interne.

**6.3.233 wait\_key**

Type: booléen

Valeur par défaut: yes

Contrôle si Mutt demande d'appuyer sur une touche après les commandes *shell-escape*, *pipe-message*, *pipe-entry*, *print-message*, et *print-entry*.

Cette variable est aussi utilisée pour la visualisation avec "5.4 (auto\_view)", à condition que l'entrée correspondante dans le mailcap ait un drapeau *needsterminal*, et que le programme externe soit interactif.

Lorsque cette variable est positionnée, Mutt vous demande toujours d'appuyer sur une touche. Sinon, Mutt n'attend cet appui que si le programme externe retourne une valeur non-nulle.

**6.3.234 weed**

Type: booléen

Valeur par défaut: yes

Lorsque cette variable est positionnée, Mutt nettoie les entêtes avant d'afficher, renvoyer, imprimer ou répondre à un message.

**6.3.235 wrap\_search**

Type: booléen

Valeur par défaut: yes

Quand cette variable est positionnée, les recherches continuent à l'autre bout de la boîte à lettre. Sinon les recherches s'arrêtent lorsqu'elles atteignent le premier (ou le dernier) message.

**6.3.236 wrapmargin**

Type: nombre

Valeur par défaut: 0

Contrôle la marge laissée sur le côté droit du terminal lorsque l'afficheur de Mutt doit gérer les retours à la ligne.

**6.3.237 write\_inc**

Type: nombre

Valeur par défaut: 10

Lors de l'écriture d'une boîte à lettres, un message sera affiché tous les *write\_inc* courriers pour indiquer la progression. Si cette variable est positionnée à 0, un unique message sera affiché avant l'écriture de la boîte à lettres.

Voir également la variable "6.3.177 (\$read\_inc)".

**6.3.238 write\_bcc**

Type: booléen

Valeur par défaut: yes

Contrôle si Mutt écrit l'entête Bcc au moment de la préparation du message à envoyer. Cette variable peut intéresser les utilisateurs d'Exim.

**6.4 Fonctions**

Vous trouverez ci-dessous la liste des fonctions disponibles, classées suivant les endroits où elles sont accessibles. Les touches par défaut sont données, ainsi qu'une explication sur ce que fait la fonction. L'affectation de ces fonctions à d'autres touches peut être réalisée à l'aide de la commande 3.3 (bind).

**6.4.1 generic**

Le menu *generic* n'est pas un vrai menu, mais spécifie les fonctions les plus courantes disponibles dans tous les menus à l'exception de *pager* et *editor*. Changer les réglages pour ce menu affectera les affectation de touche par défaut de tous les menus (à part les exception indiquées plus haut).

|                 |           |  |
|-----------------|-----------|--|
| bottom-page     | L         | se déplacer en bas de la page                                |
| current-bottom  | not bound | déplacer l'entrée au bas de l'écran                          |
| current-middle  | not bound | déplacer l'entrée au milieu de l'écran                       |
| current-top     | not bound | déplacer l'entrée en haut de l'écran                         |
| enter-command   | :         | entrer une commande muttrc                                   |
| exit            | q         | sortir de ce menu  |
| first-entry     | =         | se déplacer sur la première entrée                           |
| half-down       | ]         | descendre d'1/2 page   |
| half-up         | [         | remonter d'1/2 page  |
| help            | ?         | cet écran  |
| jump            | number    | aller à un numéro d'index                                    |
| last-entry      | *         | aller à la dernière entrée                                   |
| middle-page     | M         | aller au milieu de la page                                   |
| next-entry      | j         | aller à l'entrée suivante                                    |
| next-line       | >         | descendre d'une ligne  |
| next-page       | z         | aller à la page suivante                                     |
| previous-entry  | k         | aller à l'entrée précédente                                  |
| previous-line   | <         | remonter d'une ligne   |
| previous-page   | Z         | aller à la page précédente                                   |
| refresh         | ^L        | effacer l'écran et réafficher                                |
| search          | /         | rechercher une expression rationnelle                        |
| search-next     | n         | rechercher la prochaine occurrence                           |
| search-opposite | not bound | rechercher la prochaine occurrence dans la direction opposée |
| search-reverse  | ESC /     | rechercher en arrière une expression rationnelle             |
| select-entry    | RET       | sélectionner l'entrée courant                                |
| shell-escape    | !         | exécuter une commande dans un sous-shell                     |
| tag-entry       | t         | marquer l'entrée courante                                    |
| tag-prefix      | ;         | appliquer la prochaine fonction aux messages marqués         |
| top-page        | H         | aller en haut de la page                                     |

