

Der Mailreader Mutt

von Michael Elkins <*me@cs.hmc.edu*>

Version 1.4

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

Inhaltsverzeichnis

1	Einführung	10
1.1	Die Homepage von Mutt	10
1.2	Mailing Listen	10
1.3	Quellen für die Software-Distribution	11
1.4	IRC	11
1.5	USENET	11
1.6	Copyright	11
2	Einstieg	11
2.1	Navigieren in den Menüs	12
2.2	Editieren von Eingabefeldern	12
2.3	Mail lesen - Der Index und der Pager	13
2.3.1	Der Nachrichten Index	13
2.3.2	Der Pager	14
2.3.3	Thread Modus	15
2.3.4	Verschiedene Funktionen	16
2.4	Mail versenden	17
2.4.1	Editieren der Kopfzeilen	19
2.4.2	Mutt mit PGP verwenden	19
2.4.3	Verschicken anonymer Nachrichten über “mixmaster”	20
2.5	Nachrichten weiterleiten	20
2.6	Nachrichten zurückstellen	21
3	Konfiguration	21
3.1	Syntax der Konfigurationsdateien	22
3.2	Definieren und Benutzen von Aliasnamen	23
3.3	Ändern der Standard Tastenbelegung	23
3.4	Aliasnamen für Zeichensätze definieren	25
3.5	Von einer Mailbox abhängig Variablen setzen	25
3.6	Tastatur Makros	25

3.7	Verwendung von Farb- und Mono Video Attributen	26
3.8	Ignorieren (weeding) von nicht gewünschten Kopfzeilen	28
3.9	Mailinglisten	28
3.10	Mehrere Spool-Mailboxen verwenden	29
3.11	Mailboxen angeben, die Nachrichten empfangen	29
3.12	Benutzerdefinierte Kopfzeilen	29
3.13	Sortierung der Kopfzeilen in der Nachrichtenanzeige festlegen	30
3.14	Standard-Datei zum Speichern festlegen	30
3.15	Standard Fcc: Mailbox beim Verfassen festlegen	31
3.16	Standard-Dateiname zum Speichern und Fcc: Mailbox zusammen festlegen	31
3.17	Einstellungen abhängig vom Empfänger ändern	31
3.18	Verändere Einstellungen, bevor eine Nachricht formatiert wird	31
3.19	Auswahl des PGP Schlüssels für einen Empfänger	32
3.20	Befehlssequenzen in den Tastaturpuffer schreiben	32
3.21	Ausführen von Funktionen	32
3.22	Nachrichten bewerten (Scoring)	32
3.23	Der Umgang mit Konfigurations-Variablen	33
3.24	Lesen von Konfigurations-Kommandos aus einer anderen Datei	33
3.25	Entfernen von Hooks	34
4	Fortgeschrittene Anwendung	34
4.1	Reguläre Ausdrücke	34
4.2	Suchmuster	37
4.2.1	Anpassen von Suchmustern	38
4.2.2	Komplexe Muster	38
4.2.3	Suchen nach Datum	39
4.3	Verwendung von Markierungen (Tags)	40
4.4	Das Arbeiten mit Hooks (“Haken”)	40
4.4.1	Auswählen von Nachrichten durch Hooks	40
4.5	Adressbuch-Einträge aus externen Quellen	41
4.6	Mailbox-Formate	42
4.7	Kurzschreibweisen für Mailboxen	42
4.8	Der Umgang mit Mailinglisten	42
4.9	Unterstützung von Delivery Status Notification (DSN)	44
4.10	POP3 Unterstützung (OPTIONAL)	44
4.11	IMAP-Unterstützung (OPTIONAL)	44
4.11.1	Auswahl der Mailboxen	45

4.11.2	Anmeldung am Server	45
4.12	Mehrere IMAP/POP Konten verwalten (OPTIONAL)	46
4.13	URLs mit dem Webbrowser anschauen (EXTERN)	46
5	Die MIME-Unterstützung	47
5.1	Verwendung von MIME in Mutt	47
5.1.1	MIME-Nachrichten im Pager anzeigen	47
5.1.2	Das Anhänge-Menü	47
5.1.3	Das “Nachricht erstellen”-Menü	48
5.2	Konfiguration der MIME-Typen in <code>mime.types</code>	48
5.3	Konfiguration der MIME-Anzeige in <code>mailcap</code>	49
5.3.1	Die Grundlagen der Mailcap-Datei	49
5.3.2	Sichere Verwendung der mailcap	50
5.3.3	Fortgeschrittene Nutzung von mailcap	50
5.3.4	Beispiele für mailcap-Dateien	53
5.4	MIME-Autoview	54
5.5	MIME Multipart/Alternative	54
6	Referenz	55
6.1	Optionen an der Kommandozeile	55
6.2	Konfigurations-Kommandos	55
6.3	Konfigurations-Variablen	57
6.3.1	<code>abort_nosubject</code>	57
6.3.2	<code>abort_unmodified</code>	57
6.3.3	<code>alias_file</code>	57
6.3.4	<code>alias_format</code>	57
6.3.5	<code>allow_8bit</code>	58
6.3.6	<code>allow_ansi</code>	58
6.3.7	<code>alternates</code>	58
6.3.8	<code>arrow_cursor</code>	58
6.3.9	<code>ascii_chars</code>	59
6.3.10	<code>askbcc</code>	59
6.3.11	<code>askcc</code>	59
6.3.12	<code>attach_format</code>	59
6.3.13	<code>attach_sep</code>	60
6.3.14	<code>attach_split</code>	60
6.3.15	<code>attribution</code>	60
6.3.16	<code>autoedit</code>	60

6.3.17	auto_tag	61
6.3.18	beep	61
6.3.19	beep_new	61
6.3.20	bounce_delivered	61
6.3.21	charset	61
6.3.22	check_new	61
6.3.23	collapse_unread	62
6.3.24	uncollapse_jump	62
6.3.25	compose_format	62
6.3.26	confirmappend	62
6.3.27	confirmcreate	62
6.3.28	connect_timeout	63
6.3.29	copy	63
6.3.30	date_format	63
6.3.31	default_hook	63
6.3.32	delete	63
6.3.33	delete_untag	64
6.3.34	digest_collapse	64
6.3.35	display_filter	64
6.3.36	dotlock_program	64
6.3.37	dsn_notify	64
6.3.38	dsn_return	64
6.3.39	duplicate_threads	65
6.3.40	edit_headers	65
6.3.41	editor	65
6.3.42	encode_from	65
6.3.43	envelope_from	65
6.3.44	escape	66
6.3.45	fast_reply	66
6.3.46	fcc_attach	66
6.3.47	fcc_clear	66
6.3.48	folder	66
6.3.49	folder_format	66
6.3.50	followup_to	67
6.3.51	force_name	68
6.3.52	forward_decode	68
6.3.53	forward_format	68

6.3.54	forward_quote	68
6.3.55	from	68
6.3.56	gecos_mask	68
6.3.57	hdrs	69
6.3.58	header	69
6.3.59	help	69
6.3.60	hidden_host	69
6.3.61	hide_limited	70
6.3.62	hide_missing	70
6.3.63	hide_top_limited	70
6.3.64	hide_top_missing	70
6.3.65	history	70
6.3.66	honor_followup_to	70
6.3.67	hostname	70
6.3.68	ignore_list_reply_to	71
6.3.69	imap_authenticators	71
6.3.70	imap_delim_chars	71
6.3.71	imap_force_ssl	71
6.3.72	imap_home_namespace	71
6.3.73	imap_keepalive	72
6.3.74	imap_list_subscribed	72
6.3.75	imap_pass	72
6.3.76	imap_passive	72
6.3.77	imap_peek	72
6.3.78	imap_servernoise	73
6.3.79	imap_user	73
6.3.80	implicit_autoview	73
6.3.81	include	73
6.3.82	indent_string	73
6.3.83	index_format	73
6.3.84	ispell	76
6.3.85	keep_flagged	76
6.3.86	locale	76
6.3.87	mail_check	76
6.3.88	mailcap_path	76
6.3.89	mailcap_sanitize	76
6.3.90	maildir_trash	77

6.3.91 mark_old	77
6.3.92 markers	77
6.3.93 mask	77
6.3.94 mbox	77
6.3.95 mbox_type	78
6.3.96 metoo	78
6.3.97 menu_scroll	78
6.3.98 meta_key	78
6.3.99 mh_purge	78
6.3.100 mh_seq_flagged	78
6.3.101 mh_seq_replied	79
6.3.102 mh_seq_unseen	79
6.3.103 mime_forward	79
6.3.104 mime_forward_decode	79
6.3.105 mime_forward_rest	79
6.3.106 mix_entry_format	79
6.3.107 mixmaster	80
6.3.108 move	80
6.3.109 message_format	80
6.3.110 pager	80
6.3.111 pager_context	81
6.3.112 pager_format	81
6.3.113 pager_index_lines	81
6.3.114 pager_stop	81
6.3.115 pgp_autosign	81
6.3.116 pgp_autoencrypt	81
6.3.117 pgp_ignore_subkeys	82
6.3.118 pgp_entry_format	82
6.3.119 pgp_good_sign	83
6.3.120 pgp_long_ids	83
6.3.121 pgp_replyencrypt	83
6.3.122 pgp_replysign	83
6.3.123 pgp_replysignencrypted	83
6.3.124 pgp_retainable_sigs	83
6.3.125 pgp_show_unusable	84
6.3.126 pgp_sign_as	84
6.3.127 pgp_strict_enc	84

6.3.128	pgp_timeout	84
6.3.129	pgp_verify_sig	84
6.3.130	pgp_sort_keys	84
6.3.131	pgp_create_traditional	85
6.3.132	pgp_decode_command	85
6.3.133	pgp_getkeys_command	86
6.3.134	pgp_verify_command	86
6.3.135	pgp_decrypt_command	86
6.3.136	pgp_clearsign_command	86
6.3.137	pgp_sign_command	86
6.3.138	pgp_encrypt_sign_command	86
6.3.139	pgp_encrypt_only_command	87
6.3.140	pgp_import_command	87
6.3.141	pgp_export_command	87
6.3.142	pgp_verify_key_command	87
6.3.143	pgp_list_secring_command	87
6.3.144	pgp_list_pubring_command	87
6.3.145	forward_decrypt	88
6.3.146	ssl_starttls	88
6.3.147	certificate_file	88
6.3.148	ssl_usesystemcerts	88
6.3.149	entropy_file	88
6.3.150	ssl_use_sslv2	88
6.3.151	ssl_use_sslv3	89
6.3.152	ssl_use_tlsv1	89
6.3.153	pipe_split	89
6.3.154	pipe_decode	89
6.3.155	pipe_sep	89
6.3.156	pop_authenticators	89
6.3.157	pop_auth_try_all	90
6.3.158	pop_checkinterval	90
6.3.159	pop_delete	90
6.3.160	pop_host	90
6.3.161	pop_last	90
6.3.162	pop_reconnect	90
6.3.163	pop_user	91
6.3.164	pop_pass	91

6.3.165 post_indent_string	91
6.3.166 postpone	91
6.3.167 postponed	91
6.3.168 preconnect	91
6.3.169 print	92
6.3.170 print_command	92
6.3.171 print_decode	92
6.3.172 print_split	92
6.3.173 prompt_after	93
6.3.174 query_command	93
6.3.175 quit	93
6.3.176 quote_regexp	93
6.3.177 read_inc	93
6.3.178 read_only	94
6.3.179 realname	94
6.3.180 recall	94
6.3.181 record	94
6.3.182 reply_regexp	94
6.3.183 reply_self	94
6.3.184 reply_to	95
6.3.185 resolve	95
6.3.186 reverse_alias	95
6.3.187 reverse_name	95
6.3.188 reverse_realname	96
6.3.189 rfc2047_parameters	96
6.3.190 save_address	96
6.3.191 save_empty	96
6.3.192 save_name	96
6.3.193 score	97
6.3.194 score_threshold_delete	97
6.3.195 score_threshold_flag	97
6.3.196 score_threshold_read	97
6.3.197 send_charset	97
6.3.198 sendmail	98
6.3.199 sendmail_wait	98
6.3.200 shell	98
6.3.201 sig_dashes	98

6.3.202 sig_on_top	99
6.3.203 signature	99
6.3.204 simple_search	99
6.3.205 smart_wrap	99
6.3.206 smileys	99
6.3.207 sleep_time	99
6.3.208 sort	100
6.3.209 sort_alias	100
6.3.210 sort_aux	100
6.3.211 sort_browser	101
6.3.212 sort_re	101
6.3.213 spoolfile	101
6.3.214 status_chars	101
6.3.215 status_format	102
6.3.216 status_on_top	103
6.3.217 strict_threads	104
6.3.218 suspend	104
6.3.219 text_flowed	104
6.3.220 thread_received	104
6.3.221 thorough_search	104
6.3.222 tilde	104
6.3.223 timeout	105
6.3.224 tmpdir	105
6.3.225 to_chars	105
6.3.226 tunnel	105
6.3.227 use_8bitmime	105
6.3.228 use_domain	106
6.3.229 use_from	106
6.3.230 user_agent	106
6.3.231 visual	106
6.3.232 wait_key	106
6.3.233 weed	107
6.3.234 wrap_search	107
6.3.235 wrapmargin	107
6.3.236 write_inc	107
6.3.237 write_bcc	107
6.4 Funktionen	107

6.4.1	allgemein (generic)	108
6.4.2	Index	108
6.4.3	Pager	110
6.4.4	Adressbuch (aliases)	112
6.4.5	Adressenabfrage (query)	112
6.4.6	Anhänge (attach)	112
6.4.7	Nachricht erstellen (compose)	112
6.4.8	Nachricht zurückstellen (postpone)	113
6.4.9	Datei aus Liste auswählen (browser)	113
6.4.10	PGP	114
6.4.11	Editor	114
7	Diverses	114
7.1	Anerkennungen	114
7.2	Über dieses Dokument	115
7.3	Über die vorliegende Übersetzung	115

1 Einführung

Mutt ist ein kleiner, aber mächtiger, textbasierter und MIME-fähiger Mailclient. Mutt ist extrem konfigurierbar. Aufgrund seiner fortgeschrittenen Fähigkeiten wie frei definierbarer Tastenbelegung, Tastatur-Makros, “Threading” von Nachrichten, Verwendung von regulären Ausdrücken und einer mächtigen Suchsprache für die Gruppierung von Nachrichten ist er für Benutzer gut geeignet, die große Mengen an Mail zu bewältigen haben.

1.1 Die Homepage von Mutt

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

1.2 Mailing Listen

Sende eine Mail mit dem Wort *subscribe* im Body an `listen-name-request@mutt.org`, um eine der folgenden Mailinglisten zu abonnieren.

- *mutt-announce@mutt.org* – Liste für Ankündigungen mit geringem Mailaufkommen.
- *mutt-users@mutt.org* – Der Ort für Hilfestellungen, Fehlermeldungen und Wünsche
- *mutt-dev@mutt.org* – Die Entwickler-Mailingliste

Anmerkung: Alle Mails an *mutt-announce* werden automatisch an *mutt-users* weitergeleitet, man muss also nicht beide Listen abonnieren.

Anmerkung der Übersetzer: Die genannten Listen sind englischsprachig. Eine deutsche Mailing-Liste zu mutt ist uns nicht bekannt.

1.3 Quellen für die Software-Distribution

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

Eine Liste der Server, die das mutt-Archiv spiegeln, ist auf <http://www.mutt.org/download.html> erhältlich.

1.4 IRC

Besuche den Channel `#mutt` im *OpenProjects.Net* (www.openprojects.net), um mit anderen an Mutt interessierten Personen zu chatten.

1.5 USENET

Schau mal in die Newsgruppe *comp.mail.mutt* vorbei. **Anmerkung der Übersetzer:** Im deutschsprachigen Teil des Usenet ist mutt in *de.comm.software.mailreader.misc* on topic.

1.6 Copyright

Mutt ist Copyright (C) 1996-2000 Michael R. Elkins <me@cs.hmc.edu> und andere.

Dieses Programm ist freie Software; Es kann unter den Bedingungen der "GNU General Public License", wie sie von der Free Software Foundation veröffentlicht wurde, weitergegeben und/oder verändert werden; es gilt entweder die Version 2 der Lizenz oder (wenn gewünscht) jede neuere Version.

Dieses Programm wird in der Hoffnung verteilt, daß es nützlich ist, jedoch wird KEINE GARANTIE übernommen. Darüber hinaus gibt es keine Garantie bezüglich QUALITÄT und ANWENDBARKEIT FÜR EINEN BESTIMMTEN ZWECK. Lies die GNU General Public License für weitere Details.

Zusammen mit diesem Programm solltest du eine Kopie der GNU General Public License erhalten haben; wenn nicht, schreib an die Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111, USA.

Die Übersetzer möchten darauf hinweisen, daß diese Übersetzung weder von der FSF autorisiert ist, noch irgend einen juristisch Wert hat. Im Zweifelsfall ist das englische Original allein ausschlaggebend.

2 Einstieg

Dieses Kapitel soll einen kurzen Überblick über die Benutzung von Mutt liefern. Es gibt wesentlich mehr Features, die an anderer Stelle dieses Manuals beschrieben sind. Weitere Informationen sind in der Mutt FAQ und auf diversen Web-Seiten nachzulesen. Genauerer kann man auf der *Mutt Homepage* erfahren.

Bei den in diesem Kapitel beschriebenen Tastenbelegungen handelt es sich um die in der Distribution voreingestellten Werte. Möglicherweise hat dein lokaler Systemadministrator die Belegungen auf dem von dir benutzten Rechner geändert. Du kannst in jedem Menü "?" aufrufen, um dir die aktuell vorhandene Tastenbelegung anzusehen.

Um mit mutt zu arbeiten, musst du es als erstes aufrufen. Das geschieht einfach durch die Eingabe von "mutt" auf der Kommandozeile. Es gibt diverse Kommandozeilen-Optionen; dazu verraten dir entweder die mutt-Manpage oder die 6.1 (Referenz) mehr.

2.1 Navigieren in den Menüs

Informationen werden ähnlich wie bei ELM in Menüs dargestellt. Die folgende Tabelle zeigt die wesentlichen Tasten, die zum Navigieren in den Menüs von Mutt benutzt werden.

j oder <Cursor runter>	next-entry	Gehe zum nächsten Eintrag
k oder <Cursor rauf>	previous-entry	Gehe zum vorherigen Eintrag
z oder <Bild runter>	page-down	Gehe zur nächsten Seite
Z oder <Bild rauf>	page-up	Gehe zur vorherigen Seite
= oder <Pos1>	first-entry	Gehe zum ersten Eintrag
* oder <Ende>	last-entry	Gehe zum letzten Eintrag
q	quit	Verlasse das aktuelle Menü
?	help	Zeige alle Tastenbelegungen für das aktuelle Menü

2.2 Editieren von Eingabefeldern

Mutt verfügt über einen eingebauten Zeileneditor, der für die Eingabe von Daten wie E-Mail-Adressen oder Dateinamen benutzt wird. Die Tastaturbelegung für das Navigieren beim Editieren ist der des Editors Emacs sehr ähnlich.