## 6.4.2 index

|                        |           |   |
|------------------------|-----------|---|
| bounce-message         | b         | renvoyer un message à un autre utilisateur                  |
| change-folder          | c         | ouvrir un dossier différent                                 |
| change-folder-readonly | ESC c     | ouvrir un dossier différent en lecture seule                |
| clear-flag             | W         | effacer un indicateur de statut d'un message                |
| copy-message           | C         | copier un message dans un fichier ou une boîte aux lettres  |
| create-alias           | a         | créer un alias à partir de l'expéditeur d'un message        |
| decode-copy            | ESC C     | faire une copie décodée (text/plain)                        |
| decode-save            | ESC s     | faire une copie décodée (text/plain) et effacer             |
| delete-message         | d         | effacer l'entrée courante                                   |
| delete-pattern         | D         | effacer les messages correspondant à un motif               |
| delete-subthread       | ESC d     | effacer tous les messages dans la sous-discussion           |
| delete-thread          | ^D        | effacer tous les messages dans la discussion                |
| display-address        | @         | afficher l'adresse complète de l'expéditeur                 |
| display-toggle-weed    | h         | afficher le message et inverser la restriction des en-têtes |
| display-message        | RET       | afficher un message   |
| edit                   | e         | éditer le message brut                                      |
| edit-type              | ^E        | éditer le content-type de l'attachement                     |
| exit                   | x         | sortir de ce menu   |
| extract-keys           | ^K        | extraire les clés publiques PGP                             |
| fetch-mail             | G         | récupérer le courrier depuis un serveur POP                 |
| flag-message           | F         | modifier l'indicateur 'important' d'un message              |
| forget-passphrase      | ^F        | effacer la phrase de passe PGP de la mémoire                |
| forward-message        | f         | faire suivre un message avec des commentaires               |
| group-reply            | g         | répondre à tous les destinataires                           |
| limit                  | l         | afficher seulement les messages correspondant à un motif    |
| list-reply             | L         | répondre à la liste spécifiée                               |
| mail                   | m         | composer un nouveau message                                 |
| mail-key               | ESC k     | envoyer une clé publique PGP                                |
| next-new               | TAB       | aller au nouveau message suivant                            |
| next-subthread         | ESC n     | aller à la sous-discussion suivante                         |
| next-thread            | ^N        | aller à la discussion suivante                              |
| next-undeleted         | j         | aller au message non effacé suivant                         |
| next-unread            | not bound | aller au message non lu suivant                             |
| parent-message         | P         | aller au message père dans la discussion                    |
| pipe-message           |           | passer le message/l'attachement à une commande shell        |
| previous-new           | ESC TAB   | aller au nouveau message précédent                          |
| previous-page          | Z         | aller à la page précédente                                  |
| previous-subthread     | ESC p     | aller à la sous-discussion précédente                       |
| previous-thread        | ^P        | aller à la discussion précédente                            |
| previous-undeleted     | k         | aller au message non effacé précédent                       |
| previous-unread        | not bound | aller au message non lu précédent                           |
| print-message          | p         | imprimer l'entrée courante                                  |
| query                  | Q         | demander des adresses à un programme externe                |
| quit                   | q         | sauver les modifications de la boîte aux lettres et quitter |
| read-subthread         | ESC r     | marquer la sous-discussion courante comme lue               |
| read-thread            | ^R        | marquer la discussion courante comme lue                    |
| recall-message         | R         | rappeler un message ajourné                                 |
| reply                  | r         | répondre à un message                                       |