^A oder <Pos1>	bol	Gehe zum Zeilenanfang
^B oder <Cursor links>	backward-char	Gehe ein Zeichen nach links
ESC B	backward-word	Gehe ein Wort nach links
ESC c	capitalize-word	Mache ersten Buchstaben des Worts groß
^D oder <Entf>	delete-char	Lösche das Zeichen unter dem Cursor
ESC d	kill-eow	Lösche bis zum Ende des Worts
^E oder <Ende>	eol	Gehe zum Zeilenende
^F oder <Cursor rechts>	forward-char	Gehe ein Zeichen nach rechts
ESC F	forward-word	Gehe ein Wort nach rechts
<TAB>	complete	Datei- oder Aliasnamen ergänzen
^K	kill-eol	Lösche bis zum Zeilenende
ESC l	downcase-word	Konvertiere Wort in Kleinbuchstaben
^T	complete-query	Adresse aus externer Quelle ergänzen
^U	kill-line	Lösche die gesamte Zeile
ESC u	upcase-word	Konvertiere Wort in Großbuchstaben
^V	quote-char	Nächste Taste "wörtlich" übernehmen
^W	kill-word	Lösche das Wort rechts vom Cursor
<Cursor rauf>	history-up	Vorherigen Text aus History laden
<Cursor runter>	history-down	Nächsten Text aus History laden
<Backspace>	backspace	Lösche das Zeichen links vom Cursor
^G	n/a	Abbruch
<Return>	n/a	Editieren beenden

Du kannst die Tastaturbelegung für die *editor* Funktionen mit Hilfe des 3.3 (bind) Befehls ändern. Um zum Beispiel die *Entf* Taste dazu zu benutzen, das Zeichen links vom Cursor zu löschen (an Stelle des Zeichens unter dem Cursor), kannst du folgendes tun:

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

2.3 Mail lesen - Der Index und der Pager

Wie bei vielen anderen Mailreadern auch, gibt es bei Mutt zwei Modi, in denen Mail gelesen wird. Der erste ist die Übersicht über alle Nachrichten in der Mailbox, dieser wird als "Index" bezeichnet. Der zweite Modus ist die Anzeige des Inhalts einer Mail, diesen nennt man den "Pager".

Die nächsten Abschnitte beschreiben die Funktionen, die in jedem dieser Modi zur Verfügung stehen.

2.3.1 Der Nachrichten Index

c	Öffne eine andere Mailbox
ESC c	Öffne eine andere Mailbox im Nur-Lesen-Modus
C	Kopiere Nachricht in Datei/Mailbox
ESC C	Erzeuge decodierte Kopie (text/plain)
ESC s	Erzeuge decodierte Kopie (text/plain) und lösche
D	Lösche Nachrichten nach Muster
d	Lösche aktuellen Eintrag
F	Schalte 'Wichtig'-Markierung der Nachricht um
l	Wähle anzuzeigende Nachrichten mit Muster aus
N	Setze/entferne den "neu"-Indikator einer Nachricht
o	Sortiere Nachrichten
O	Sortiere Nachrichten in umgekehrter Reihenfolge
q	Speichere Änderungen in Mailbox und beende das Programm
s	Speichere Nachricht/Anhang in Datei
t	Markiere aktuellen Eintrag
ESC t	Markiere aktuellen Diskussionsfaden
T	Markiere Nachrichten nach Muster
u	Entferne Löschkmarkierung vom aktuellen Eintrag
U	Entferne Löschkmarkierung nach Muster
v	Zeige MIME-Anhänge
x	Menü verlassen
<Return>	Zeige Nachricht an
<Tab>	Springe zur nächsten neuen Nachricht
@	Zeige komplette Absenderadresse
\$	Speichere Änderungen in Mailbox
/	Suche nach regulärem Ausdruck
ESC /	Suche rückwärts nach regulärem Ausdruck
^L	Erzeuge Bildschirmanzeige neu
^T	Entferne Markierung nach Muster

Status Bezeichner Zusätzlich zu Autor und Subject wird neben der laufenden Nummer eine kurze Zusammenfassung über den Status der Nachricht ausgegeben. Folgende "Bezeichner" können dort erscheinen; sie haben die jeweils angegebene Bedeutung:

D

Nachricht wurde gelöscht

K

Enthält einen öffentlichen PGP-Schlüssel

M

Benötigt mailcap für die Anzeige

N

Nachricht ist neu und ungelesen

O

Nachricht ist alt und ungelesen

P

Nachricht ist mit PGP verschlüsselt

r

Nachricht ist beantwortet

S

Nachricht ist mit PGP signiert

s

Nachricht ist mit PGP signiert, aber nicht überprüft

!

Nachricht ist als wichtig markiert

*

Nachricht ist ausgewählt

Einige der Status Bezeichner können wie folgt ein- und ausgeschaltet werden:

- **Bezeichner setzen:** **set-flag** (Voreinstellung: w)
- **Bezeichner löschen:** **clear-flag** (Voreinstellung: W)

Desweiteren zeigen die folgenden Bezeichner an, an wen die Nachricht adressiert ist. Sie können über die Variable 6.3.225 (\$to_chars) konfiguriert werden.

```
+      Nachricht ging an dich und nur an dich
T      Nachricht ging an dich, aber per To oder per Cc auch an andere
C      Nachricht ging per Cc an dich
F      Nachricht ist von dir
L      Nachricht ging an eine abonnierte Mailing-Liste
```

2.3.2 Der Pager

Standardmäßig verwendet Mutt seinen eingebauten Pager, um den Inhalt von Nachrichten anzuzeigen. Dieser Pager ist dem Unix-Programm *less* sehr ähnlich, aber längst nicht so umfangreich.

```
<Return>      Gehe eine Zeile nach unten
<Space>      Zeige die nächste Seite (oder die nächste Nachricht,
              wenn am Ende der Nachricht)
-            Gehe zurück zur vorherigen Seite
n            Zeige die nächste Nachricht an
S            Übergehe zitierten Text
T            Schalte Anzeige von zitiertem Text ein/aus
?            Zeige die Tastaturbelegung
/            Suche nach einem regulären Ausdruck (Suchmuster)
```

ESC /	Suche rückwärts nach einem regulären Ausdruck
\	Suchmusterfärbung ein-/ausschalten
^	Springe zum Nachrichtenanfang
\$	Springe zum Nachrichtenende

Zusätzlich sind viele der Funktionen aus dem Index auch im Pager verfügbar, wie z.B. *delete-message* oder *copy-message* (Dieses ist ein Vorteil gegenüber der Verwendung eines externen Pagers).

Außerdem unterstützt der interne Pager diverse andere erweiterte Funktionen. Zum einen akzeptiert und übersetzt er die "Standard" nroff Kodierungen für fett und unterstrichen. Diese Kodierungen bestehen aus einer Folge von entweder Buchstabe, Backspace (`^H`) und nochmal der Buchstabe für fett oder Buchstabe, Backspace, `"_"` für unterstrichen. Mutt wird versuchen, diese Sequenzen als fett bzw. unterstrichen darzustellen, sofern dein Terminal dieses unterstützt. Wenn nicht, dann kannst du die 3.7 ("color") Objekte für fett und unterstrichen benutzen und ihnen ein Farb- oder Mono-Attribut zuweisen.

Desweiteren unterstützt der interne Pager die ANSI Escape Sequenzen für Zeichenattribute. Mutt übersetzt diese in die korrekten Farb- und Zeicheneigenschaften. Die von Mutt unterstützten Sequenzen lauten:

```
ESC [ Ps;Ps;Ps;...;Ps m
wobei Ps =
0  Alle Attribute aus
1  Fett ein
4  Unterstrichen ein
5  Blinkend ein
7  Invers ein
3x Farbe für Vordergrund ist x
4x Farbe für Hintergrund ist x
```

```
Die Farben sind
0  schwarz
1  rot
2  grün
3  gelb
4  blau
5  magenta
6  cyan
7  weiß
```

Mutt verwendet diese Attribute, um text/enriched Nachrichten darzustellen. Sie können ebenso von einem externen 5.4 (autoview) Skript verwendet werden, um Text hervorzuheben.

Anmerkung: Wenn du die Farben deines Terminals änderst, wirkt sich das auf Mutt aus. Wenn du z.B. im `xterm color2` eine andere Farbe zuordnest, dann verwendet Mutt diese Farbe anstatt grün.

2.3.3 Thread Modus

Wenn die Mailbox nach *threads* 6.3.208 (sortiert) ist, dann gibt es einige weitere Funktionen in den Modi *index* und *pager*.

<code>^D</code>	<code>delete-thread</code>	Lösche alle Nachrichten im aktuellen Thread
<code>^U</code>	<code>undelete-thread</code>	Hebe die Löschung auf für alle Nachrichten im aktuellen Thread
<code>^N</code>	<code>next-thread</code>	Springe zum Anfang des nächsten Threads
<code>^P</code>	<code>previous-thread</code>	Springe zum Anfang des vorherigen Threads

<code>^R</code>	<code>read-thread</code>	Markiere den aktuellen Thread als gelesen
<code>ESC d</code>	<code>delete-subthread</code>	Lösche alle Nachrichten im aktuellen Subthread
<code>ESC u</code>	<code>undelete-subthread</code>	Hebe die Löschung auf für alle Nachrichten im aktuellen Subthread
<code>ESC n</code>	<code>next-subthread</code>	Springe zum Anfang des nächsten Subthreads
<code>ESC p</code>	<code>previous-subthread</code>	Springe zum Anfang des vorherigen Subthreads
<code>ESC r</code>	<code>read-subthread</code>	Markiere den aktuellen Subthread als gelesen
<code>ESC t</code>	<code>tag-thread</code>	Marke setzen/rücksetzen für den aktuellen Thread
<code>ESC v</code>	<code>collapse-thread</code>	Aktuellen Thread ein/ausklappen
<code>ESC V</code>	<code>collapse-all</code>	Alle Threads ein/ausklappen
<code>P</code>	<code>parent-message</code>	Springe zur ‘‘übergeordneten’’ Mail im Thread

Anmerkung: Nach dem Einklappen eines Threads wird nur die erste Nachricht dieses Threads angezeigt; die anderen werden versteckt. Das ist nützlich, wenn Threads so viele Nachrichten enthalten, dass sonst nur wenige von ihnen gleichzeitig angezeigt werden könnten. Siehe dazu `%M` in 6.3.83 (`$index_format`). Beispielsweise kannst du `”%?M?(#%03M)&(%4l)?”` in 6.3.83 (`$index_format`) verwenden, um optional die Anzahl der versteckten Nachrichten anzuzeigen, wenn ein Thread eingeklappt ist.

Vergleiche auch: 6.3.217 (`$strict_threads`).

2.3.4 Verschiedene Funktionen

Erzeuge Aliasnamen (`create-alias`) (Voreinstellung: `a`)

Erzeugt einen neuen Alias (Adressbucheintrag), basierend auf der aktuellen Nachricht (oder erfragt einen neuen). Wenn das Editieren beendet ist, wird ein 3.2 (Alias) Eintrag in die durch 6.3.3 (`$alias_file`) spezifizierte Datei gemacht. **Anmerkung:** Das Setzen von 6.3.3 (`$alias_file`) stellt noch nicht die dort eingetragenen Aliasnamen zur Verfügung, die Datei muss auch 3.24 (eingelassen) werden.

Suche traditionelles PGP (`check-traditional-pgp`) (Voreinstellung: `ESC P`)

Diese Funktion durchsucht die aktuelle Nachricht nach Inhalten, die auf ‘‘traditionelle’’ Weise (d.h. ohne passende MIME-Deklaration) mit PGP signiert oder verschlüsselt wurden. Technisch gesehen verändert sie dazu zeitweilig den MIME ‘‘Content-Type’’ jener Teile der Nachricht, die PGP-Daten enthalten. Damit erzielt sie eine ähnliche Wirkung wie die Funktion 2.3.4 (`edit-type`).

Kopfzeilen anzeigen (`display-toggle-weed`) (Voreinstellung: `h`)

Schaltet das Verbergen der durch die 3.8 (`ignore`) Befehle angegebenen Header-Felder ein/aus.

Editiere Nachricht (`edit`) (Voreinstellung: `e`)

Dieser Befehl (verfügbar im ‘‘Index’’ und im ‘‘Pager’’) erlaubt es, die aktuelle Nachricht in ihrem ‘‘rohen’’ Zustand (so, wie sie in der Mailbox gespeichert ist) zu editieren. Wenn du damit fertig bist, wird die geänderte Nachricht an die aktuelle Mailbox angehängt und das Original zum Löschen markiert.

Content-Type verändern (`edit-type`) (Voreinstellung: `^E` im Attachment-Menü, sowie im Pager und im Index-Menü; `^T` im ‘‘Nachricht erstellen’’ Menü)

Dieser Befehl kann verwendet werden, um zeitweilig den Content-Type eines Anhangs zu verändern. Das ermöglicht es zum Beispiel, falsche Angaben zum Zeichensatz zu korrigieren. Wenn der Befehl aus dem Index oder vom Pager aus aufgerufen wird, bekommst du die Gelegenheit, den Content-Type des ersten (übergeordneten) Anhangs zu ändern. Im 5.1.2 (Attachment-Menü) stehen sämtliche Anhänge zur Auswahl. Die Änderungen sind nicht dauerhaft; sie gehen beim Wechseln der Mailbox verloren.

Dieser Befehl ist ebenfalls im 5.1.3 (Nachricht erstellen-Menü) verfügbar. Dort kann er verwendet werden, um die Eigenschaften des Anhangs einzustellen, den du verschicken willst.

Gib muttrc-Kommando ein (`enter-command`) (Voreinstellung: `“:”`)

Dieser Befehl wird benutzt, um ein beliebiges Kommando auszuführen, das normalerweise in die `muttrc-`

Konfigurationsdatei eingetragen wird. Ein üblicher Anwendungsfall ist das Prüfen von Variablenwerten oder - im Zusammenhang mit 3.6 (Makros) - das direkte Ändern dieser Werte.

Extrahiere öffentliche PGP-Schlüssel (extract-keys) (Voreinstellung: ^K)

Dieser Befehl extrahiert öffentliche PGP-Schlüssel aus der aktuellen oder aus allen markierten Nachricht(en) und fügt sie deinem öffentlichen PGP-Schlüsselbund hinzu.

Entferne PGP-Mantra aus dem Speicher (forget-passphrase) (Voreinstellung: ^F)

Dieser Befehl entfernt das PGP-Mantra aus dem Speicher. Das ist hilfreich, wenn du dich bei der Eingabe vertippt hast.

Antworte an Mailing-Listen (list-reply) (Voreinstellung: L)

Antworte auf die aktuelle oder alle markierten Nachricht(en), wobei nur Adressen verwendet werden, die mit den Befehlen 3.9 (lists oder subscribe) definiert wurden. Wenn die Variable 6.3.66 (\$honor_followup_to) gesetzt ist, wird überdies der Mail-Followup-To Header berücksichtigt. Wenn man diesen Befehl zum Antworten auf Mails aus Mailinglisten benutzt, vermeidet man, dass eine überflüssige Kopie an den Autor der Mail geht.

Bearbeite (pipe) Nachricht oder Anhang mit Shell-Kommando (pipe-message) (Voreinstellung: |)

Erfragt ein externes Unix-Shell-Kommando und piped die aktuelle Nachricht oder alle markierten in diesen Befehl. Die Variablen 6.3.154 (\$pipe_decode), 6.3.153 (\$pipe_split), 6.3.155 (\$pipe_sep) und 6.3.232 (\$wait_key) kontrollieren das genaue Verhalten dieser Funktion.

Nachricht erneut versenden (resend-message) (Voreinstellung: ESC e)

Bei Verwendung von resend-message nimmt mutt die aktuelle Nachricht als Vorlage einer neuen. Diese Funktion ist mit dem Weiterbearbeiten einer zurückgestellten Nachricht vergleichbar, funktioniert jedoch in beliebigen Mailboxen. Sie lässt sich einsetzen, um bequem MIME-Nachrichten unter Beibehaltung ihrer ursprünglichen Struktur weiterzuleiten. Man beachte, dass die Menge der berücksichtigten Kopfzeilen (Header) vom Wert der Variablen 6.3.233 (\$weed) abhängt.

Diese Funktion ist auch im Attachment-Menü verfügbar. Dort kannst du sie verwenden, um auf einfache Weise eine Nachricht neu zu verschicken, die unzustellbar war und daher (als message/rfc822 markiert) an dich zurückgeschickt wurde.

Rufe Kommando in Shell auf (shell-escape) (Voreinstellung: !)

Erfragt ein externes Unix-Shell-Kommando und führt es aus. Die Variable 6.3.232 (\$wait_key) kann benutzt werden um festzulegen, ob Mutt - abhängig vom Returncode des Kommandos - auf einen Tastendruck warten soll, nachdem das Kommando ausgeführt wurde (um etwa das Lesen einer Ausgabe zu ermöglichen).

Schalte Anzeige von zitiertem Text ein oder aus (toggle-quoted) (Voreinstellung: T)

Der *Pager* benutzt die Variable 6.3.176 (\$quote_regexp), um zitierten Text (Quotes) in der Nachricht zu finden. Diese Funktion schaltet die Anzeige des zitierten Texts in der Nachricht ein bzw. aus. Das ist besonders dann nützlich, wenn man sich nur für die Antwort interessiert, aber eine große Menge an zitiertem Text vorhanden ist.

Überspringe zitierten Text (skip-quoted) (Voreinstellung: S)

Diese Funktion springt zur nächsten Zeile, die selbst keinen zitierten Text enthält und auf eine zitierte Zeile folgt.

2.4 Mail versenden

Die folgenden Tastenbelegungen sind im *index* verfügbar, um Mail zu versenden.

m	compose	Erzeuge neue Nachricht
r	reply	Beantworte Nachricht

g	group-reply	Antworte an alle Empfänger
L	list-reply	Antworte an Mailing-Listen
f	forward	Leite Nachricht mit Kommentar weiter
b	bounce	Versende Nachricht erneut an anderen Empfänger
ESC k	mail-key	Verschicke öffentlichen PGP-Schlüssel

“bounce” versendet eine Nachricht so wie sie ist an einen anderen Empfänger. Ein “forward” einer Nachricht erlaubt es, diese noch mit einem eigenen Kommentar zu versehen oder zu verändern. Diese Besonderheiten werden im Kapitel 2.5 (“Nachrichten weiterleiten”) ausführlicher beschrieben.

Mutt geht dann in das *Nachricht erstellen* Menü und fragt die Empfänger ab, die im “To:” Header eingetragen werden sollen. Als nächstes wird nach dem Subject (Betreff) der Nachricht gefragt. Wenn auf eine Mail geantwortet oder eine Nachricht weitergeleitet wird, wird das Feld mit dem ursprünglichen Subject vorbelegt. In den Abschnitten über 6.3.11 (\$askcc), 6.3.10 (\$askbcc), 6.3.16 (\$autoedit), und 6.3.45 (\$fast_reply) steht, wie man die Art und Weise ändern kann, auf die Mutt diese Dinge erfragt.

Mutt startet dann automatisch deinen 6.3.41 (\$editor) für den Nachrichtentext. Wenn die Variable 6.3.40 (\$edit_headers) gesetzt ist, dann werden alle Kopfzeilen (Header) der Nachricht mit in den Editor kopiert. Alle Nachrichten, auf die du antwortest, werden in der Reihenfolge angefügt, in der sie im Index sortiert waren und entsprechend den Variablen 6.3.15 (\$attribution), 6.3.82 (\$indent_string) und 6.3.165 (\$post_indent_string) formatiert. Wenn eine Nachricht mittels “forward” weitergeleitet wird, dann wird eine Kopie der ursprünglichen Nachricht angefügt, es sei denn, die Variable 6.3.103 (\$mime_forward) ist gesetzt. Wenn du eine 6.3.203 (\$signature) definiert hast, wird auch diese an die Nachricht angehängt.