|                    |       |  |
|--------------------|-------|--|
| resend-message     | ESC e | utiliser le message courant comme modèle pour un nouveau message |
| save-message       | s     | sauver le message/l'attachement dans un fichier                  |
| set-flag           | w     | mettre un indicateur d'état sur un message                       |
| show-version       | V     | afficher la version de Mutt (numéro et date)                     |
| show-limit         | ESC l | afficher le motif de limitation actuel                           |
| sort-mailbox       | o     | trier les messages   |
| sort-reverse       | O     | trier les messages dans l'ordre inverse                          |
| sync-mailbox       | \$    | sauver les modifications de la boîte aux lettres                 |
| tag-pattern        | T     | marquer les messages correspondant à un motif                    |
| tag-thread         | ESC t | marquer la discussion courante                                   |
| toggle-new         | N     | inverser l'indicateur 'nouveau' d'un message                     |
| toggle-write       | %     | changer l'option de mise à jour de la boîte aux lettres          |
| undelete-message   | u     | récupérer l'entrée courante                                      |
| undelete-pattern   | U     | récupérer les messages correspondant à un motif                  |
| undelete-subthread | ESC u | récupérer tous les messages de la sous-discussion                |
| undelete-thread    | ^U    | récupérer tous les messages de la discussion                     |
| untag-pattern      | ^T    | démarquer les messages correspondant à un motif                  |
| view-attachments   | v     | afficher les attachements MIME                                   |

### 6.4.3 pager

|                        |           |   |
|------------------------|-----------|---|
| bottom                 | not bound | aller à la fin du message                                   |
| bounce-message         | b         | renvoyer un message à un autre utilisateur                  |
| change-folder          | c         | ouvrir un dossier différent                                 |
| change-folder-readonly | ESC c     | ouvrir un dossier différent en lecture seule                |
| copy-message           | C         | copier un message dans un fichier ou une boîte aux lettres  |
| create-alias           | a         | créer un alias à partir de l'expéditeur d'un message        |
| decode-copy            | ESC C     | faire une copie décodée (text/plain)                        |
| decode-save            | ESC s     | faire une copie décodée (text/plain) et effacer             |
| delete-message         | d         | effacer l'entrée courante                                   |
| delete-subthread       | ESC d     | effacer tous les messages dans la sous-discussion           |
| delete-thread          | ^D        | effacer tous les messages dans la discussion                |
| display-address        | @         | afficher l'adresse complète de l'expéditeur                 |
| display-toggle-weed    | h         | afficher le message et inverser la restriction des en-têtes |
| edit                   | e         | éditer le message brut                                      |
| edit-type              | ^E        | éditer le content-type de l'attachement                     |
| enter-command          | :         | entrer une commande muttrc                                  |
| exit                   | i         | retourner à l'index des messages                            |
| extract-keys           | ^K        | extraire les clés publiques PGP                             |
| flag-message           | F         | modifier l'indicateur 'important' d'un message              |
| forget-passphrase      | ^F        | effacer la phrase de passe PGP de la mémoire                |
| forward-message        | f         | faire suivre un message avec des commentaires               |
| group-reply            | g         | répondre à tous les destinataires                           |
| half-up                | not bound | remonter d'1/2 page   |
| half-down              | not bound | descendre d'1/2 page  |
| help                   | ?         | cet écran   |
| list-reply             | L         | répondre à la liste spécifiée                               |
| mail                   | m         | composer un nouveau message                                 |
| mail-key               | ESC k     | envoyer une clé publique PGP                                |
| mark-as-new            | N         | inverser l'indicateur 'nouveau' d'un message                |