Wenn du mit dem Editieren der Nachricht fertig bist, kehrt Mutt in das *Nachricht erstellen* Menü zurück. Folgende Optionen sind dort möglich:

a	attach-file	Hänge Datei(en) an diese Nachricht an
A	attach-message	Hänge Nachricht(en) an diese Nachricht an
ESC k	attach-key	Hänge öffentlichen PGP-Schlüssel an
d	edit-description	Editiere die Beschreibung des Anhangs
D	detach-file	Entferne Datei/Nachricht als Anhang
t	edit-to	Editiere Empfängerliste (To)
ESC f	edit-from	Editiere das From-Feld
r	edit-reply-to	Editiere Reply-To-Feld
c	edit-cc	Editiere die CC-Liste
b	edit-bcc	Editiere die BCC-Liste
y	send-message	Verschicke Nachricht
s	edit-subject	Editiere die Subject-Zeile
f	edit-fcc	bestimme eine ‘‘Fcc’’ Mailbox
p	pgp-menu	Zeige PGP-Optionen
P	postpone-message	Stelle Nachricht zum späteren Versand zurück
q	quit	Breche das Versenden der Nachricht ab
w	write-fcc	Schreibe Nachricht in Mailbox
i	ispell	Rechtschreibprüfung via ispell
~F	forget-passphrase	Entferne PGP-Mantra aus Speicher

Anmerkung: Die attach-message Funktion fragt nach einer Mailbox, aus der Nachrichten angehängt werden sollen. Dann kannst du darin Mails markieren. Diese werden automatisch an die Nachricht angehängt. Es ist zu beachten, dass bestimmte Operationen, wie das Erstellen einer neuen Nachricht, Antworten, Weiterleiten etc. nicht zur Verfügung stehen, wenn man sich im ‘attach-message’-Modus befindet. Der Platzhalter %r in 6.3.215 (\$status_format) gibt in diesem Fall ein ‘A’ aus, um anzuzeigen, dass man sich im ‘attach-message’-Modus befindet.

2.4.1 Editieren der Kopfzeilen

Beim Editieren der Kopfzeilen (Header) einer Nachricht stehen einige besondere Funktionen zur Verfügung.

Wenn du

Fcc: *Dateiname*

angibst, dann speichert Mutt in der Datei *Dateiname* eine Kopie der ausgehenden Nachricht - so, als ob du die Funktion *edit-fcc* im *Nachricht erstellen* Menü verwendet hättest.

Du kannst Dateien auch durch Angabe von

Attach: *Dateiname* [*Beschreibung*]

an deine Nachricht anhängen. Dabei ist *Dateiname* die Datei, die angehängt werden soll und *Beschreibung* der Text, der als Beschreibung des Anhangs dienen soll.

Wenn du auf eine Nachricht antwortest und dabei die *In-Reply-To:* Zeile aus den Headern entfernst, dann erzeugt Mutt keine *References:* Zeile. So ist es möglich, einen neuen Thread zu beginnen.

Vergleiche auch 6.3.40 (*edit_headers*).

2.4.2 Mutt mit PGP verwenden

Wenn du PGP verwenden möchtest, dann kannst du

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

angeben.

“E” verschlüsselt, “S” signiert und “S<id>” signiert mit dem angegebenen Schlüssel, wobei 6.3.126 (*\$pgp_sign_as*) dauerhaft gesetzt wird.

Wenn Du mutt angewiesen hast, eine Nachricht mit PGP zu verschlüsseln, wird beim Versenden der Mail folgendermaßen ein Schlüssel ausgewählt: Mutt stellt keine Fragen, wenn es um Schlüssel geht, die eine zertifizierte Benutzer-ID haben, die zu einer der Empfänger-Adressen passt. Es gibt jedoch Situationen, in denen mehrere Schlüssel möglich sind, die Benutzer-ID nur schwach zertifiziert ist oder überhaupt kein passender Schlüssel gefunden werden kann.

In diesen Fällen wird ein Menü angezeigt, das eine Liste möglicher Schlüssel enthält, von denen Du einen aussuchen kannst. Wenn du dieses Menü abbrichst oder mutt überhaupt keinen passenden Schlüssel findet, wirst Du nach einer Benutzer-ID gefragt. Diese Abfrage kann man - wie gewohnt - mit *~G* abbrechen. Wenn das geschieht, kehrt mutt zum *Nachricht erstellen*-Menü zurück.

Sobald die Auswahl eines Schlüssels erfolgreich abgeschlossen ist, wird die Nachricht mit dem gewählten öffentlichen Schlüssel kodiert und verschickt.

Die Einträge im Auswahl-Menü für Schlüssel enthalten verschiedene Felder (siehe auch 6.3.118 (*\$pgp_entry_format*)), deren Bedeutung in den meisten Fällen unmittelbar klar ist. Dennoch sind einige Erläuterungen zu den Feldern “Bezeichner”, “Fähigkeiten” und “Gültigkeit” angebracht:

Das “Bezeichner-Feld” (*%f*) enthält eine der folgenden Markierungen:

R	Der Schlüssel wurde zurückgezogen und kann nicht verwendet werden
X	Der Schlüssel ist abgelaufen und kann nicht verwendet werden
d	Du hast den Schlüssel gesperrt
c	Es gibt unbekannte kritische selbst-signierte Blöcke

Das “Fähigkeiten-Feld” (*%c*) erzeugt eine Folge aus zwei Zeichen, die die Möglichkeiten eines Schlüssels darstellen. Das erste Zeichen steht für die Verschlüsselungs-Fähigkeit: Ein Minus-Zeichen (-) bedeutet, dass der Schlüssel nicht zum Kodieren verwendet werden kann. Ein Punkt (.) wird bei Schlüsseln gesetzt, die in einer

der Benutzer-IDs als “Signatur-Schlüssel” vorgesehen sind, aber mit denen dennoch auch Verschlüsselung möglich ist. Der Buchstabe **e** zeigt, daß ein Schlüssel zum Kodieren geeignet ist.

Das zweite Zeichen steht für die Signier-Fähigkeit des Schlüssels. Wieder bedeutet “-” “nicht zum signieren”, “.” steht für einen Schlüssel, der in einer der Benutzer-IDs als reiner “Kodierungs-Schlüssel” vorgesehen ist und “s” bezeichnet einen Schlüssel, der zum Signieren verwendet werden kann.

Das Gültigkeits-Feld (%t) zeigt schließlich, wie sicher eine Benutzer-ID zertifiziert ist. Ein Fragezeichen (?) steht für undefinierte Gültigkeit, ein Minus (-) für eine nicht vertrauenswürdige Zuordnung, das Leerzeichen für eine teilweise vertrauenswürdige Zuordnung und ein Plus (+) symbolisiert vollständige Gültigkeit.

2.4.3 Verschicken anonymer Nachrichten über “mixmaster”

Vielleicht hast du mutt so konfiguriert, daß es mit Mixmaster (einem anonymen Remailer) zusammenarbeitet. Mixmaster erlaubt es, über eine Kette von Remailern Mail anonym zu versenden. Die in mutt eingebaute Unterstützung von Mixmaster ist auf die Versionen 2.04 (beta 45 scheint die neueste zu sein) und 2.03 des Programms abgestimmt. Es unterstützt weder frühere Versionen, noch die sogenannten Version 3-Betas, von denen die neueste augenscheinlich 2.9b23 ist.

Um diese Funktionalität nutzen zu können, muss man bestimmte Einschränkungen einhalten. Die wichtigste davon ist, dass man weder den **Cc** noch den **Bcc** Header benutzen kann. Damit Mutt Mixmaster verwendet, muss man mit Hilfe der “Mix”-Funktion im *Nachricht erstellen*-Menü eine Remailer-Kette auswählen.

Der Bildschirm zur Auswahl einer Kette ist in zwei Bereiche unterteilt. Der (größere) obere Bereich enthält eine Liste von Remailern, die Du verwenden kannst. Im unteren Bereich ist die gerade ausgewählte Remailer-Kette dargestellt.

Du kannst innerhalb der Kette mit den **chain-prev** und **chain-next** Funktionen navigieren, die in der Voreinstellung auf die linke und rechte Pfeiltaste sowie auf die Tasten **h** und **l** gelegt sind (denke an die vi-Tastenbelegung). Um einen Remailer an der aktuellen Stelle in die Kette einzufügen, steht die **insert** Funktion zur Verfügung. Um einen Remailer hinter der aktuellen Position anzuhängen, verwendet man **select-entry** oder **append**. Mit der entsprechenden Funktion lassen sich auch Einträge aus der Kette entfernen. Schließlich entscheidet man sich, die Veränderungen zu verwerfen, indem man das Menü verlässt, oder mittels **accept** zu übernehmen, indem man (in der Voreinstellung) **Return** drückt.

Es ist wichtig zu wissen, dass unterschiedliche Remailer unterschiedliche Fähigkeiten haben, die in den Zeilen des Remailer-Menüs im %c Feld dargestellt werden (siehe dazu 6.3.106 (\$mix_entry_format)). Am wichtigsten ist die Eigenschaft “Mittelsmann”, die am Großbuchstaben “M” zu erkennen ist: Sie bedeutet, dass der fragliche Remailer nicht als letztes Element einer Kette verwendet werden kann, sondern nur Nachrichten an andere Mixmaster-Remailer weiterleitet. Bitte konsultiere die Dokumentation von Mixmaster, um Einzelheiten zu den anderen Fähigkeiten zu erfahren.

2.5 Nachrichten weiterleiten

“Bounce” und “Forward” ermöglichen es, eine vorhandene Nachricht an andere Empfänger weiterzuleiten. Beim “Bounce” wird der 6.3.198 (sendmail) Befehl verwendet, um die Nachricht unverändert an weitere Adressen zu schicken - so, als ob sie zu den Originalempfängern zählen würden. “Forward” hingegen erlaubt es, die Nachricht vor dem Weiterleiten noch zu verändern (z.B. durch Einfügen eines eigenen Kommentars).

Folgende Tastenbelegungen sind standardmäßig vorhanden:

f	forward	Leite Nachricht mit Kommentar weiter
b	bounce	Versende Nachricht erneut an anderen Empfänger

Das Weiterleiten ist entweder durch Einfügen der Originalnachricht in den Rumpf der neuen Nachricht möglich (wobei Mutt automatisch kenntlich macht, wo die weitergeleitete Mail beginnt und endet) oder durch einen MIME-Anhang. Was verwendet wird, hängt davon ab, wie die Variable 6.3.103 (`$mime_forward`) gesetzt ist. Die Umwandlung von Anhängen in Text (wie sie im Pager erfolgt) kann dabei über die Variablen 6.3.52 (`$forward_decode`) und 6.3.104 (`$mime_forward_decode`) gesteuert werden. Es kann vom Inhalt einer Mail abhängen, wie man sie weiterleiten will, deshalb handelt es sich bei `$mime_forward` um eine Quad-Option, die zum Beispiel auf “ask-no” gesetzt werden kann.

Ob und welche Kopfzeilen eingefügt werden, wird durch die Einstellung der Variablen 6.3.233 (`$weed`) festgelegt, wenn nicht 6.3.103 (`mime_forward`) verwendet wird.

Das Editieren einer weitergeleiteten Nachricht erfolgt genauso wie beim Senden einer Nachricht oder beim Antworten.

2.6 Nachrichten zurückstellen

Wahrscheinlich möchtest du manchmal das Absenden einer Nachricht verschieben, nachdem du sie zu schreiben begonnen hast. Wenn die `postpone-message` Funktion im *Nachricht erstellen* Menü aufgerufen wird, dann werden der Rumpf der Nachricht und alle Anhänge in einer Mailbox gespeichert, die durch die Variable 6.3.167 (`$postponed`) festgelegt wird. Das bedeutet, dass du die Nachricht auch dann wieder zurückholen kannst, wenn du Mutt zwischendurch verlassen hast.

Sobald eine Nachricht zurückgestellt wurde, gibt es mehrere Möglichkeiten, sie weiterzubearbeiten. Auf der Kommandozeile kannst du dazu die “-p” Option verwenden. Wenn du mittels `compose` aus dem *index* oder *pager* eine neue Nachricht erstellst, wirst du gefragt, ob du an der zurückgestellten Nachricht weiterschreiben willst. Wenn es mehrere zurückgestellte Nachrichten gibt, dann erscheint das *Zurückgestellte Nachrichten* Menü, über das die Nachricht ausgewählt werden kann, die zurückgeholt werden soll.

Anmerkung: Wenn du eine Antwort auf eine Nachricht (Reply) zurückstellst, dann wird die Nachricht erst dann als beantwortet gekennzeichnet, wenn die Antwort wirklich abgeschlossen und gesendet wurde. Es ist ebenfalls erforderlich, dass du dich in derselben Mailbox befindest, in der die beantwortete Nachricht abgelegt ist, damit ihr Status korrekt gesetzt werden kann.

Siehe auch die Quad-Option 6.3.166 (`$postpone`).

3 Konfiguration

Die Standardkonfiguration ermöglicht es, Mutt sofort (“out-of-the-box”) zu benutzen. Dennoch ist es oft wünschenswert, Mutt an die eigenen Bedürfnisse anzupassen. Beim ersten Aufruf von Mutt versucht das Programm, die systemweite Konfigurationsdatei (Einstellungen, die durch deinen lokalen Systemadministrator vorgenommen wurden) zu lesen, es sei denn, es wurde die 6.1 (Kommandozeilen-) Option “-n” angegeben. Diese Datei heißt üblicherweise `/usr/local/share/mutt/Muttrc` oder `/etc/Muttrc`. Als nächstes sucht Mutt in deinem Heimatverzeichnis nach einer Datei mit dem Namen `.muttrc`. Wenn diese Datei nicht existiert und es in deinem Heimatverzeichnis ein Unterverzeichnis mit dem Namen `.mutt` gibt, dann versucht Mutt eine Datei mit dem Namen `.mutt/muttrc` zu laden.

`.muttrc` ist die Datei, in der du normalerweise die 6.2 (Befehle) einträgst, um Mutt zu konfigurieren.

Zusätzlich unterstützt Mutt versionsabhängige Konfigurationsdateien, die anstelle der oben genannten Standarddateien ausgewertet werden. Wenn es zum Beispiel auf deinem System eine Datei `Muttrc-0.88` im Verzeichnis für die systemweiten Konfigurationsdateien gibt und du Mutt in der Version 0.88 aufrufst, dann wird Mutt diese Datei anstelle der Datei `Muttrc` verwenden. Dasselbe gilt für die benutzerspezifische Konfigurationsdatei: Wenn du eine Datei `.muttrc-0.88.6` in deinem Heimatverzeichnis hast und Mutt-Version

0.88.6 benutzt, wird diese Datei anstelle der Standarddatei `.muttrc` eingelesen. Die Versionsnummer ist genau die, die angezeigt wird, wenn Mutt mit der “-v” 6.1 (Kommandozeilen-) Option aufgerufen oder im Menü *index* die `show-version` Taste (Voreinstellung: V) benutzt wird.

3.1 Syntax der Konfigurationsdateien

Eine Konfigurationsdatei besteht aus einer Folge von 6.2 (Befehlen). Jede Zeile in der Datei darf einen oder mehrere Befehle enthalten. Wenn mehrere Befehle in einer Zeile angegeben werden, dann müssen sie durch ein Semikolon (;) voneinander getrennt werden.

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

Das “Hash” oder “Lattenzaun” (“#”) wird als Kommentarzeichen verwendet. Du kannst es benutzen, um Anmerkungen in deine Konfigurationsdatei zu schreiben. Mutt ignoriert alles, was in einer Zeile hinter diesem Kommentarzeichen steht. Ein Beispiel:

```
my_hdr X-Disclaimer: Warum hörst du mir zu? # Dies ist ein Kommentar
```

Mit einfachen (‘) und doppelten (”) Anführungszeichen können Zeichenketten eingegeben werden, die Leer- oder Sonderzeichen enthalten. Der Unterschied zwischen den Anführungszeichen ist der gleiche wie in vielen populären Shell-Programmen: Das einfache Anführungszeichen dient dazu eine Zeichenkette festzulegen, in der keine Shellvariablen ersetzt oder mit dem “Backslash” (\) maskierte Zeichen interpretiert werden [vergleiche nächster Abschnitt], während dies bei Zeichenketten in doppelten Anführungszeichen geschieht. Zum Beispiel werden “backtics” (‘) innerhalb doppelter Anführungszeichen ausgewertet, innerhalb von einfachen Anführungszeichen jedoch **nicht**.

\ maskiert das nächste Zeichen, genauso wie in Shells wie `bash` und `zsh`. Wenn du zum Beispiel doppelte Anführungszeichen “” innerhalb einer Zeichenkette verwenden möchtest, dann kannst du “\” benutzen, um zu verhindern, daß das folgende Zeichen als Ende des Strings interpretiert wird.

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

Mit “\” kann das Zeichen “\” in die Zeile eingefügt werden. “\n” und “\r” haben die in C üblichen Bedeutungen von Zeilenumbruch (linefeed) und Wagenrücklauf (carriage-return).

Ein \ am Ende einer Zeile dient dazu, Befehle auf mehrere Zeilen aufzuteilen. Dabei darf aber nicht in der Mitte von Befehlsnamen getrennt werden.

Desweiteren ist es möglich, die Ausgabe eines Unix-Befehls in die Konfigurationsdatei einzubinden. Dazu muss das entsprechende Kommando in “backquotes” (‘) eingeschlossen werden. Zum Beispiel:

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

Die Ausgabe des Unix-Befehls “`uname -a`” wird eingefügt, bevor die Zeile ausgewertet wird. Zu beachten ist, daß die Konfigurationsdatei zeilenorientiert ist. Daher wird auch nur die erste Zeile der Ausgabe eines Unix-Befehls übernommen.

Auf UNIX Umgebungsvariablen kann genauso wie aus einer Shell wie `sh` oder `bash` zugegriffen werden, indem man vor den Namen der Variablen ein “\$” setzt. Zum Beispiel:

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

In den folgenden Abschnitten werden die Befehle erklärt, die Mutt versteht. Eine vollständige Liste ist in der 6.2 (Befehlsreferenz) enthalten.

3.2 Definieren und Benutzen von Aliasnamen

Benutzung: `alias Schlüssel Adresse [, Adresse, ...]`

Es ist für gewöhnlich etwas lästig, sich an die komplette Mailadresse eines Gesprächspartners zu erinnern oder sie komplett einzutippen. Mutt erlaubt es "Aliasnamen" zu erzeugen, die einen kurzen Schlüssel für eine komplette Mailadresse definieren.

Anmerkung: Wenn du einen Aliasnamen für eine Gruppe definierst (indem du mehr als eine Adresse angibst), dann müssen die einzelnen Adressen durch ein Komma (",") voneinander getrennt werden.

Entfernen eines oder mehrerer Aliasnamen ("*" steht für alle):

`unalias [* | Schlüssel ...]`

```
alias muttdude me@cs.hmc.edu (Michael Elkins)
alias theguys manny, moe, jack
```

Im Gegensatz zu anderen Mailreadern ist es bei Mutt nicht notwendig, dass die Aliasnamen in einer besonderen Datei gespeichert werden. Der `alias` Befehl darf in irgend einer Konfigurationsdatei auftauchen, solange diese Datei nur 3.24 (eingelassen) wird. Daher kann es bei Bedarf auch mehrere Alias-Dateien geben, oder es können alle Aliasnamen in der `.muttrc` definiert sein.