|                    |           |  |
|--------------------|-----------|--|
| next-line          | RET       | descendre d'une ligne  |
| next-entry         | J         | aller à l'entrée suivante                                    |
| next-new           | TAB       | aller au nouveau message suivant                             |
| next-page          |           | aller à la page suivante                                     |
| next-subthread     | ESC n     | aller à la sous-discussion suivante                          |
| next-thread        | ^N        | aller à la discussion suivante                               |
| next-undeleted     | j         | aller au message non effacé suivant                          |
| next-unread        | not bound | aller au message non lu suivant                              |
| parent-message     | P         | aller au message père dans la discussion                     |
| pipe-message       |           | passer le message/l'attachement à une commande shell         |
| previous-line      | BackSpace | remonter d'une ligne   |
| previous-entry     | K         | aller à l'entrée précédente                                  |
| previous-new       | not bound | aller au nouveau message précédent                           |
| previous-page      | -         | aller à la page précédente                                   |
| previous-subthread | ESC p     | aller à la sous-discussion précédente                        |
| previous-thread    | ^P        | aller à la discussion précédente                             |
| previous-undeleted | k         | aller au message non effacé précédent                        |
| previous-unread    | not bound | aller au message non lu précédent                            |
| print-message      | p         | imprimer l'entrée courante                                   |
| quit               | Q         | sauver les modifications de la boîte aux lettres et quitter  |
| read-subthread     | ESC r     | marquer la sous-discussion courante comme lue                |
| read-thread        | ^R        | marquer la discussion courante comme lue                     |
| recall-message     | R         | rappeler un message ajourné                                  |
| redraw-screen      | ^L        | effacer l'écran et réafficher                                |
| reply              | r         | répondre à un message  |
| save-message       | s         | sauver le message/l'attachement dans un fichier              |
| search             | /         | rechercher une expression rationnelle                        |
| search-next        | n         | rechercher la prochaine occurrence                           |
| search-opposite    | not bound | rechercher la prochaine occurrence dans la direction opposée |
| search-reverse     | ESC /     | rechercher en arrière une expression rationnelle             |
| search-toggle      | \         | inverser la coloration du motif de recherche                 |
| shell-escape       | !         | exécuter une commande dans un sous-shell                     |
| show-version       | V         | afficher la version de Mutt (numéro et date)                 |
| skip-quoted        | S         | sauter le texte cité   |
| sync-mailbox       | \$        | sauver les modifications de la boîte aux lettres             |
| tag-message        | t         | marquer l'entrée courante                                    |
| toggle-quoted      | T         | inverser l'affichage du texte cité                           |
| top                | ^         | aller au début du message                                    |
| undelete-message   | u         | recupérer l'entrée courante                                  |
| undelete-subthread | ESC u     | recupérer tous les messages de la sous-discussion            |
| undelete-thread    | ^U        | recupérer tous les messages de la discussion                 |
| view-attachments   | v         | afficher les attachements MIME                               |

#### 6.4.4 alias

|                |       |  |
|----------------|-------|--|
| search         | /     | rechercher une expression rationnelle                        |
| search-next    | n     | rechercher la prochaine occurrence                           |
| search-reverse | ESC / | rechercher la prochaine occurrence dans la direction opposée |

## 6.4.5 query

|                 |           |   |
|-----------------|-----------|---|
| create-alias    | a         | créer un alias à partir de l'expéditeur d'un message                |
| mail            | m         | composer un nouveau message   |
| query           | Q         | demander des adresses à un programme externe                        |
| query-append    | A         | ajouter les nouveaux résultats de la requête aux résultats courants |
| search          | /         | rechercher une expression rationnelle                               |
| search-next     | n         | rechercher la prochaine occurrence                                  |
| search-opposite | not bound | rechercher la prochaine occurrence dans la direction opposée        |
| search-reverse  | ESC /     | rechercher en arrière une expression rationnelle                    |

## 6.4.6 attach

|                     |       |   |
|---------------------|-------|---|
| bounce-message      | b     | renvoyer un message à un autre utilisateur                              |
| delete-entry        | d     | effacer l'entrée courante   |
| display-toggle-weed | h     | afficher le message et inverser la restriction des en-têtes             |
| edit-type           | ^E    | éditer le content-type de l'attachement                                 |
| extract-keys        | ^K    | extraire les clés publiques PGP   |
| forward-message     | f     | faire suivre un message avec des commentaires                           |
| group-reply         | g     | répondre à tous les destinataires                                       |
| list-reply          | L     | répondre à la liste spécifiée   |
| pipe-entry          |       | passer le message/l'attachement à une commande shell                    |
| print-entry         | p     | imprimer l'entrée courante  |
| reply               | r     | répondre à un message   |
| resend-message      | ESC e | utiliser le message courant comme modèle pour un nouveau message        |
| save-entry          | s     | sauver le message/l'attachement dans un fichier                         |
| undelete-entry      | u     | récupérer l'entrée courante   |
| view-attach         | RET   | visualiser l'attachement en utilisant l'entrée mailcap si nécessaire    |
| view-mailcap        | m     | forcer la visualisation d'un attachment en utilisant le fichier mailcap |
| view-text           | T     | visualiser un attachment en tant que texte                              |