Auf der anderen Seite kann die 2.3.4 (create-alias) Funktion nur eine einzige Datei benutzen, nämlich genau die, die durch die Variable 6.3.3 (`$alias_file`) (die standardmäßig den Wert `~/muttrc` hat) festgelegt wurde. Diese Datei ist ebenfalls nichts "besonderes" in dem Sinne, dass Mutt Aliasnamen an jede beliebige Datei anhängen kann. Damit sie wirksam werden, ist es aber erforderlich, auch diese Datei explizit 3.24 (einzulesen).

Zum Beispiel:

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

Die Benutzung der Aliasnamen erfolgt einfach dadurch, dass der Aliasname überall dort eingegeben werden kann, wo Mutt nach einer Adresse fragt, so zum Beispiel am *To:* oder *Cc:* Prompt. Wenn die Variable 6.3.40 (`$edit_headers`) gesetzt ist, kannst Du die Aliasnamen auch mit deinem Editor in die entsprechenden Kopfzeilen eintragen.

Zusätzlich kannst du an den verschiedenen Stellen, wo nach einer Adresse gefragt wird, die TAB-Taste benutzen, um einen teilweise eingegebenen Aliasnamen zu vervollständigen. Wenn es mehrere Möglichkeiten gibt, geht Mutt in ein Menü, in dem (wie in einem Adressbuch) eine Liste aller passenden Aliasnamen angezeigt wird. Wenn du die komplette Liste aller Aliasnamen sehen möchtest, drückst du die TAB-Taste einfach ohne vorher etwas einzugeben, etwa am leeren Prompt oder nach einem Komma für mehrere Adressen.

Im Alias Menü kannst du mit der *select-entry* Taste (Voreinstellung: Return) so viele Aliasnamen auswählen, wie du möchtest. Mit der *exit* Taste (Voreinstellung: q) kehrst du zur Adresseingabe zurück.

3.3 Ändern der Standard Tastenbelegung

Benutzung: `bind Menü Taste Funktion`

Dieser Befehl erlaubt es, die voreingestellte Tastenbelegung (welche Funktion auf einen Tastendruck hin aufgerufen wird) zu ändern.

Menü legt fest, in welches Menü die Tastenbelegung gehört. Die momentan gültigen Werte sind:

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

Taste ist die Taste (oder Tastenkombination), die du zuordnen möchtest. Um ein Kontrollzeichen anzugeben, wird die Sequenz `\Cx` verwendet, wobei *x* der Buchstabe für das Kontrollzeichen ist (“\Ca” steht zum Beispiel für Control-A (bzw. Strg-A auf deutschen Tastaturen, *A.d.Ü.*)). Beachte, daß die Groß-/Kleinschreibung sowohl bei *x* als auch bei `\C` ignoriert wird. Die Definitionen `\CA`, `\Ca`, `\cA` und `\ca` bedeuten daher alle das gleiche. Alternative ist es möglich, die Taste als 3-stellige Oktalzahl mit führendem “\” anzugeben (zum Beispiel ist `\177` äquivalent zu `\c?`).

Desweiteren darf *Taste* folgende Strings enthalten:

<code>\t</code>	Tab
<code><tab></code>	Tab
<code>\r</code>	Wagenrücklauf (<i>carriage return</i>)
<code>\n</code>	Zeilenvorschub (<i>newline</i>)
<code>\e</code>	Escape
<code><esc></code>	Escape
<code><up></code>	Cursor hoch
<code><down></code>	Cursor runter
<code><left></code>	Cursor links
<code><right></code>	Cursor rechts
<code><pageup></code>	Bild hoch
<code><pagedown></code>	Bild runter
<code><backspace></code>	Backspace
<code><delete></code>	Entf
<code><insert></code>	Einf
<code><enter></code>	Enter
<code><return></code>	Return (Eingabetaste)
<code><home></code>	Pos1
<code><end></code>	Ende
<code><space></code>	Leertaste
<code><f1></code>	Funktionstaste F1
<code><f10></code>	Funktionstaste F10

Taste muss nicht in Anführungszeichen eingeschlossen werden, außer wenn Leerzeichen (“ ”) enthalten sind.

Funktion legt fest, welche Aktion ausgeführt wird, wenn die *Taste* gedrückt wird. Eine komplette Liste aller Funktionen ist in der 6.4 (Referenz) aufgeführt. Die Sonderfunktion *noop* hebt die Definition einer Tastenbelegung auf.

3.4 Aliasnamen für Zeichensätze definieren

Benutzung: `charset-hook` *Alias Zeichensatz*

Benutzung: `iconv-hook` *Zeichensatz lokaler Zeichensatz*

Mit dem Befehl `charset-hook` kann man Alias-Namen für Zeichensätze definieren. Dies ist nützlich, um Nachrichten richtig anzuzeigen, in denen ein Zeichensatz deklariert wurde, den mutt nur unter anderem Namen kennt.

Der Befehl `iconv-hook` definiert einen system-spezifischen Namen für einen Zeichensatz. Das ist hilfreich, falls die Bibliothek zur Konvertierung von Zeichensätzen auf deinem System darauf besteht, Zeichensätze mit seltsamen, system-spezifischen Namen zu versehen.

3.5 Von einer Mailbox abhängig Variablen setzen

Benutzung: `folder-hook` [!]*Muster Befehl*

Oft ist es wünschenswert, Einstellungen in Abhängigkeit von der gerade benutzten Mailbox zu verändern. Der Befehl `folder-hook` bietet die Möglichkeit, einen beliebigen Konfigurationsbefehl auszuführen. *Muster* ist ein regulärer Ausdruck, der angibt, vor dem Öffnen welcher Mailboxen der *Befehl* ausgeführt werden soll. Wenn für eine Mailbox mehrere `folder-hook`'s zutreffen, dann werden sie in der Reihenfolge ausgeführt, in der sie in der `muttrc` stehen.

Anmerkung: Wenn du am Anfang des Musters das Ausrufezeichen (“!”) als Abkürzung für die 6.3.213 (\$Spooldatei) verwendest, dann muss es in einfachen oder doppelten Anführungsstrichen stehen, damit es vom logischen *Nicht* Operator des Musters unterschieden werden kann.

Beachte, dass die alten Einstellungen nicht wieder hergestellt werden, wenn die Mailbox verlassen wird.

Als Beispiel ist ein möglicher Befehl, die Art der Sortierung der Nachrichten abhängig von der gelesenen Mailbox zu ändern:

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

Die Art der Sortierung wird hierbei nicht auf den alten Wert zurückgesetzt, wenn eine andere Mailbox geladen wird. Mit dem Muster “.” kann ein *Standard* Befehl festgelegt werden.

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

3.6 Tastatur Makros

Benutzung: `macro` *Menü Taste Sequenz* [*Beschreibung*]

Makros sind nützlich, wenn du möchtest, dass auf einen Tastendruck hin eine ganze Folge von Aktionen ausgeführt wird. Wenn du im Menü *Menü* die Taste *Taste* drückst, verhält sich Mutt so, als wenn du *Sequenz* eingegeben hättest. Wenn du also eine Folge von Befehlen hast, die du häufig hintereinander eingibst, dann kannst du ein Makro erzeugen, um diese Befehlsfolge mit einem Tastendruck auszuführen.

Prinzipiell sind *Taste* und *Sequenz* nach denselben Regeln aufgebaut wie die 3.3 (Tastenbelegungen). Allerdings gibt es ein paar Besonderheiten: Die erste ist, dass Kontrollzeichen innerhalb einer *Sequenz* auch als \hat{x} geschrieben werden können. Um das Caret (“^”) selbst zu erzeugen, muss daher $\hat{\hat{}}$ eingegeben werden. Zweitens kann man die Schreibweisen \langle *Tastename* \rangle und \langle *Funktionsname* \rangle verwenden, um spezielle Tasten wie *up* (Cursor rauf) anzugeben oder Funktionen direkt aufzurufen. Eine Auflistung der Tastennamen ist im Abschnitt zu den 3.3 (Tastenbindungen) enthalten. Die Funktionen sind in der 6.4 (Funktions-Referenz) aufgeführt.

Der Vorteil bei direkter Verwendung der Funktionsnamen ist, dass die Makros nicht die gegenwärtigen Tastenbindungen verwenden und ihre Funktion auf diese Weise nicht davon abhängt, dass der Benutzer bestimmte Tasten-Definitionen hat. Dadurch werden sie robuster und portabler; es erleichtert auch die Definition von Makros in Dateien, die von mehr als einer Person genutzt werden (z.B. in der globalen Muttrc).

Optional kannst du nach der *Sequenz* eine Beschreibung angeben, die dann in den Hilfebildschirmen angezeigt wird.

Anmerkung: Makrodefinitionen (so vorhanden), die in den Hilfebildschirmen angezeigt werden, werden bei der Anzeige auf Bildschirmbreite abgeschnitten und nicht umgebrochen.

3.7 Verwendung von Farb- und Mono Video Attributen

Benutzung: `color Objekt Vordergrund Hintergrund [Muster]`

Benutzung: `color index Vordergrund Hintergrund Muster`

Benutzung: `uncolor index Vordergrund Hintergrund [Muster]`

Wenn dein Terminal farbfähig ist, kannst du Mutt durch ein selbst definiertes Farbschema verschönern. Um die Farbe für ein Objekt (eine bestimmte Art von Information) festzulegen, musst du eine Vordergrundfarbe **und** eine Hintergrundfarbe festlegen (es ist nicht möglich nur eins von beiden anzugeben).

Objekt darf eines der folgenden sein:

- attachment
- body (Übereinstimmung mit *Muster* im Rumpf der Nachrichten)
- bold (Hervorheben von Fettdruck im Rumpf der Nachrichten)
- error (Fehlermeldungen, die Mutt ausgibt)
- header (Übereinstimmung mit *Muster* in den Kopfzeilen der Nachrichten)
- hdrdefault (Standardfarbe für die Kopfzeilen im Pager)
- index (Übereinstimmung mit *Muster* im Index)
- indicator (Pfeil oder Balken zur Markierung des aktuellen Eintrags in einem Menü)
- markers (die “+” Markierungen am Anfang von umgebrochenen Zeilen im Pager)
- message (Informative Hinweistexte)
- normal
- quoted (Übereinstimmung mit 6.3.176 (`$quote_regexp`) im Rumpf einer Nachricht)
- quoted1, quoted2, ..., quotedN (Höhere Zitatebenen)
- search (Markierung von Worten im Pager)
- signature
- status (Modi zur Anzeige von Informationen über die Mailbox oder Nachricht)
- tilde (die “~”, die Leerzeilen im Pager auffüllt)
- tree (Thread-Baum, der im Nachrichtenindex und im Attachment Menü angezeigt wird)
- underline (Hervorheben von Unterstreichungen im Rumpf der Nachrichten)

Vordergrund und *Hintergrund* kann eine der folgenden Farben sein:

- white (weiß)
- black (schwarz)
- green (grün)
- magenta
- blue (blau)
- cyan
- yellow (gelb)
- red (rot)
- default
- *colorx*

Beim *Vordergrund* kann der jeweiligen Farbe das Schlüsselwort `bright` vorangestellt werden, um die Vordergrundfarbe fett gedruckt zu bekommen (zum Beispiel `brightred`).

Wenn es dein Terminal unterstützt, dann kann das besondere Schlüsselwort `default` als transparente Farbe benutzt werden. Der Wert `brightdefault` ist ebenfalls gültig. Wenn Mutterm gegen die *S-Lang* Bibliothek gebunden wurde, dann musst du noch die Umgebungsvariable `COLORFGBG` auf die Standardfarben deines Terminals setzen, damit das funktioniert; zum Beispiel (für Bourne-artige Shells):

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

Anmerkung: Die *S-Lang* Bibliothek verlangt, dass anstelle von `white` und `yellow` die Schlüsselwörter `lightgray` und `brown` benutzt werden, wenn diese Variable gesetzt wird.

Anmerkung: Der `uncolor` Befehl kann nur auf das Objekt `index` angewendet werden. Es entfernt Einträge aus der Liste. Du **musst** dasselbe Muster wie beim entsprechenden `color` Befehl angeben, damit der Eintrag entfernt wird. Das Muster `"*"` ist ein besonderes Zeichen mit der Bedeutung, dass die komplette Liste gelöscht werden soll.

Mutt erkennt auch die Schlüsselwörter `color0`, `color1`, ..., `colorN-1` (wobei **N** die Anzahl der von deinem Terminal unterstützten Farben ist). Das ist nützlich, wenn du die Farben für deine Anzeige undefinierst (zum Beispiel, indem du im `xterm` die Farbe änderst, die mit `color2` verknüpft ist), weil dann die Farbnamen ihre normale Bedeutung verlieren.

Wenn dein Terminal keine Farben unterstützt, ist es mit dem `"mono"` Befehl dennoch möglich, Videoattribute zu verändern.

Benutzung: `mono <Objekt> <Attribut> [regulärer Ausdruck]`

Benutzung: `mono index Attribut Muster`

Benutzung: `unmono index Muster [Muster ...]`

wobei *Attribut* eines der folgenden ist:

- none (keins)
- bold (fett)

- underline (unterstrichen)
- reverse (invers)
- standout

3.8 Ignorieren (weeding) von nicht gewünschten Kopfzeilen

Benutzung: `[un]ignore Muster [Muster ...]`

Nachrichten haben oft viele Kopfzeilen, die automatisch eingefügt werden oder solche, bei denen es nicht sinnvoll erscheint, sie am Bildschirm anzuzeigen. Dieser Befehl erlaubt es, Kopfzeilen festzulegen, die du normalerweise nicht sehen willst.

Es ist dabei nicht nötig, den kompletten Namen des Headers anzugeben. Zum Beispiel ignoriert der Eintrag “ignore content-” alle Kopfzeilen, die mit “content-” anfangen.

Um ein zuvor eingetragenes Muster aus der Liste auszutragen, wird der “unignore” Befehl benutzt. “unignore *” entfernt alle Muster aus der ignore-Liste.

Zum Beispiel:

```
# Sven's drakonische Auswahl von Kopfzeilen
ignore *
unignore from date subject to cc
unignore organization organisation x-mailer: x-newsreader: x-mailing-list:
unignore posted-to:
```

3.9 Mailinglisten

Benutzung: `[un]lists Adresse [Adresse ...]`

Benutzung: `[un]subscribe Adresse [Adresse ...]`

Mutt verfügt über einige schöne Funktionen zur 4.8 (Behandlung von Mailinglisten). Um von diesen Funktionen zu profitieren, musst du angeben, welche Adressen die Mailinglisten haben und welche davon du gerade abonniert hast. Sobald du das getan hast, wird der Befehl 2.3.4 (list-reply) für alle bekannten Listen funktionieren.

Zusätzlich wird mutt in Nachrichten an abonnierte Listen einen Mail-Followup-To Header setzen, um die Mailprogramme anderer darauf hinzuweisen, dass du keine Kopien ihrer Antworten an deine Privatadresse wünschst. Beachte bitte, dass der Mail-Followup-To Header eine noch nicht standardisierte Erweiterung ist, die nicht von allen Programmen unterstützt wird. Ihn einer Nachricht hinzuzufügen schließt nicht mit völliger Sicherheit aus, dass du trotzdem persönliche Kopien von Listen-Nachrichten bekommst. Beachte auch, dass die Erzeugung des Mail-Followup-To Headers von der Variablen 6.3.50 (followup_to) gesteuert wird.

Um genauer zu sein, verwaltet mutt Listen der ihm bekannten und der abonnierten Mailinglisten. Jede abonnierte Mailingliste zählt dabei automatisch als “bekannt”. Um eine Mailingliste als bekannt zu markieren, steht der Befehl “lists” zur Verfügung. Um sie als abonniert zu kennzeichnen, verwendet man “subscribe”.

Du solltest nur so viel von der Adresse angeben, wie nötig ist, um Mehrdeutigkeiten zu vermeiden. Wenn du zum Beispiel die Mailing-Liste von mutt abonniert hast, erhältst du an `mutt-users@mutt.org` adressierte Mail. Um mutt mitzuteilen, dass es sich dabei um eine Mailingliste handelt, kannst Du “lists mutt-users” in deine `.muttrc` schreiben; um mutt beizubringen, dass du diese Liste abonniert hast, würdest du statt dessen den Befehl “subscribe mutt-users” verwenden. Wenn du zufälligerweise auch Mail von jemandem mit der Adresse `mutt-users@example.com` bekommst, würdest du “lists mutt-users@mutt.org” bzw. “subscribe mutt-users@mutt.org” verwenden, um nur Mail zu erfassen, die tatsächlich von der Liste stammt.

Der Befehl “unlists” wird verwendet, um einen Eintrag aus den Listen der bekannten und abonnierten Mailing-Listen zu entfernen. Mit “unlists *” lassen sich die Listen auch vollständig leeren.

Um einen Eintrag zwar aus der Liste der abonnierten Mailinglisten zu entfernen, ihn aber in der Liste der bekannten Mailinglisten zu belassen, existiert der Befehl “unsubscribe”.

3.10 Mehrere Spool-Mailboxen verwenden

Benutzung: `mbox-hook` [!] *Muster Mailbox*

Dieser Befehl wird verwendet, um gelesene Nachrichten automatisch aus den sogenannten “Spool-Mailboxen” (die nicht zur Archivierung von Mail bestimmt sind) in eine andere zu verschieben, wenn du mutt verlässt oder in eine andere Mailbox wechselst.

Muster ist ein regulärer Ausdruck, der angibt, welche Mailbox als Spool-Mailbox angesehen werden soll. *Mailbox* gibt an, in welcher Mailbox die Nachrichten daraus gespeichert werden sollen, wenn sie gelesen wurden.

Anders als bei anderen *hook* Befehlen wird hier nur der *erste* Eintrag benutzt, auf den das Muster zutrifft (es ist nicht möglich, gelesene Mail in mehr als einer Mailbox zu speichern).

3.11 Mailboxen angeben, die Nachrichten empfangen

Benutzung: `mailboxes` [!] *Dateiname* [*Dateiname ...*]

Mit diesem Befehl werden die Mailboxen definiert, in denen das System empfangene Nachrichten ablegt und die deshalb auf neue Nachrichten überprüft werden sollen. Standardmäßig wird in der Hauptstatuszeile angezeigt, wie viele dieser Mailboxen neue Nachrichten enthalten.

Beim Wechsel der Mailbox kann man mit der *Leertaste* durch die Mailboxen blättern, die neue Nachrichten enthalten.

Wenn man in einer Datei-Liste die Tabulator-Taste drückt, erscheint ein Menü, in dem die Dateien aufgeführt sind, die mit dem `mailboxes` Befehl angegeben wurden und in dem angezeigt wird, welche davon neue Nachrichten enthalten. Mutt startet automatisch in diesem Modus, wenn es von der Kommandozeile aus mit der Option `-y` aufgerufen wurde.

Anmerkung: Neue Mail wird erkannt, indem für die Mailbox-Datei das letzte Zugriffsdatum mit dem Datum der letzten Änderung verglichen wird. Hilfsprogramme wie `biff`, `frm` und andere, die auf die Mailbox zugreifen, können (wenn sie die Zugriffszeit nicht zurücksetzen) daher dafür verantwortlich sein, dass Mutt nie neue Mail meldet. Eine weitere Ursache für geänderte Zugriffszeiten sind Backup-Programme.

Anmerkung: Die Dateinamen im Befehl `mailboxes` werden bereits aufgelöst, wenn der Befehl ausgeführt wird. Wenn diese Namen also 4.7 (Kürzel) (wie etwa “=” and “!”) enthalten, dann sollte jede Variablendefinition, die auf die Bedeutung dieser Zeichen Einfluss hat (etwa 6.3.48 (\$folder) und 6.3.213 (\$spoolfile)) vor dem Befehl `mailboxes` stehen.

3.12 Benutzerdefinierte Kopfzeilen

Benutzung:

`my_hdr` *Zeichenkette*

`unmy_hdr` *Feld* [*Feld ...*]

Der Befehl “my_hdr” erlaubt es eigene Kopfzeilen zu definieren, die in jede Nachricht, die man verschickt, eingefügt werden.

Wenn du zum Beispiel eine “Organization:” Zeile in jede ausgehende Nachricht einfügen möchtest, dann kannst du dafür den Befehl

```
my_hdr Organization: Eine Wirklich Große Firma, Irgendwo, Deutschland
```

in deine `.muttrc` aufnehmen.

Anmerkung: Leerzeichen zwischen dem Schlüsselwort und dem Doppelpunkt (“:”) sind *nicht* erlaubt. Der Standard für E-Mail (RFC822) untersagt Leerzeichen an dieser Stelle, daher erzwingt Mutt die Einhaltung dieser Regel.

Wenn du eine Kopfzeile nur in eine einzelne Nachricht einfügen möchtest, dann kannst du entweder die Variable 6.3.40 (`edit_headers`) setzen, oder du verwendest die Funktion `edit-headers` (in der Voreinstellung: “E”) im *Nachricht erstellen* Menü, um die Kopfzeilen zusammen mit der Nachricht editieren zu können.

Zum Entfernen von benutzerdefinierten Kopfzeilen dient der Befehl “`unmy_hdr`”. Du kannst dabei entweder einen Stern (“*”) angeben, um alle Kopfzeilen zu entfernen, oder den Namen der zu löschenden Zeile. Um zum Beispiel alle “To” und “Cc” Kopfzeilen zu entfernen, kannst Du folgendes angeben:

```
unmy_hdr to cc
```

3.13 Sortierung der Kopfzeilen in der Nachrichtenanzeige festlegen

Benutzung: `hdr_order Kopfzeile1 Kopfzeile2 Kopfzeile3`

Mit diesem Befehl wird die Reihenfolge festgelegt, in der die Kopfzeilen ausgegeben werden sollen, wenn eine Nachricht angezeigt wird.

“`unhdr_order *`” entfernt alle zuvor definierten Kopfzeilen aus der Sortierreihenfolge und schaltet somit die in der systemweiten Konfigurationsdatei definierte Sortierung ab.

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

3.14 Standard-Datei zum Speichern festlegen

Benutzung: `save-hook [!]Muster Dateiname`

Mit diesem Befehl wird die Standard-Datei geändert, in der Nachrichten gespeichert werden. Die Datei *Dateiname* wird als Standard-Datei benutzt, wenn im *From:* Header der Nachricht eine Adresse steht, die auf das *Muster* passt, oder wenn du die Nachricht geschrieben hast und die Adresse im *To:* auf das *Muster* passt.

Im Kapitel 4.4.1 (Auswählen von Nachrichten) können Informationen über das genaue Format von *Muster* nachgelesen werden.

Beispiele:

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

Siehe auch den Befehl 3.16 (`fcc-save-hook`).

3.15 Standard Fcc: Mailbox beim Verfassen festlegen

Benutzung: `fcc-hook [!]Muster Mailbox`

Mit diesem Befehl können Kopien gesendeter Nachrichten in einer anderen Mailbox als 6.3.181 (`$record`) gespeichert werden. Mutt sucht in der Liste der Empfänger nach der ersten Übereinstimmung mit *Muster* und benutzt dann die *Mailbox* als Voreinstellung für die Fcc: Mailbox. Wenn keine Übereinstimmung gefunden wurde, dann wird die Nachricht in der Mailbox gespeichert, die in 6.3.181 (`$record`) angegeben ist.

Im Kapitel 4.4.1 (Auswählen von Nachrichten) können Informationen über das genaue Format von *Muster* nachgelesen werden.

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

Das obige speichert eine Kopie von allen Nachrichten, die an die Domain aol.com gesendet werden, in der Mailbox '+spammers'. Siehe auch den Befehl 3.16 (`fcc-save-hook`).

3.16 Standard-Dateiname zum Speichern und Fcc: Mailbox zusammen festlegen

Benutzung: `fcc-save-hook [!]Muster Mailbox`

Dieser Befehl ist lediglich eine Abkürzung, er ist äquivalent zur gleichzeitigen Ausführung der beiden Befehle 3.15 (`fcc-hook`) und 3.14 (`save-hook`) mit den angegebenen Argumenten.

3.17 Einstellungen abhängig vom Empfänger ändern

Benutzung: `send-hook [!]Muster Befehl`

Mit diesem Befehl können beliebige Konfigurationsbefehle in Abhängigkeit vom Empfänger der Nachricht ausgeführt werden. *Muster* ist ein regulärer Ausdruck für die gewünschte Empfängeradresse. *Befehl* wird ausgeführt, wenn *Muster* auf einen der Empfänger der Nachricht zutrifft. Wenn es mehrere Übereinstimmungen gibt, dann werden sie in der Reihenfolge ausgeführt, in der sie in der `.muttrc` stehen.

Im Kapitel 4.4.1 (Auswählen von Nachrichten) kannst du Informationen über das genaue Format von *Muster* nachlesen.

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

Eine andere typische Anwendung für diesen Befehl ist das Ändern der Werte der 6.3.15 (`$attribution`), 6.3.203 (`$signature`) und 6.3.86 (`$locale`) Variablen, um die Sprache der Anrede und Signatur abhängig vom Empfänger zu setzen.

Anmerkung: Die `send-hooks` werden nur EINMAL ausgeführt, nachdem die ursprüngliche Liste der Empfänger eingelesen wurde. Wenn man einen Empfänger hinzufügt, während die Nachricht editiert wird, dann führt das NICHT dazu, dass irgendein `send-hook` ausgeführt wird. Ebenso haben `my_hdr` Befehle, die Empfänger- oder Betreffzeilen in den Headern ändern, keinen Einfluss auf die vorliegende Nachricht, wenn sie in einem `send-hook` aufgerufen werden.

3.18 Verändere Einstellungen, bevor eine Nachricht formatiert wird

Benutzung: `message-hook [!]Muster Befehl`

Dieser Befehl kann verwendet werden, um in Abhängigkeit verschiedener Informationen über eine Nachricht beliebige Konfigurations-Befehle auszuführen, bevor sie angezeigt oder formatiert wird. Der *Befehl* wird

ausgeführt, wenn das *Muster* auf die anzuzeigende Nachricht passt. Bei mehreren Übereinstimmungen werden die Befehle in der Reihenfolge ausgeführt, in der sie in der *muttrc* angegeben wurden.

Informationen zum genauen Format des *Musters* sind im Abschnitt 4.4.1 (Suchmuster für Nachrichten in Hooks) enthalten.

Beispiel:

```
message-hook ^A 'set pager=builtin'
message-hook '~f freshmeat-news' 'set pager="less \"/>

```

3.19 Auswahl des PGP Schlüssels für einen Empfänger

Benutzung: `pgp-hook Muster keyid`

Wenn eine Nachricht mit PGP verschlüsselt werden soll, dann kann es wünschenswert sein, einen bestimmten PGP Schlüssel mit einem Empfänger zu verknüpfen. Das ist zum Beispiel sinnvoll, wenn der Schlüssel nicht direkt aus der Empfängeradresse ermittelt werden kann oder wenn man einen anderen Schlüssel verwenden möchte als den, den *mutt* normalerweise auswählen würde. Der `pgp-hook` Befehl bietet die Möglichkeit, die Benutzer-ID des öffentlichen Schlüssels anzugeben, der für die Verschlüsselung von Mails an einen bestimmten Empfänger benutzt werden soll.

3.20 Befehlssequenzen in den Tastaturpuffer schreiben

Benutzung: `push Befehlssequenz`

Dieser Befehl fügt die angegebene Befehlssequenz in den Tastaturpuffer ein. Sie kann genau wie die Sequenz im Befehl 3.6 (macro) Kontrollzeichen enthalten, Tastennamen und Funktionsnamen. Der Befehl kann benutzt werden, um beim Start von *Mutt* oder beim Betreten einer neuen Mailbox eine bestimmte Folge von Befehlen auszuführen.

3.21 Ausführen von Funktionen

Benutzung: `exec Funktion [Funktion ...]`

Dieser Befehl kann verwendet werden, um eine beliebige Funktion auszuführen. Die Funktionen sind in der 6.4 (Funktions-Referenz) aufgeführt. “`exec Funktion`” ist gleichbedeutend mit “`push <Funktion>`”.

3.22 Nachrichten bewerten (Scoring)

Benutzung: `score Muster Wert`

Benutzung: `unscore Muster [Muster ...]`

Wenn das *Muster* auf eine Nachricht passt, addiert der Befehl `score` den angegebenen *Wert* zu ihrer Punktzahl (Score). *Muster* ist eine Zeichenkette, deren Format im Abschnitt 4.2 (Suchmuster) beschrieben wird (Hinweis: Aus Gründen der Effizienz können Muster, die Informationen benötigen, die im Index nicht verfügbar sind - etwa `~b`, `~B` oder `~h` - nicht verwendet werden). *Wert* ist eine positive oder negative ganze Zahl. Die endgültige Punktzahl einer Nachricht ist die Gesamtsumme aus allen passenden `score` Einträgen. Es ist allerdings möglich, dem *Wert* ein Gleichheitszeichen (“=”) voranzustellen, um die Berechnung des Score an dieser Stelle zu stoppen, wenn eine Übereinstimmung vorliegt. Ein negatives Endergebnis wird auf 0 aufgerundet.

Der Befehl `unset` entfernt Score-Einträge aus der Liste. Du **musst** dasselbe Muster wie beim entsprechenden `score` Befehl angeben, damit der Eintrag entfernt wird. Das Muster “*” ist ein besonderes Zeichen mit der Bedeutung, dass die komplette Liste der Score-Einträge gelöscht werden soll.

3.23 Der Umgang mit Konfigurations-Variablen

Benutzung: `set [no|inv] Variable[= Wert] [Variable ...]`

Benutzung: `toggle Variable [Variable ...]`

Benutzung: `unset Variable [Variable ...]`

Benutzung: `reset Variable [Variable ...]`

Dieses Kommando dient dem Setzen (`set`) und Entfernen (`unset`) von 6.3 (Variablen). Es gibt vier grundlegende Typen von Variablen: `boolean` (Wahrheitswert), `number` (positive Ganzzahl), `string` (Zeichenkette) und `Quad-Option`. `boolean`-Variablen können mit `set` auf wahr (`true`) und mit `unset` auf unwahr (`false`) geschaltet werden. `number`-Variablen können positive ganzzahlige Werte aufnehmen.

`string`-Variablen bestehen aus einer beliebigen Anzahl von druckbaren Zeichen. Sie müssen von Anführungszeichen umschlossen werden, wenn Leerzeichen oder Tabulatoren enthalten sind. Die Escape-Sequenzen `\n` für den Zeilenvorschub und `\t` für den Tabulator können wie in der Programmiersprache “C” benutzt werden.

`quadoption`-Variablen legen fest, ob Mutt bei einer bestimmten Aktion den Benutzer fragen oder eine Standard-Aktion ausführen soll. Der Wert `yes` sorgt für eine sofortige Ausführung der Aktion, genau so, als wenn der Benutzer auf die vom Programm gestellte Frage mit “ja” geantwortet hätte. `no` funktioniert ähnlich, nur dass die Aktion so ausgeführt wird, als wenn der Benutzer mit “nein” geantwortet hätte. Der Wert `ask-yes` führt zu einer Benutzerabfrage, wobei die voreingestellte Antwort “ja” ist, die nur noch mit Return bestätigt werden muss. Bei `ask-no` ist die voreingestellte Antwort “nein”.

Eine Variable mit vorangestelltem “no” schaltet die betreffende Option ab. Beispiel: `set noaskbcc`.

`boolean`-Variablen kann `inv` vorangestellt werden, um den Wert auf den jeweils anderen Zustand umzuschalten. Das ist besonders beim Schreiben von Makros sinnvoll. Beispiel: `set invsmart-wrap`.

Das `toggle`-Kommando setzt automatisch `inv` vor die angegebenen Variablennamen.

Das `unset`-Kommando setzt automatisch `no` vor die angegebenen Variablennamen.

Mit der `enter-command`-Funktion aus dem Menü `index` kann der Wert einer Variable abgefragt werden. Dazu muss dem Variablennamen ein Fragezeichen vorangestellt werden:

```
set ?allow_8bit
```

Benötigt wird das Fragezeichen nur bei Variablen der Typen `boolean` und `Quad-Option`.

Das `reset`-Kommando setzt alle angegebenen Variablen auf die einkompilierten Standardwerte, die in diesem Handbuch aufgeführt sind. Das `set`-Kommando und ein “&” vor dem Variablennamen führen zum selben Ergebnis.

Für `reset` gibt es die spezielle Variable “all”, die sämtliche Variablen auf Standardwerte zurücksetzt.

3.24 Lesen von Konfigurations-Kommandos aus einer anderen Datei

Benutzung: `source dateiname`

Mit diesem Kommando können Konfigurations-Kommandos aus anderen Dateien eingebunden werden. Ich speichere beispielsweise meine Alias-Namen (eine Art Adressbuch) in der Datei `~/mail_aliases`. Auf diese Weise kann ich `~/muttrc` für alle lesbar machen und gleichzeitig meine Alias-Namen privat halten.

Wenn der Dateiname mit einer Tilde (“~”) beginnt, wird er durch den Pfad des Home-Verzeichnisses ersetzt.

Wenn der Dateiname mit einem Balken (“|”) endet, interpretiert mutt diesen Dateinamen als ausführbares Programm und liest dessen Ausgabe ein. Beispiel: `source ~/bin/myscript|`

3.25 Entfernen von Hooks

Benutzung: `unhook [* | Hook-Art]`

Dieser Befehl ermöglicht es dir, Hooks wieder zu entfernen, die du vorher definiert hattest. Du kannst entweder alle Hooks entfernen, indem du das Zeichen “*” als Argument angibst, oder du kannst alle Hooks eines speziellen Typs entfernen, indem du etwas wie `unhook send-hook` verwendest.

4 Fortgeschrittene Anwendung

4.1 Reguläre Ausdrücke

Alle Such-Muster in Mutt, auch komplexere 4.2 (Muster), müssen als reguläre Ausdrücke (regex) in der “erweiterten POSIX”-Syntax (was mehr oder weniger die von egrep und GNU awk verwendete Syntax ist) angegeben werden. Eine kurze Beschreibung dieser Syntax folgt weiter unten.

Die Suche unterscheidet nur dann zwischen Groß- und Kleinschreibung, wenn der zu suchende Ausdruck mindestens einen Großbuchstaben enthält. Sonst findet keine Unterscheidung statt. Beachte, dass “\” in regulären Ausdrücken maskiert werden muss, wenn es in einem Konfigurations-Kommando verwendet wird: “\\”.

Ein regulärer Ausdruck ist ein Muster, das eine Anzahl von Zeichenketten beschreibt. Reguläre Ausdrücke werden - analog zu arithmetischen Ausdrücken - durch die Verwendung von verschiedenen Operatoren aufgebaut, die kleinere Ausdrücke verbinden.

Beachte, dass ein regulärer Ausdruck entweder von “” oder ‘ ’ umgeben / begrenzt werden kann, was nützlich ist, falls der Ausdruck Leerzeichen enthält. Der Abschnitt 3.1 (Syntax der Konfigurationsdateien) enthält mehr Information zum Verständnis der Begrenzer “” und ‘ ’. Um ein wörtliches “” oder ‘ ’ einzufügen, musst du es mit einem \ (Backslash) maskieren.

Die kleinstmöglichen Einheiten sind reguläre Ausdrücke, die nur ein einzelnes Zeichen beschreiben. Die meisten Zeichen - alle Buchstaben und Ziffern eingeschlossen - sind reguläre Ausdrücke, die sich selbst beschreiben. Jedes Metazeichen (Zeichen mit einer besonderen Bedeutung) kann maskiert werden, indem man ihm einen Backslash voranstellt.

Der Punkt “.” passt auf jedes beliebige einzelne Zeichen. “^” und das Dollarzeichen “\$” sind Metazeichen, die die leere Zeichenkette am Anfang bzw. Ende einer Zeile beschreiben.

Eine Liste von Zeichen, die zwischen “[” und “]” steht, ist eine Beschreibung für ein einzelnes (beliebiges) Zeichen in der Liste. Wenn das erste Zeichen dieser Liste “^” ist, dann beschreibt die Liste genau die Zeichen, die *nicht* in der Liste vorkommen. Beispielsweise steht der reguläre Ausdruck `[0123456789]` für eine beliebige Ziffer. Ein Bereich von ASCII-Zeichen kann durch Angabe des ersten und letzten Zeichens, getrennt durch einen Bindestrich (“-”) beschrieben werden. Die meisten Metazeichen verlieren ihre spezielle Bedeutung, wenn sie innerhalb von Listen angewandt werden. Um das Zeichen “[” verwenden zu können, muss es jedoch als erstes in der Liste stehen. Mit dem “^” verhält es sich ähnlich, nur dass es überall außer an erster Stelle stehen darf. Schließlich kann auch der Bindestrich ohne seine spezielle Bedeutung verwendet werden. Dazu muss er als letztes Zeichen in der Liste stehen.

Bestimmte Zeichenklassen sind vordefiniert. Zeichenklassen bestehen aus “[:”, einem Schlüsselwort (dem Namen der Klasse) und “:]”. Folgende Klassen sind im POSIX-Standard definiert:

[`:alnum:`]

Alphanumerische Zeichen.

[`:alpha:`]

Zeichen des Alphabets.

[`:blank:`]

Leerzeichen oder Tabulatoren.

[`:cntrl:`]

Steuerzeichen.

[`:digit:`]

Numerische Zeichen.

[`:graph:`]

Zeichen, die sowohl druck- als auch sichtbar sind. (Ein Leerzeichen ist druckbar, aber nicht sichtbar, während ein "a" sowohl druck- als auch sichtbar ist.)

[`:lower:`]

Kleinbuchstaben.

[`:print:`]

Druckbare Zeichen. (Zeichen, die keine Steuerzeichen sind.)

[`:punct:`]

Satzzeichen (Zeichen, die weder Buchstaben, Ziffern, Steuerzeichen noch Leerzeichen sind.)

[`:space:`]

Zwischenraum (z. B. Leerzeichen, Tabulator, Wagenrücklauf)

[`:upper:`]

Großbuchstaben.

[`:xdigit:`]

Hexadezimale Ziffern

Eine Zeichenklasse ist in einem regulären Ausdruck nur innerhalb der eckigen Klammern ("`[`", "`]`") einer Zeichenliste gültig. Beachte, dass die Klammern in den Klassennamen Teil des symbolischen Namens sind und zusätzlich zu den Klammern der Zeichenliste gesetzt werden müssen. [`:digit:`] ist beispielsweise äquivalent zu [`0-9`].

In diesen Zeichenlisten können überdies zwei besondere Zeichenfolgen erscheinen. Diese finden in Nicht-ASCII-Zeichensätzen Anwendung, die einzelne Symbole besitzen, die durch mehrere Zeichen repräsentiert werden ("collating elements", Gruppierungselemente), sowie für Zeichen, die von Such- und Sortiermethoden als gleich behandelt werden sollen:

Gruppierungssymbole - "Collating Symbols"

Ein Gruppierungssymbol ("collating symbol") besteht aus einem Gruppierungselement ("collating element"), das von "`[.`" und "`.]`" umschlossen wird. Wenn beispielsweise "`ch`" ein Gruppierungselement ist, dann ist [`.ch.`] ein regulärer Ausdruck, der dieses Element beschreibt, während der reguläre Ausdruck [`ch`] "`c`" oder "`h`" beschreibt.