## 6.4.7 compose

|                     |       |   |
|---------------------|-------|---|
| attach-file         | a     | attacher un fichier à ce message                                      |
| attach-message      | A     | attacher un message à ce message                                      |
| attach-key          | ESC k | attacher une clé publique PGP   |
| copy-file           | C     | sauver le message/l'attachement dans un fichier                       |
| detach-file         | D     | effacer l'entrée courante   |
| display-toggle-weed | h     | afficher le message et inverser la restriction des en-têtes           |
| edit-bcc            | b     | éditer la liste BCC   |
| edit-cc             | c     | éditer la liste CC  |
| edit-description    | d     | éditer la description de l'attachement                                |
| edit-encoding       | ^E    | edit attachment transfer-encoding                                     |
| edit-fcc            | f     | entrer le nom d'un fichier dans lequel sauver une copie de ce message |
| edit-from           | ESC f | éditer le champ from  |
| edit-file           | ^X e  | éditer le fichier à attacher  |
| edit-headers        | E     | éditer le message avec ses en-têtes                                   |
| edit                | e     | éditer le message   |
| edit-mime           | m     | éditer l'attachement en utilisant l'entrée mailcap                    |
| edit-reply-to       | r     | éditer le champ Reply-To  |

|                   |     |  |
|-------------------|-----|--|
| edit-subject      | s   | éditer l'objet de ce message   |
| edit-to           | t   | éditer la liste TO   |
| edit-type         | ^T  | éditer le content-type de l'attachement                              |
| filter-entry      | F   | filtrer un attachement au moyen d'une commande shell                 |
| forget-passphrase | ^F  | effacer la phrase de passe PGP de la mémoire                         |
| ispell            | i   | lancer ispell sur le message   |
| new-mime          | n   | composer un nouvel attachement en utilisant l'entrée mailcap         |
| pgp-menu          | p   | afficher les options PGP   |
| pipe-entry        |     | passer le message/l'attachement à une commande shell                 |
| postpone-message  | P   | sauvegarder ce message pour l'envoyer plus tard                      |
| print-entry       | l   | imprimer l'entrée courante   |
| rename-file       | R   | renommer/déplacer un fichier attaché                                 |
| send-message      | y   | envoyer le message   |
| toggle-unlink     | u   | inverser l'option de suppression de fichier après envoi              |
| view-attach       | RET | visualiser l'attachement en utilisant l'entrée mailcap si nécessaire |
| write-fcc         | w   | sauvegarder le message dans un dossier                               |

#### 6.4.8 postpone

|                |   |                             |
|----------------|---|-----------------------------|
| delete-entry   | d | effacer l'entrée courante   |
| undelete-entry | u | récupérer l'entrée courante |

#### 6.4.9 browser

|                   |       |  |
|-------------------|-------|--|
| change-dir        | c     | changer de répertoires   |
| check-new         | TAB   | vérifier la présence de nouveaux messages dans les boîtes aux lettres    |
| enter-mask        | m     | entrer un masque de fichier  |
| search            | /     | rechercher une expression rationnelle                                    |
| search-next       | n     | rechercher la prochaine occurrence                                       |
| search-reverse    | ESC / | rechercher en arrière une expression rationnelle                         |
| select-new        | N     | sélectionner un nouveau fichier dans ce répertoire                       |
| sort              | o     | trier les messages   |
| sort-reverse      | O     | trier les messages dans l'ordre inverse                                  |
| toggle-mailboxes  | TAB   | changer entre l'affichage des BAL et celui de tous les fichiers          |
| view-file         | SPACE | visualiser le fichier  |
| subscribe         | s     | s'abonner à la boîte aux lettres courante (IMAP seulement)               |
| unsubscribe       | u     | se désabonner de la boîte aux lettres courante (IMAP seulement)          |
| toggle-subscribed | T     | changer entre voir toutes les BAL/voir les BAL abonnées (IMAP seulement) |

#### 6.4.10 pgp

|            |   |  |
|------------|---|--|
| view-name  | % | afficher le numéro d'utilisateur de la clé |
| verify-key | c | vérifier une clé publique PGP              |

#### 6.4.11 editor

|               |           |   |
|---------------|-----------|---|
| backspace     | BackSpace | effacer le caractère situé devant le curseur      |
| backward-char | ^B        | déplacer le curseur d'un caractère vers la gauche |
| backward-word | ESC b     | déplacer le curseur d'un mot vers la gauche       |