Äquivalenz-Klassen

Eine Äquivalenz-Klasse ist ein von der Einstellung der locale-Variablen abhängiger Name, der für eine Reihe von Zeichen steht, die als gleich behandelt werden sollen. Der Name wird in “[=” und “=]” eingeschlossen. Beispielsweise könnte der Name “e” für “è” “é” und “e” stehen. In diesem Fall ist “[=e=]” ein regulärer Ausdruck, der “è”, “é” und “e” beschreibt.

Einem regulären Ausdruck, der ein einzelnes Zeichen beschreibt, können folgende Wiederholungs-Operatoren folgen:

?

Das vorangehende Element ist optional und darf höchstens einmal auftreten.

*

Das vorangehende Element kommt kein- oder mehrmals vor.

+

Das vorangehende Element tritt ein- oder mehrmals auf.

{n}

Das vorangehende Element kommt genau n mal vor.

{n,}

Das vorangehende Element kommt n mal oder öfter vor.

{,m}

Das vorangehende Element kommt höchstens m mal vor.

{n,m}

Das vorangehende Element kommt mindestens n mal, aber nicht öfter als m mal vor.

Zwei reguläre Ausdrücke können verknüpft werden. Der reguläre Ausdruck, der sich daraus ergibt, passt auf jeden String, der durch die Verknüpfung zweier Substrings entsteht, auf die die ursprünglichen regulären Ausdrücke passen.

Zwei reguläre Ausdrücke können durch “|” kombiniert werden. Der daraus entstandene Ausdruck beschreibt eine Zeichenkette, in der entweder der eine oder der andere Ausdruck auftritt.

Wiederholungsoperatoren haben Vorrang vor einfacher Aneinanderreihung, die wiederum Vorrang vor der Auswahl hat. Ein Teil-Ausdruck darf durch runde Klammern (“(”, “)”) umschlossen werden, um diese Regeln aufzuheben.

Hinweis: Wenn Mutt mit dem GNU *rx*-Paket kompiliert wurde, können auch die folgenden Operatoren in regulären Ausdrücken verwendet werden.

\\y

Beschreibt den leeren String am Anfang oder Ende eines Wortes.

\\B

Beschreibt den leeren String innerhalb eines Wortes.

\\<

Beschreibt den leeren String am Anfang eines Wortes.

`\\>`

Beschreibt den leeren String am Ende eines Wortes.

`\\w`

Steht für jedes Zeichen, das in einem Wort vorkommen kann (Buchstaben, Ziffern oder Unterstrich “_”).

`\\W`

Steht für jedes Zeichen, das nicht in einem Wort vorkommen kann.

`\\‘`

Beschreibt den leeren String am Anfang eines Puffers (String).

`\\’`

Beschreibt den leeren String am Ende eines Puffers.

Beachte, dass diese Operatoren nicht von POSIX definiert und deshalb möglicherweise nicht in den Bibliotheken aller Systeme enthalten sind.

4.2 Suchmuster

Viele von Mutts Kommandos ermöglichen die Angabe eines Suchmusters (`limit`, `tag-pattern`, `delete-pattern`, etc.). Es gibt verschiedene Kriterien für die Auswahl von Nachrichten:

<code>~A</code>	Alle Nachrichten
<code>~b AUSDRUCK</code>	Nachrichten, in deren Body AUSDRUCK vorkommt
<code>~B AUSDRUCK</code>	Nachrichten, in deren Body oder Header AUSDRUCK vorkommt
<code>~c NAME</code>	Nachrichten die als Kopie (Cc) an NAME gehen
<code>~C AUSDRUCK</code>	Nachricht mit AUSDRUCK in To: oder Cc:
<code>~D</code>	gelöschte Nachrichten
<code>~d [MIN]-[MAX]</code>	Nachrichten mit ‘‘date-sent’’ in einem bestimmten Zeitraum
<code>~E</code>	veraltete Nachrichten (expire)
<code>~e AUSDRUCK</code>	Nachrichten mit AUSDRUCK im ‘‘Sender’’-Feld
<code>~F</code>	als wichtig markierte Nachrichten
<code>~f NAME</code>	Nachrichten von NAME
<code>~g</code>	mit PGP signierte Nachrichten
<code>~G</code>	mit PGP verschlüsselte Nachrichten
<code>~h AUSDRUCK</code>	Nachrichten mit AUSDRUCK im Header
<code>~k</code>	Nachricht enthält einen PGP-Schlüssel
<code>~i ID</code>	Nachricht mit ID im ‘‘Message-ID’’-Feld
<code>~L AUSDRUCK</code>	Nachricht ist von AUSDRUCK oder wurde von AUSDRUCK empfangen
<code>~l</code>	Nachricht ist an eine bekannte Mailingliste adressiert
<code>~m [MIN]-[MAX]</code>	Nachrichten von MIN bis MAX *)
<code>~n [MIN]-[MAX]</code>	Nachrichten mit einer Punktzahl (Score) im Bereich MIN bis MAX *)
<code>~N</code>	neue Nachrichten
<code>~O</code>	alte Nachrichten
<code>~p</code>	Nachricht ist an dich adressiert (verwendet \$alternates)
<code>~P</code>	Nachricht ist von dir (verwendet \$alternates)
<code>~Q</code>	bereits beantwortete Nachrichten

<code>~R</code>	gelesene Nachrichten
<code>~r [MIN]-[MAX]</code>	Nachrichten mit ‘‘date-received’’ im angegebenen Zeitraum
<code>~S</code>	veraltete Nachrichten (supersede)
<code>~s SUBJECT</code>	Nachrichten mit SUBJECT im ‘‘Subject’’-Feld (Betreff)
<code>~T</code>	markierte Nachrichten
<code>~t NAME</code>	Nachrichten an NAME
<code>~U</code>	ungelesene Nachrichten
<code>~v</code>	Nachricht ist Teil eines nicht angezeigten Threads
<code>~x AUSDRUCK</code>	Nachricht, die AUSDRUCK im ‘‘References’’-Feld enthält
<code>~y AUSDRUCK</code>	Nachricht, die AUSDRUCK im ‘‘X-Label’’-Feld enthält
<code>~z [MIN]-[MAX]</code>	Nachrichten mit einer Größe von MIN bis MAX *)
<code>~=</code>	doppelte Nachrichten (siehe <code>\$duplicate_threads</code>)

Dabei handelt es sich bei AUSDRUCK, NAME, ID und SUBJECT um 4.1 (reguläre Ausdrücke).

*) Die Formen `<[MAX]`, `>[MIN]`, `[MIN]-` und `-[MAX]` sind ebenfalls zulässig.

4.2.1 Anpassen von Suchmustern

Beachte, dass Suchmuster, die sich auf Adress-Listen beziehen (c, C, p, P und t) bereits zutreffen, wenn sie auf mindestens einen Eintrag in der ganzen Liste zutreffen. Wenn alle Elemente der Liste geprüft werden sollen, muss dem Muster ein ‘‘^’’ vorangestellt werden. Dieses Beispiel erfasst alle Nachrichten, die nur Empfänger in Deutschland haben.

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

4.2.2 Komplexe Muster

Das logische UND wird durch Angabe von mehr als einem Kriterium erreicht. Beispiel:

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

Hier würden alle Nachrichten ausgewählt, die das Wort ‘‘mutt’’ in der Liste der Empfänger **und** das Wort ‘‘elkins’’ im ‘‘From’’-Header haben.

Mutt erkennt außerdem die folgenden Operatoren, mit denen komplexere Suchmuster erstellt werden können:

- `!` – logischer NICHT-Operator
- `|` – logischer ODER-Operator
- `()` – logischer Gruppierungs-Operator

Hier ist ein Beispiel, das ein komplexes Suchmuster zeigt. Das Muster trifft auf alle Nachrichten zu, die nicht ‘‘mutt’’ im ‘‘To’’- oder ‘‘Cc’’-Feld enthalten und die von ‘‘elkins’’ sind.

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

Hier ist ein Beispiel, das innerhalb des regulären Ausdrucks Leerzeichen verwendet (beachte die Begrenzungszeichen ‘‘ ’ ’ und ‘‘) ’ ’). Damit es zutrifft, muss auf das Subject der Mail ‘‘^Junk +From +Me\$’’ zutreffen und die Mail muss entweder von ‘‘Jim +Somebody’’ oder von ‘‘Ed +SomeoneElse’’ stammen:

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

4.2.3 Suchen nach Datum

Mutt unterstützt *absolute* und *relative* Datumsangaben.

Absolut: Das Datum **muss** im TT/MM/JJ-Format eingegeben werden (Monat und Jahr sind optional und beziehen sich bei Nichtangabe auf den aktuellen Monat und das aktuelle Jahr). Ein Beispiel für einen gültigen Datumsbereich:

Begrenze auf Nachrichten nach Muster: `~d 20/1/95-31/10`

Wenn das Mindest-Datum (die erste Angabe) ausgelassen und nur “-TT/MM/JJ” angegeben wird, werden alle Nachrichten *vor* diesem Datum ausgewählt. Wenn das zweite Datum ausgelassen und nur “TT/MM/JJ” angegeben wird, werden alle Nachrichten *nach* diesem Datum ausgewählt. Wenn nur ein Datum ohne “-” verwendet wird, werden nur an diesem Datum versandte Nachrichten ausgewählt.

Fehlergrenzen. Absolute Daten kann man mit Fehlergrenzen versehen. Eine Fehlergrenze ist ein Vorzeichen (+ oder -), gefolgt von einer Zahl und einer der folgenden Einheiten:

y	Jahre
m	Monate
w	Wochen
d	Tage

Als Spezialfall kann man das Vorzeichen durch das Zeichen “*” ersetzen; dies ist äquivalent dazu, eine identische positive und negative Fehlergrenze anzugeben.

Beispiel: Um alle Nachrichten auszuwählen, die innerhalb von zwei Wochen um den 15. Januar 2001 verschickt wurden, würde man das folgende Muster verwenden:

Begrenze auf Nachrichten nach Muster: `~d 15/1/2001*2w`

Relativ. Dieser Datumstyp ist relativ zum aktuellen Datum und kann so angegeben werden:

- *>Abweichung* (Nachrichten, die älter als *Abweichung* Einheiten sind)
- *<Abweichung* (Nachrichten, die neuer als *Abweichung* Einheiten sind)
- *=Abweichung* (Nachrichten, die genau *Abweichung* Einheiten alt sind)

Die *Abweichung* wird als positive Zahl mit einer der folgenden Einheiten verwendet.

y	Jahre
m	Monate
w	Wochen
d	Tage

Beispiel: Für die Auswahl aller Nachrichten, die jünger als einen Monat sind, kann man dieses Muster verwenden:

Begrenze auf Nachrichten nach Muster: `~d <1m`

Hinweis: Alle Datumsangaben, die beim Suchen verwendet werden, beziehen sich auf die **lokale** Zeitzone. Wenn die 6.3.83 (\$index_format)-Variable kein %[...] -Format enthält, ist das **nicht** das Datum, das im Index angezeigt wird.

4.3 Verwendung von Markierungen (Tags)

Manchmal soll eine Operation mit einer ganzen Gruppe von Nachrichten auf einmal ausgeführt werden anstatt nur mit einer. Ein Beispiel könnte das Speichern von Nachrichten einer Mailingliste in eine andere Mailbox oder das Löschen aller Nachrichten mit einem bestimmten Subject (Betreff) sein. Um Nachrichten zu markieren, auf die ein vom Benutzer angegebenes Muster zutrifft, kann die “tag-pattern”-Funktion benutzt werden, die per Voreinstellung auf “shift-T” (im Deutschen manchmal “Umschalt-T” genannt, *A. d. Ü.*) gelegt ist. Eine andere Möglichkeit ist es, verschiedene Nachrichten manuell mit der “tag-message”-Funktion zu markieren, die sich standardmäßig auf der Taste “t” befindet. Das Kapitel 4.2 (Muster) enthält nähere Informationen über die erwartete Syntax.

Sobald die gewünschten Nachrichten markiert sind, kann der “tag-prefix” Operator benutzt werden, der sich in der Voreinstellung auf der “;”-Taste (Semikolon) befindet. Wenn der “tag-prefix”-Operator benutzt wird, bezieht sich die **nächste** Operation auf alle markierten Nachrichten, sofern dies mit der Art der Operation vereinbar ist. Wenn die 6.3.17 (\$auto_tag)-Variable gesetzt ist, beziehen sich Operationen - sofern Markierungen existieren - automatisch auf die markierten Nachrichten, ohne dass “tag-prefix” verwendet werden muss.

4.4 Das Arbeiten mit Hooks (“Haken”)

Hooks sind ein Konzept, das aus dem Editor Emacs stammt und dem Benutzer das Ausführen beliebiger Kommandos gestattet, bevor eine bestimmte Operation durchgeführt wird. Beispielsweise könnte man seine Konfiguration davon abhängig machen, in welcher Mailbox man gerade Nachrichten liest oder wem man eine Mail schickt. In der Welt von Mutt besteht ein *Hook* aus einem 4.1 (regulären Ausdruck) oder 4.2 (Muster) zusammen mit einer Konfigurations-Option oder einem Kommando. Über Details der einzelnen Hooks geben diese Kapitel Auskunft:

- 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)

Anmerkung: Wenn ein Hook Konfigurations-Einstellungen verändert, bleiben diese Änderungen bis zum Ende der aktuellen mutt-Sitzung wirksam. Da dies in den meisten Fällen nicht gewünscht ist, muss man vor allen anderen Hooks einen speziellen Hook einfügen, der immer ausgeführt wird und die Voreinstellung wieder herstellt. Es folgt ein Beispiel mit send-hook und der my_hdr Anweisung:

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

4.4.1 Auswählen von Nachrichten durch Hooks

Hooks, die sich auf ganze Nachrichten beziehen (`send-hook`, `save-hook`, `fcc-hook`, `message-hook`), werden besonders ausgewertet. Für die anderen Arten von Hooks ist ein 4.1 (regulärer Ausdruck) ausreichend.

Für die Arbeit mit Nachrichten wird allerdings eine feinere Kontrolle nötig, da man je nach Aufgabenstellung unterschiedliche Kriterien benötigt.

Mutt erlaubt hierfür die Benutzung von 4.2 (Suchmustern) in den Hook-Kommandos. Dies funktioniert genau wie bei der Auswahl der angezeigten Nachrichten oder dem Durchsuchen der Mailbox, wobei man allerdings nur auf die Operatoren zurückgreifen kann, die sich auf Informationen beziehen, die Mutt aus dem Header (Kopf) der Nachricht erhalten kann (z. B. From, To, Cc, Date, Subject usw.).

Wenn man beispielsweise den eigenen Absender davon abhängig machen möchte, wem man gerade Mail schickt, könnte man folgendes tun:

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

Dieses Kommando würde immer dann ausgeführt, wenn Mail an *me@cs.hmc.edu* geschickt wird.

Natürlich muss für ein solches Muster nicht die vollständige in Mutt verfügbare Suchsprache genutzt werden. Man kann natürlich - wie bei anderen Hooks auch - einen einfachen *regulären Ausdruck* verwenden, den Mutt dann in eine für die Suchsprache verständliche Form umwandelt. Dafür wird die Variable 6.3.31 (`$default_hook`) verwendet. Das Suchmuster wird umgewandelt, sobald der Hook deklariert ist. Dafür wird der zu dieser Zeit gültige Wert von 6.3.31 (`$default_hook`) verwendet.

4.5 Adressbuch-Einträge aus externen Quellen

Mutt unterstützt die Anbindung an externe Suchverzeichnisse wie LDAP, ph/qi, bddb oder NIS durch ein "Wrapper-Script", das die Daten über eine einfache Schnittstelle an Mutt übergibt. Durch die Variable 6.3.174 (`$query_command`) wird Mutt angewiesen, ein bestimmtes Programm zu benutzen. Beispiel:

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

Dieses Programm/Skript sollte die Abfrage über die Kommandozeile entgegennehmen. Es muss eine einzeilige Statusmeldung und dann jeden passenden Eintrag auf einer gesonderten Zeile ausgeben. Jede Zeile besteht dabei aus einer Adresse, dem Namen und zusätzlichen optionalen Informationen, durch Tabulatorzeichen getrennt. Falls ein Fehler auftritt oder keine passende Adresse gefunden wird, muss ein Fehlercode ungleich Null zurückgeliefert und eine einzeilige Fehlermeldung ausgegeben werden.

Hier eine mögliche Ausgabe:

```
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
```

Es gibt zwei Mechanismen für den Zugriff auf Mutts Abfragefunktion. Die eine ist die Abfrage innerhalb des Index-Menüs durch die Funktion *query* (Voreinstellung: Q). Dabei wird der Benutzer nach einer Eingabe gefragt und schließlich ein Menü mit allen passenden Einträgen gezeigt. In diesem Menü können Adressen ausgewählt werden, an die Mail verschickt oder für die ein Alias erstellt werden soll. Außerdem können mehrere Nachrichten zum Mailen markiert, eine neue Abfrage gestartet oder das Ergebnis einer neuen Abfrage der Liste hinzugefügt werden.

Der andere Mechanismus für den Zugriff auf die *query*-Funktion ist das Vervollständigen von Adressen, ähnlich dem Vervollständigen von Alias-Namen. In einem beliebigen Adressfeld kann die *complete-query*-Funktion (Voreinstellung: ^T) zum Vervollständigen der gerade eingetippten Adresse verwendet werden. Wie bei einem Alias schaut Mutt von der aktuellen Cursor-Position zurück bis zum letzten Leerzeichen oder Komma. Wenn das Ergebnis der Abfrage eindeutig ist, wird es an Ort und Stelle eingebunden. Wenn nicht, zeigt Mutt das Query-Menü, wo ein oder mehrere Einträge ausgewählt werden können.

4.6 Mailbox-Formate

Mutt unterstützt das Lesen und Schreiben von vier verschiedenen Mailbox-Formaten: mbox, MMDF, MH und Maildir. Der Typ der Mailbox wird automatisch erkannt. Es ist also unnötig, die Mailboxen irgendwie kenntlich zu machen. Wenn neue Mailboxen erstellt werden, benutzt Mutt die Standardeinstellung aus der Variablen 6.3.95 (`$mbox_type`).

mbox. Dies ist das am weitesten verbreitete Mailbox-Format unter UNIX. Alle Nachrichten werden in einer einzigen Datei gespeichert. Jede Nachricht besitzt eine Zeile der Form:

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

um den Beginn einer neuen Nachricht kenntlich zu machen (sie wird oft als die “From_”-Zeile bezeichnet).

MMDF. Dies ist eine Variante des *mbox*-Formates. Jede Nachricht wird von Zeilen mit “^A^A^A^A” (vier Control-A’s) umschlossen.

MH. Hier handelt es sich um eine radikale Abkehr vom *mbox*- und *MMDF*-Format. Diese Mailbox besteht aus einem Verzeichnis, in dem jede Nachricht in einer eigenen Datei gespeichert wird. Der Dateiname steht für die Nummer der Nachricht (was allerdings nicht die Nummer sein muss, die Mutt anzeigt). Gelöschte Nachrichten werden mit einem Komma (,) als erstem Zeichen des Dateinamens kenntlich gemacht. **Hinweis:** Mutt erkennt diesen Typ von Mailbox, indem es nach einer der Dateien `.mh_sequences` oder `.xmhcache` sucht (so werden auch normale Verzeichnisse von MH-Mailboxen unterschieden).

Maildir. Dies ist das neueste der Mailbox-Formate. Es wird vom MTA Qmail (einem sendmail-Ersatz) benutzt. Es ähnelt *MH*, fügt jedoch drei Unterverzeichnisse in die Mailbox ein: *tmp*, *new* und *cur*. Dateinamen für die Nachrichten werden so gewählt, dass sie selbst dann eindeutig sind, wenn zwei Programme über NFS in die Mailbox schreiben. Es ist also kein Locking (“Sperrern” von Dateien) nötig.

4.7 Kurzschreibweisen für Mailboxen

Es gibt eine Reihe von eingebauten Kürzeln, die sich auf bestimmte Mailboxen beziehen. Diese können immer benutzt werden, wenn das Programm die Eingabe von Dateinamen oder Pfaden erwartet.

- ! – Mailbox, in der neue Mail ankommt (6.3.213 (`$spoolfile`))
- > – Mailbox, in die gelesene Mail verschoben wird (6.3.94 (`$mbox`))
- < – Mailbox, in der ausgehende Mail gespeichert wird (6.3.181 (`$record`))
- - oder !! – die zuletzt geöffnete Datei
- ~ – das Home-Verzeichnis
- = oder + – das 6.3.48 (`$folder`)-Verzeichnis
- @*alias* – 3.14 (Standard-Datei) für Mail, die durch die Adresse von *alias* festgelegt wird.

4.8 Der Umgang mit Mailinglisten

Mutt besitzt eine Reihe von Konfigurationseinstellungen, die den Umgang mit größeren Mengen an Mail einfacher machen. Zuerst muss Mutt mitgeteilt werden, welche Adressen es als Mailinglisten betrachten soll (technisch gesehen muss das nicht zwingend eine Mailingliste sein, aber zu diesem Zweck wird es häufig genutzt). Dies wird durch die Kommandos 3.9 (`lists` und `subscribe`) in der `.muttrc` erreicht.

Nachdem Mutt jetzt weiß, bei welchen Adressen es sich um Mailinglisten handelt, kann das Programm verschiedenes tun. Die erste Möglichkeit ist, im Index-Menü den Namen der Mailingliste anzuzeigen, über die du eine Nachricht empfangen hast (d.h. von einer abonnierten Liste). Das ist nützlich, um zwischen persönlichen Nachrichten und Nachrichten von der Liste in der gleichen Mailbox zu unterscheiden. In der Variablen 6.3.83 (`$index_format`) steht jetzt “%L” für “To: <list>”, sofern “list” im “To”-Feld vorkommt bzw. “Cc <list>” wenn “list” im “Cc”-Feld vorkommt (sonst wird der Name des Autors angezeigt).

Oft werden die “To”- und “Cc”-Felder in Mailinglisten recht groß. Die meisten Leute kümmern sich nicht darum, den Autor, dem sie antworten, aus der Liste der Empfänger zu entfernen, was dazu führt, dass zwei oder mehr Kopien bei der gleichen Person ankommen. Die “list-reply”-Funktion, die in der Voreinstellung auf der Taste “L” im Menü *index* und im *pager* liegt, hilft, dies zu vermeiden, indem die Antwort nur an die Liste und nicht an alle Autoren geschickt wird (außer, wenn dies im Header-Feld **Mail-Followup-To** gewünscht wird).

Mutt unterstützt auch den Header **Mail-Followup-To**. Wenn du eine Nachricht an eine Gruppe von Empfängern schickst, die eine oder mehrere abonnierte Mailinglisten enthält und die Option 6.3.50 (`$followup_to`) gesetzt ist, generiert Mutt einen **Mail-Followup-To** Header, der alle Empfänger enthält, denen du die Nachricht schickst, nicht jedoch deine eigene Adresse. Damit wird festgelegt, dass Gruppen- oder Listen-Antworten (auch als “Followups” bekannt) auf diese Nachricht nur an die ursprünglichen Adressaten der Mail geschickt werden sollen, nicht einzeln an deine Adresse - du wirst die Antwort sowieso über eine der Mailinglisten erhalten, die du abonniert hast.

Umgekehrt beachtet Mutt diesen Header, wenn du eine Gruppen- oder Listen-Antwort auf eine Nachricht schreibst, die einen **Mail-Followup-To** Header enthält und die Variable 6.3.66 (`$honor_followup_to`) gesetzt ist. Die Funktion `list-reply` (Listen-Antwort) stellt in diesem Fall ebenfalls sicher, dass die Antwort auch an die Mailingliste geht, selbst wenn ihre Adresse im **Mail-Followup-To** nicht auftaucht.

Wenn das Verändern der Header im Editor aktiviert ist, kannst du manuell einen **Mail-Followup-To** Header hinzufügen. Mutt wird diesen Header nur dann automatisch erstellen, wenn er beim Versenden noch nicht existiert.

Manche Administratoren von Mailinglisten erzeugen ein “Reply-To”-Feld, das statt auf den Autor auf die Mailingliste zeigt. Das kann zu Problemen führen, wenn man dem Autor direkt antworten möchte, da die meisten Mailprogramme automatisch die Adresse im “Reply-To”-Feld verwenden. Mutt benutzt die Variable 6.3.184 (`$reply_to`) um zu entscheiden, welche Adresse benutzt werden soll. Wenn sie gesetzt ist, wird der Benutzer gefragt, ob er die im “Reply-To”- oder die im “From”-Feld angegebene Adresse verwenden will. Ist diese Variable nicht gesetzt, wird das “Reply-To”-Feld benutzt, sofern vorhanden.

Weiterhin kann das Header-Feld “X-Label:” verwendet werden, um Mailinglisten oder ihr Thema zu nennen (oder ganz einfach, um einzelne Nachrichten zu kommentieren). Die Platzhalter “%y” und “%Y” können in der Variablen 6.3.83 (`$index_format`) verwendet werden, um “X-Label:”-Felder im Index anzuzeigen, und in den Suchmustern von Mutt kann man mittels “y” “X-Label:”-Felder nach regulären Ausdrücken durchforschen. Das Header-Feld “X-Label:” folgt keinem Standard, man kann es aber leicht von `procmail` oder einem anderen Mail-Filter eingefügen lassen.

Schließlich bietet Mutt noch die Möglichkeit, die Mailbox nach 2.3.3 (Threads) zu 6.3.208 (sortieren). Ein Thread ist eine Gruppe von Nachrichten, die alle dasselbe Subject (Betreff) tragen. Üblicherweise wird zur grafischen Darstellung eine baum-ähnliche Anordnung benutzt. Dieses Konzept ist auch in den meisten Newsreadern anzutreffen. Es macht den Umgang mit großen Mailinglisten einfacher, weil man auf einfache Weise uninteressante Threads löschen und interessante Themen finden kann.

4.9 Unterstützung von Delivery Status Notification (DSN)

RFC 1894 definiert eine Reihe von MIME Content-Typen zur Weiterleitung des Status von E-Mail. Man kann sie sich als "Empfangsbescheinigung" vorstellen. Berkeley sendmail 8.8.x besitzt derzeit einige Kommandozeilen-Optionen, durch die ein Mailprogramm festlegen kann, welche Art von Status-Nachrichten zurückgegeben werden sollen.

Um dies zu unterstützen, wurden zwei Variablen eingeführt. 6.3.37 (`$dsn_notify`) dient dazu, Rückmeldungen anzufordern, wenn verschiedene Ereignisse eintreten (z.B. Zustellung misslungen, Nachricht zugestellt etc.). 6.3.38 (`$dsn_return`) legt fest, ob dabei die ganze Nachricht oder nur die Kopfzeilen (Header) zurückgeliefert werden sollen. Die Manpage zu sendmail enthält weitere Details zu DSN.

4.10 POP3 Unterstützung (OPTIONAL)

Wenn Mutt mit der POP3-Unterstützung kompiliert wurde (durch Angabe des Schalters `-enable-pop` beim Aufruf des `configure`-Scripts), kann das Programm auf Mailboxen zugreifen, die auf einem entfernten POP3-Server liegen und auch Mail zum lokalen Lesen herunterladen.

Du kannst eine entfernte POP3-Mailbox ansprechen, indem du als Mailbox `pop://popserver/` angibst.

Es ist möglich, einen anderen Port zu verwenden, indem man ihn nach dem Servernamen angibt, d.h.: `pop://popserver:port/`.

Man kann auch für jede Mailbox einen anderen Benutzernamen angeben:

```
pop://benutzername@popserver[:port]/
```

Eine Mailbox über POP3 auf neue Mail zu überprüfen ist aufwendiger, als wenn sie lokal verfügbar ist. Deshalb kann man mit der Variablen 6.3.158 (`$pop_checkinterval`) einstellen, wie oft Mutt bei einem entfernten Server nach neuer Mail fragt; die Voreinstellung sind alle 60 Sekunden.

Wenn Mutt mit SSL-Unterstützung kompiliert wurde (indem dem `configure`-Skript die Option `-with-ssl` übergeben wurde), können Verbindungen mit POP3-Servern verschlüsselt werden. Natürlich setzt das voraus, dass der Server mit SSL verschlüsselte Verbindungen unterstützt. Um auf eine Mailbox über POP3/SSL zuzugreifen, verwendet man das Präfix `pops:`, also:

```
pops://[benutzername@]popserver[:port]/
```

Eine andere Möglichkeit über POP3 an deine Mail zu kommen, ist die Funktion `fetch-mail` (Voreinstellung: G). Sie baut eine Verbindung zu 6.3.160 (`pop_host`) auf, holt von dort alle neu eingetroffenen Nachrichten und legt sie in der lokalen 6.3.213 (Mailbox) ab. Von diesem Punkt an verhält sich Mutt so, als ob die Mails schon immer lokal vorhanden gewesen wären.

Hinweis: Wenn du nur alle Nachrichten abholen und in die lokale Mailbox schreiben willst, solltest du dir überlegen, ob du dazu nicht ein spezielles Programm wie `fetchmail` benutzen willst.

4.11 IMAP-Unterstützung (OPTIONAL)

Wenn Mutt mit IMAP-Unterstützung kompiliert wurde (durch Angabe des Schalters `-enable-imap` beim Aufruf des `configure`-Scripts), kann es mit Mailboxen auf einem entfernten IMAP-Server umgehen.

Zugriff auf die entfernte Mail-Eingangsbox ist durch Auswählen der Mailbox `imap://imapserver/INBOX` möglich, wobei `imapserver` der Name des IMAP-Servers und `INBOX` der Name der Spool-Mailbox auf dem IMAP Server ist. Um Zugriff auf eine andere Mailbox auf dem Server zu erlangen, kann man `imap://imapserver/Pfad/zur/Mailbox` angeben, wobei `Pfad/zur/Mailbox` der Pfad der Mailbox ist.

Ein anderer Port kann so angegeben werden: `imap://imapserver:port/INBOX`.

Man kann auch für jeden Mail-Ordner einen anderen Benutzernamen angeben, z.B.:

```
imap://benutzername@imapserver[:port]/INBOX
```

Wenn Mutt mit SSL-Unterstützung kompiliert wurde (dazu muss *configure* mit der Option *-with-ssl* aufgerufen werden), können Verbindungen zu IMAP-Servern verschlüsselt werden. Das setzt selbstverständlich voraus, dass der Server ebenfalls SLL-verschlüsselte Verbindungen unterstützt. Um eine Mailbox mit IMAP/SSL zu lesen, solltest Du als Pfad `imaps://[benutzername@]imapserver[:port]/Pfad/zur/Mailbox` verwenden.

Eine Pine-kompatible Notation wird ebenfalls unterstützt, also

```
{[benutzername@]imapserver[:port] [/ssl]}Pfad/zur/Mailbox
```

Nicht alle Server verwenden “/” als Hierarchie-Trenner. Mutt sollte erkennen, welches Trennzeichen der Server benutzt und die Pfade entsprechend umschreiben.

Wenn man sich in den Mailboxen auf dem IMAP-Server umschaute, kann man mit dem Kommando *toggle-subscribed* einstellen, ob man nur die abonnierten oder alle Mailboxen betrachten will. Siehe dazu auch die Variable 6.3.74 (`$imap_list_subscribed`).

Einen IMAP-Server auf neu eingetroffene Nachrichten zu überprüfen kann spürbare Verzögerungen nach sich ziehen. Aus diesem Grund solltest du die Variablen 6.3.87 (`$mail_check`) und 6.3.223 (`$timeout`) sorgfältig an deine Bedürfnisse anpassen. Ich persönlich verwende

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

und erhalte damit bei meiner langsamen Modem-Leitung relativ gute Ergebnisse.

Hinweis: Beim UW Server ist es vor v12.250 vorgekommen, dass der Server, wenn mbox als Speichermethode verwendet wurde, die Verbindung zu einem Client unterbrach, sobald ein anderer Client den gleichen Ordner auswählte.

4.11.1 Auswahl der Mailboxen

Seit Version 1.2 bietet Mutt ein Menü, um Mailboxen auf einem IMAP-Server anzuzeigen und auszuwählen. Es funktioniert sehr ähnlich wie das Menü zur Auswahl lokaler Dateien; allerdings gibt es folgende Unterschiede:

- Statt der Zugriffsrechte zeigt Mutt die Zeichenkette `IMAPn`, ggf. gefolgt vom Symbol “+”, das anzeigt, wenn ein Eintrag sowohl Nachrichten als auch Unterordner enthält. Auf Cyrus ähnelnden Servern tritt diese Situation häufig auf.
- Für den Fall, in dem ein Eintrag sowohl Nachrichten als auch Unterordner enthält, wird die Auswahl-Taste (in der Voreinstellung `enter`) bewirken, dass Mutt den Unterordner anzeigt. Wenn du in dieser Situation die Nachrichten in dem Ordner angezeigt haben willst, musst du statt dessen `view-file` verwenden (in der Voreinstellung `space`).
- Man kann Mailboxen mit dem Befehl `delete-mailbox` (Voreinstellung: `d`) löschen. Man kann Mailboxen auch abonnieren (`subscribe`, normalerweise `s`) oder ein Abonnement aufheben (`unsubscribe`, normalerweise `u`).

4.11.2 Anmeldung am Server

Mutt unterstützt vier Methoden, sich bei einem IMAP-Server anzumelden: SASL, GSSAPI, CRAM-MD5 und LOGIN (es gibt einen Patch von Grant Edwards, der für all die armen Exchange-User NTLM-

Authentifizierung möglich macht, aber er ist noch nicht in die stabile Version von Mutt integriert worden). Es gibt auch Unterstützung für das Pseudo-Protokoll ANONYMOUS, das es einem ermöglicht sich auf einem öffentlichen IMAP-Server einzuloggen, auch wenn man dort kein Benutzerkonto hat. Um ANONYMOUS zu verwenden, braucht man nur den Benutzernamen leer zu lassen oder aus `anonymous` zu setzen.

SASL ist eine besondere Art der Authentifizierung, die zwischen verschiedenen Protokollen (darunter GSSAPI, CRAM-MD5, ANONYMOUS und DIGEST-MD5) die sicherste Methode auswählt, die auf deinem Rechner und dem Server verfügbar ist. Bei einigen dieser Methoden (darunter DIGEST-MD5 und möglicherweise GSSAPI) wird die komplette Sitzung verschlüsselt und damit für die umherwimmelnden Netzwerk-Schnüffler unlesbar. Man sollte diese Option nutzen, wenn man sie hat. Um sie einzusetzen, muss man die Cyrus SASL-Bibliothek auf seinem System installiert haben und Mutt mit der Option `-with-sasl` kompilieren.

Mutt wird jede Methode ausprobieren, die einkompiliert wurde und vom Server angeboten wird, wobei es die folgende Reihenfolge verwendet: SASL, ANONYMOUS, GSSAPI, CRAM-MD5, LOGIN.

Es gibt einige Variablen, um den Anmeldevorgang zu steuern:

- 6.3.79 (`$imap_user`) - bestimmt den Benutzernamen, unter dem (unabhängig von der Methode) die Anmeldung auf dem IMAP-Server versucht wird. Dies kann überschrieben werden, indem man im Pfad der Mailbox ausdrücklich einen Benutzernamen angibt (d.h. einen Mailbox-Namen der Form `benutzer@rechner` verwendet).
- 6.3.75 (`$imap_pass`) - ein Passwort, das man voreinstellen kann; es wird bei allen Authentifizierungsmethoden verwendet, die ein Passwort erfordern.
- 6.3.69 (`$imap_authenticators`) - eine durch Doppelpunkte begrenzte Liste von IMAP Authentifizierungsmethoden, die Mutt ausprobieren soll, in der Reihenfolge, in der sie probiert werden sollen. Wenn hier ein Wert angegeben wird, überschreibt er die Voreinstellung von Mutt (versuche alles, in der oben genannten Reihenfolge).

4.12 Mehrere IMAP/POP Konten verwalten (OPTIONAL)

Falls du Benutzerkonten auf mehreren IMAP und/oder POP-Servern hast, findest du es vielleicht unbequem und fehleranfällig, all die Authentifizierungseinstellungen zu verwalten. Vielleicht kann der Befehl `account-hook` dir dabei helfen. Dieser Hook funktioniert wie der `folder-hook`, wird aber aufgerufen, wann immer du auf eine entfernte Mailbox zugreifst (also auch innerhalb des Menüs zur Auswahl von Mailboxen), nicht nur wenn du eine Mailbox tatsächlich öffnest.

Hier sind einige Beispiele:

```
account-hook . 'unset imap_user; unset imap_pass; unset tunnel'
account-hook imap://rechner1/ 'set imap_user=me1 imap_pass=foo'
account-hook imap://rechner2/ 'set tunnel="ssh rechner2 /usr/libexec/imapd"'
```

4.13 URLs mit dem Webbrowser anschauen (EXTERN)

Wenn eine Nachricht URLs (*unified resource locator* = WWW-Adresse wie z.B. `http://www.mutt.org/`) enthält, ist es sehr bequem, wenn man diese Adressen in einem Menü angezeigt bekommt und die Möglichkeit hat, sie gleich mit einem Webbrowser zu besuchen. Diese Funktionalität setzt voraus, dass man das externe Programm `urlview` installiert (erhältlich unter `ftp://ftp.guug.de/pub/mutt/contrib/`) und folgende Einstellungen vornimmt:

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

5 Die MIME-Unterstützung

Es wurden große Anstrengungen unternommen, um Mutt zum ersten Textmodus-Mailprogramm zu machen, das MIME unterstützt. Dabei wurde keine Mühe gescheut, die Funktionalität einzubauen, die ein Anwender braucht, wobei auf die Einhaltung der Standards Wert gelegt wurde, wo immer es möglich war. Für den Umgang mit MIME benutzt Mutt zwei zusätzliche Arten von Konfigurationsdateien. Die erste ist die Datei `mime.types`, die Dateinamens-Erweiterungen mit IANA MIME-Typen verknüpft. Die zweite ist die `mailcap`-Datei, die externe Programme zum Umgang mit den unterschiedlichen MIME-Typen angibt.

5.1 Verwendung von MIME in Mutt

Es gibt drei Bereiche/Menüs, in denen Mutt mit MIME in Berührung kommt. Das sind der Pager (beim Anzeigen einer Nachricht), das Attachment-Menü und das Compose-Menü.

5.1.1 MIME-Nachrichten im Pager anzeigen

Wenn eine Nachricht im Index-Menü ausgewählt und im Pager angezeigt wird, dekodiert Mutt die Nachricht in eine lesbare Text-Darstellung. Mutt unterstützt intern eine Reihe von MIME-Typen, u.a. `text/plain`, `text/enriched`, `message/rfc822` und `message/news`. Zusätzlich erkennt die internationale Version von Mutt einige PGP-MIME-Typen; PGP/MIME und `application/pgp` eingeschlossen.