|                 |           |   |
|-----------------|-----------|---|
| bol             | ^A        | aller au début de la ligne                                    |
| buffy-cycle     | Space     | parcourir les boîtes aux lettres recevant du courrier         |
| capitalize-word | ESC c     | convertir en majuscule le premier caractère du mot            |
| complete        | TAB       | compléter un nom de fichier ou un alias                       |
| complete-query  | ^T        | compléter une adresse grâce à une requête                     |
| delete-char     | ^D        | effacer le caractère situé sous le curseur                    |
| downcase-word   | ESC l     | convertir le mot courant en minuscules                        |
| eol             | ^E        | aller en fin de ligne   |
| forward-char    | ^F        | déplacer le curseur d'un caractère vers la droite             |
| forward-word    | ESC f     | déplacer le curseur jusqu'au prochain mot                     |
| history-down    | not bound | remonter dans l'historique                                    |
| history-up      | not bound | remonter dans l'historique                                    |
| kill-eol        | ^K        | effacer la fin de la ligne à partir du curseur                |
| kill-eow        | ESC d     | effacer la fin du mot à partir du curseur                     |
| kill-line       | ^U        | effacer tous les caractères de la ligne                       |
| kill-word       | ^W        | effacer le mot situé devant le curseur                        |
| quote-char      | ^V        | entrer le caractère correspondant à la prochaine touche       |
| transpose-chars | not bound | échanger le caractère situé sous le curseur avec le précédent |
| upcase-word     | ESC u     | convertir en majuscule toutes les lettres du mot              |

## 7 Divers

### 7.1 Remerciements

Kari Hurтта <kari.hurtta@fmi.fi>

a co-développé le code d'analyse MIME originel à l'époque de ELM-ME.

Les personnes suivantes ont été très utiles au développement de Mutt:

Vikas Agnihotri <vikasa@writeme.com> ,  
 Francois Berjon <Francois.Berjon@aar.alcatel-alsthom.fr> ,  
 Aric Blumer <aric@fore.com> ,  
 John Capo <jc@irbs.com> ,  
 David Champion <dgc@uchicago.edu> ,  
 Brendan Cully <brendan@kublai.com> ,  
 Liviu Daia <daia@stoilow.imar.ro> ,  
 Thomas E. Dickey <dickey@herndon4.his.com> ,  
 David DeSimone <fox@convex.hp.com> ,  
 Nikolay N. Dudorov <nnd@wint.itfs.nsk.su> ,  
 Ruslan Ermilov <ru@freebsd.org> ,  
 Edmund Grimley Evans <edmund@rano.org> ,  
 Michael Finken <finken@conware.de> ,  
 Sven Guckes <guckes@math.fu-berlin.de> ,  
 Lars Hecking <lhecking@nmrc.ie> ,  
 Mark Holloman <holloman@nando.net> ,  
 Andreas Holzmann <holzmann@fmi.uni-passau.de> ,  
 Marco d'Itri <md@linux.it> ,  
 Björn Jacke <bjacke@suse.com> ,  
 Byrial Jensen <byrial@image.dk> ,  
 David Jeske <jeske@igcom.net> ,

Christophe Kalt <kalt@hugo.int-evry.fr> ,  
Tommi Komulainen <Tommi.Komulainen@iki.fi> ,  
Felix von Leitner (a.k.a “Fefe”) <leitner@math.fu-berlin.de> ,  
Brandon Long <blong@fiction.net> ,  
Jimmy Mäkelä <jmy@flashback.net> ,  
Lars Marowsky-Bree <lmb@pointer.in-minden.de> ,  
Thomas “Mike” Michlmayr <mike@cosy.sbg.ac.at> ,  
Andrew W. Nosenko <awn@bcs.zp.ua> ,  
David O’Brien <obrien@Nuxi.cs.ucdavis.edu> ,  
Clint Olsen <olsenc@ichips.intel.com> ,  
Park Myeong Seok <pms@romance.kaist.ac.kr> ,  
Thomas Parmelan <tom@ankh.fr.eu.org> ,  
Ollivier Robert <roberto@keltia.freenix.fr> ,  
Thomas Roessler <roessler@does-not-exist.org> ,  
Roland Rosenfeld <roland@spinnaker.de> ,  
TAKIZAWA Takashi <taki@luna.email.ne.jp> ,  
Allain Thivillon <Allain.Thivillon@alma.fr> ,  
Gero Treuner <gero@faveve.uni-stuttgart.de> ,  
Vsevolod Volkov <vvv@lucky.net> ,  
Ken Weinert <kenw@ihs.com>

## 7.2 À propos de cette documentation

Ce document a été écrit en SGML, puis mis en page à l’aide du paquet [sgml-tools](#) .