Mutt macht Anhänge (Attachments) durch einige beschreibende Zeilen kenntlich. Diese Zeilen sind so aufgebaut:

```
[-- Anhang #1: Beschreibung --]
[-- Typ: text/plain, Codierung: 7bit, Größe: 10000 --]
```

Bei *Beschreibung* handelt es sich um die Beschreibung oder den Dateinamen des Attachments. Als *Codierung* sind die Formate `7bit/8bit/quoted-printable/base64/binary` möglich.

Wenn Mutt mit einem MIME-Typ nichts anzufangen weiß, zeigt es eine solche Meldung an:

```
[-- image/gif wird nicht unterstützt (benutzen Sie 'v', um diesen Teil anzuzeigen) --]
```

5.1.2 Das Anhänge-Menü

Die voreingestellte Taste für `view-attachments` ist `v`, was zur Anzeige des Anhänge-Menüs für die Nachricht führt. Dieses Menü zeigt eine Liste der Anhänge in der Nachricht an. Aus dem Anhänge-Menü heraus kann man Attachments speichern, drucken, an Programme weiterleiten (pipen), löschen und anzeigen. Diese Operationen können sich auch auf mehrere Anhänge gleichzeitig beziehen, wenn sie vorher markiert wurden und man den `tag-prefix`-Operator benutzt. Man kann außerdem eine Antwort auf die aktuelle Nachricht aus dem Menü heraus abschicken, wobei nur das gerade ausgewählte (oder alle markierten) Attachment(s) in der Antwort zitiert werden. Attachments können entweder als Text oder durch Benutzung der `mailcap`-Einstellungen angezeigt werden.

Schließlich ist es auch möglich, die normalen auf Nachrichten bezogenen Funktionen (wie 2.3.4 (`resend-message`)) und die Antworten- und Weiterleiten-Funktionen) auf Attachments des Typs `message/rfc822` anzuwenden.

Nähere Informationen gibt die Online-Hilfe im Anhänge-Menü.

5.1.3 Das “Nachricht erstellen”-Menü

Das “Nachricht erstellen”-Menü wird vor dem Senden einer Nachricht angezeigt. Es ermöglicht das Bearbeiten der Empfängerliste, des Betreffs (Subjects) und anderer Aspekte der Nachricht. Es enthält außerdem eine Liste der Anhänge der Nachricht, den Hauptteil eingeschlossen. In diesem Menü kann ein Anhang (oder eine Liste markierter Anhänge) gedruckt, kopiert, gefiltert, an Programme weitergeleitet (pipen), editiert, bearbeitet, angezeigt und umbenannt werden. Außerdem können die Anhang-Information, Typ, Kodierung und die Beschreibung verändert werden.

Anhänge werden folgendermaßen angezeigt:

```
- 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>
```

Der Bindestrich ‘-’ zeigt an, dass Mutt die so gekennzeichnete Datei nach dem Senden (bzw. Zurückstellen oder Abbrechen) der Nachricht löschen wird. Das kann mit dem `toggle-unlink`-Kommando (Voreinstellung: `u`) eingestellt werden. Das nächste Feld ist der MIME-Typ, der mit dem `edit-type`-Kommando (Voreinstellung: `^T`) verändert werden kann. Das dritte Feld zeigt die Art der Kodierung an, durch die das Versenden von Binär-Nachrichten über 7bit-Verbindungen möglich wird. Eine Änderung ist durch das Kommando `edit-encoding` möglich (Voreinstellung: `^E`). Das vierte Feld gibt die Größe des Attachments an, gerundet auf Kilo- oder Megabyte. Das nächste Feld enthält den Dateinamen, der mit dem `rename-file`-Kommando (Voreinstellung: `R`) modifiziert werden kann. Im letzten Feld steht eine Beschreibung des Attachments. Eine Änderung ist durch das Kommando `edit-description` (Voreinstellung: `d`) möglich.

5.2 Konfiguration der MIME-Typen in `mime.types`

Wenn ein Attachment an eine Nachricht angehängt wird, durchsucht Mutt die private `mime.types`-Datei in `${HOME}/.mime.types` und dann die systemweite `mime.types`-Datei in `/usr/local/share/mutt/mime.types` oder `/etc/mime.types`.

Die Datei `mime.types` besteht aus Zeilen, die den MIME-Typ und eine durch Leerzeichen abgetrennte Liste von Datei-Erweiterungen enthält. Hier ein Beispiel:

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

Eine beispielhafte `mime.types`-Datei wird bei Mutt mitgeliefert und sollte die meisten gebräuchlichen MIME-Typen enthalten.

Wenn Mutt den MIME-Typ nicht an der Dateinamens-Erweiterung der anzuhängenden Datei erkennen kann, schaut es die Datei selbst an. Wenn sie keine binären Informationen enthält, nimmt Mutt an, dass es sich um eine Textdatei handelt und setzt den Typ auf `text/plain`. Wenn die Datei binäre Informationen enthält, setzt Mutt den Typ auf `application/octet-stream`. Der MIME-Typ kann durch das Kommando `edit-type` (Voreinstellung: `^T`) im Compose-Menü verändert werden. Der MIME-Typ besteht tatsächlich aus einem MIME-Haupt-Typen, gefolgt vom Unter-Typ; beide werden durch ein ‘/’ voneinander getrennt. 6 major Typen (`application`, `text`, `image`, `video`, `audio` und `model`) sind nach diversen Internet-Diskussionen zugelassen worden. Mutt erkennt sie alle, falls ein passender Eintrag in der `mime.types`-Datei vorhanden ist. Es erkennt auch andere MIME-Haupt-Typen, zum Beispiel den Typ “chemical”, der im Zusammenhang mit Molekül-Modellen verwendet wird, um molekulare Daten in verschiedenen Formen an diverse Molekül-Betrachter zu schicken. Unbekannte MIME-Typen sollten nur verwendet werden, wenn man annehmen kann, dass der Empfänger der Nachricht derartige Anhänge erwartet.

5.3 Konfiguration der MIME-Anzeige in mailcap

Mutt unterstützt die MIME-Konfiguration nach RFC 1524, genauer gesagt das im Anhang A von RFC 1524 beschriebene Unix-spezifische Format. Dieses Dateiformat wird oft als das mailcap-Format bezeichnet. Viele Programme, die MIME unterstützen, verwenden das mailcap-Format, um dem Benutzer die Konfiguration der MIME-Typen für alle Programme in derselben Datei zu ermöglichen. Die Programme Netscape, XMOsaic, lynx und metamail benutzen beispielsweise dieses Format.

Für den Umgang mit den verschiedenen MIME-Typen, die Mutt nicht intern behandeln kann, untersucht es eine Reihe von Konfigurationsdateien, um ein geeignetes externes Programm zu finden. Standardmäßig wird nach diesen Dateien gesucht, die in der Liste durch Doppelpunkte voneinander getrennt sind:

```

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

```

Bei \$HOME handelt es sich um das Heimat-Verzeichnis des Benutzers.

Im Normalfall wird das metamail-Paket eine mailcap-Datei in `/usr/local/etc/mailcap` installieren, das einige grundlegende Einträge enthält.

5.3.1 Die Grundlagen der Mailcap-Datei

Eine mailcap-Datei besteht aus einer Reihe von Zeilen, die entweder Kommentar-, Definitions- oder Leerzeilen sind.

Eine Kommentarzeile besteht aus einem #-Zeichen, gefolgt von beliebigem Text.

Eine Leerzeile ist einfach nur leer.

Eine Definitionszeile besteht aus einem Content-Type ("Inhalts-Typ"), einem Anzeige-Kommando und einer beliebigen Anzahl von optionalen Feldern. Jedes Feld in einer Definition wird durch ein Semicolon ';' abgetrennt.

Der Content-Type wird durch die Standard-MIME Methode Typ/Subtyp angegeben. Beispiel: `text/plain`, `text/html`, `image/gif`, usw. Zusätzlich sind im mailcap-Format zwei Platzhalter vorgesehen. Einer ist der spezielle Subtyp '*'. Beim anderen wird nur der Haupt-Typ (major type) angegeben und der Subtyp ausgelassen. Beispielsweise treffen `image/*` oder `video` auf alle Image- bzw. Video-Typen zu.

Das Anzeige-Kommando ist ein Unix-Kommando zur Anzeige des betreffenden MIME-Typs. Es gibt zwei verschiedene Arten von unterstützten Kommandos. Standard ist es, den Rumpf der MIME-Nachricht über die Standardeingabe an das angegebenen Kommando zu schicken. Dies kann geändert werden, indem man %s als Parameter des Anzeige-Kommandos angibt. In diesem Fall speichert Mutt den Rumpf der MIME-Nachricht in einer temporären Datei ab und ruft das Anzeige-Kommando auf, wobei die Folge %s durch den Namen der temporären Datei ersetzt wird. In beiden Fällen übergibt Mutt die Kontrolle über das Terminal an das Anzeige-Programm, bis dieses Programm beendet ist. Dann löscht Mutt die temporäre Datei, sofern sie noch vorhanden ist.

Die einfachste Form ist das Schicken einer `text/plain`-Nachricht an den externen Pager `more` über die Standardeingabe:

```
text/plain; more
```

Eine andere Möglichkeit ist die Verwendung einer temporären Datei:

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

lynx kann zum interaktiven Anzeigen von `text/html`-Nachrichten verwendet werden:

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

In diesem Fall nimmt lynx keine Dateien über die Standardeingabe an. Deshalb muss die %s-Syntax benutzt werden. **Hinweis:** Einige ältere Versionen von lynx enthalten einen Bug, der dafür sorgt, dass lynx die mailcap-Datei nach einem Anzeigeprogramm für text/html durchsucht. Auf diese Weise wird natürlich lynx gefunden, was dafür sorgt, dass sich lynx bis in alle Ewigkeit selbst startet.

Wer lynx nicht interaktiv benutzen möchte, kann text/html nach text/plain umwandeln lassen:

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

Möglicherweise soll lynx als Anzeigeprogramm für text/html-Dateien und ein Pager für alle anderen Formate verwendet werden:

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

Dies ist die einfachste Form einer mailcap-Datei.

5.3.2 Sichere Verwendung der mailcap

Die Auswertung von Shell-Meta-Zeichen, die in MIME-Parameter eingebettet sind, kann generell zu Sicherheits-Problemen führen. Mutt versucht, Parameter korrekt zu maskieren, wenn es die %s-Syntax auflöst. Außerdem vermeidet es riskante Zeichen, indem es sie ersetzt, vergleiche dazu die Variable 6.3.89 (mailcap_sanitize).

Obwohl Mutts Vorgehensweise beim Aufrufen von Programmen über die mailcap sicher erscheint, wird die mailcap auch von anderen Anwendungen eingelesen, die eventuell weniger sorgfältig vorgehen. Deshalb sollte man sich an folgende Regeln halten:

Maskiere die %-Sequenzen nicht selbst. Maskiere sie weder mit einzelnen, noch mit doppelten Anführungszeichen. Mutt übernimmt das und macht es richtig, so wie jedes Programm es tun sollte, das die mailcap liest. Lasse ihren Inhalt nicht durch Backticks (‘) auswerten. Sei generell sehr vorsichtig, wenn etwas von der Shell interpretiert wird (eval statements) und vermeide solche Konstrukte, wo immer es möglich ist. Zu versuchen, fehlerhaftes Verhalten durch reine Anführungszeichen zu korrigieren, erzeugt wieder neue Schlupflöcher - es gibt keine Alternative zu einer korrekten Maskierung, die von Anfang an vom Programm vorgenommen wird, das die mailcap verwendet.

Wenn du den Wert einer %-Sequenz in einem Zusammenhang verwenden willst, in dem du eigene Maskierung oder Backtick-Auswertung brauchst, schreibe den Wert erst in eine Shell-Variable und beziehe dich wo nötig auf diese Variable, wie in folgendem Beispiel gezeigt wird (\$charset innerhalb der Backtick-Auswertung zu verwenden ist sicher, da es selbst nicht weiter interpretiert wird):

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

5.3.3 Fortgeschrittene Nutzung von mailcap

Optionale Felder Zusätzlich zu den benötigten content-type und Anzeige-Kommando-Feldern können, durch Semikola “;” abgetrennt, Felder für zusätzliche Optionen angegeben werden. Mutt erkennt folgende optionale Felder:

copiousoutput

Diese Option teilt Mutt mit, dass das Kommando möglicherweise große Mengen von Text auf die Standardausgabe schreibt. Das führt dazu, dass Mutt einen Pager (entweder den internen oder einen externen Pager, wenn die pager-Variable entsprechend gesetzt ist) benutzt, um das Ergebnis des view-Kommandos anzuzeigen. Ohne diese Option nimmt Mutt an, dass das Kommando interaktiv ist. Man könnte das benutzen, um die Pipe an `more` im `lynx -dump` Beispiel zu ersetzen:

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

Auf diese Weise formatiert lynx den HTML-Code als `text/plain` und Mutt zeigt das Ergebnis im Standard-Pager an.

needsterminal

Mutt benutzt diese Option bei der Anzeige von Anhängen mit 5.4 (autoview), um zu entscheiden, ob die Variable 6.3.232 (`$wait_key`) beachtet werden soll. Wenn ein Anhang durch ein interaktives Programm angezeigt wird und der entsprechende mailcap-Eintrag die *needsterminal*-Option aktiviert hat, entscheidet Mutt anhand von 6.3.232 (`$wait_key`) und dem exit-Status des Programms, ob der Benutzer aufgefordert werden soll, eine Taste zu drücken, wenn das Programm beendet ist. In allen anderen Situationen ist kein Tastendruck nötig.

compose=<Kommando>

Diese Option gibt ein Programm an, mit dem ein neuer Anhang mit dem angegebenen MIME-Typ erstellt werden soll. Mutt unterstützt diese Option im “Nachricht erstellen”-Menü.

composetyped=<Kommando>

Diese Option gibt ein Programm an, mit dem ein neuer Anhang mit dem angegebenen MIME-Typ erstellt werden soll. Es unterscheidet sich dadurch von der `compose`-Option, dass Mutt hier Standard-MIME-Header in den Daten erwartet. Das kann dafür benutzt werden, Parameter, Dateiname, Beschreibung usw. eines neuen Anhangs zu bestimmen. Mutt unterstützt diese Option im “Nachricht erstellen”-Menü.

print=<Kommando>

Diese Option gibt ein Programm an, das zum Drucken eines bestimmten MIME-Typs benutzt werden kann. Mutt unterstützt diese Funktion im “Anhänge-” und “Nachricht erstellen”-Menü.

edit=<Kommando>

Diese Option gibt ein Programm zum Bearbeiten eines bestimmten MIME-Typs an. Mutt unterstützt diese Funktion im “Nachricht erstellen”-Menü und benutzt es außerdem für das Erstellen neuer Anhänge. Mutt benutzt für Text-Anhänge standardmäßig den eingestellten Editor.

nametemplate=<Vorlage>

Dieses Feld gibt das Format des Dateinamens an, der im Kommando-Feld für `%s` eingesetzt wird. Einige Programme benötigen eine bestimmte Dateinamens-Erweiterung, um eine Datei korrekt anzuzeigen. Lynx erkennt z.B. eine Datei nur dann als `text/html` an, wenn der Dateiname auf `.html` endet. Deshalb sollte lynx für die Anzeige von `text/html` in der Mailcap-Datei folgendermaßen konfiguriert sein:

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

test=<Kommando>

Dieses Feld gibt ein Programm an, das testet, ob dieser Mailcap-Eintrag benutzt werden soll. Das Kommando unterliegt den Regeln zur Kommando-Expansions, die im übernächsten Abschnitt beschrieben werden. Wenn das Programm Null zurückgibt, ist der Test bestanden, und Mutt benutzt diesen Eintrag. Wenn das Programm einen Wert ungleich Null zurückgibt, ist der Test fehlgeschlagen und Mutt sucht weiter nach einen passenden Eintrag.

Hinweis: *Der Content-Typ muss zutreffen, bevor Mutt den Test vornimmt.*

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

In diesem Beispiel führt Mutt das Programm “RunningX” aus, das Null zurückgibt, wenn X11 läuft. Wenn RunningX Null zurückliefert, startet Mutt netscape zur Anzeige des text/html-Objekts. Wenn X nicht läuft, geht Mutt zum nächsten Eintrag und zeigt das Objekt mit lynx an.

Such-Reihenfolge Mutt sucht in der Mailcap-Datei nach dem nützlichsten Eintrag für seine Zwecke. Wenn beispielsweise ein image/gif-Objekt ausgedruckt werden soll und die folgenden Einträge in der Mailcap-Datei stehen, sucht Mutt nach einem Eintrag, der das print-Kommando enthält:

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

Mutt überspringt den image/*-Eintrag und benutzt statt dessen den print-Eintrag von image/gif.

Zusätzlich können durch das 5.4 (Autoview)-Kommando zwei Programme zur Anzeige eines Anhangs angegeben werden. Eines für die automatische Anzeige und das andere für die interaktive Verwendung im Anhänge-Menü. Außerdem kann das Test-Feature benutzt werden, um ein Anzeigeprogramm abhängig von Umgebungsvariablen auszuwählen.

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

Für 5.4 (Autoview) benutzt Mutt den dritten Eintrag, da dieser die copiousoutput-Option enthält. Für die interaktive Anzeige startet Mutt das Programm RunningX, um festzustellen, ob der erste Eintrag benutzt werden soll. Wenn das Programm einen Wert ungleich Null zurückliefert, wird der zweite Eintrag für die interaktive Anzeige benutzt.

Kommando-Expansion Die verschiedenen in den Mailcap-Dateien angegebenen Kommandos werden durch die system()-Funktion an /bin/sh weitergereicht. Bevor ein Kommando an /bin/sh -c geschickt wird, ersetzt Mutt spezielle Parameter. Diese Parameter erkennt Mutt:

%s

Wie in den vorhergehenden Kapiteln beschrieben, wird diese Variable durch einen Dateinamen ersetzt, den das aufrufende Programm festlegt. Er gibt eine Datei an, die den Rumpf der Nachricht enthält, die angezeigt / gedruckt / bearbeitet werden soll oder legt fest, wo ein Programm sein Ergebnis abspeichern soll. Zusätzlich sorgt dieses Schlüsselwort dafür, dass Mutt den Rumpf der Nachricht nicht über die Standardeingabe weiterleitet.

%t

Mutt ersetzt %t durch die Text-Darstellung des Content-Typs der Nachricht. Sie hat dieselbe Form wie der erste Parameter der Mailcap-Definitions-Zeile (z. B. text/html oder image/gif).

%{<Parameter>}

Mutt ersetzt dies durch den Wert des angegebenen Parameters aus dem “Content-Type:”-Header der Nachricht. Nehmen wir an, die Nachricht enthält folgendes:

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

Mutt ersetzt in diesem Fall `{charset}` durch `iso-8859-1`. Die von `metamail` standardmäßig installierte Mailcap-Datei benutzt dieses Feature zum Test des Zeichensatzes.

`\%`

Dieses Schlüsselwort wird durch `%` ersetzt.

Mutt unterstützt die Schlüsselwörter `%F` und `%n` aus RFC 1524 gegenwärtig nicht. Der Hauptzweck dieser Parameter ist die Behandlung von mehrteiligen Nachrichten, die von Mutt intern verarbeitet werden.

5.3.4 Beispiele für mailcap-Dateien

Diese mailcap-Datei ist ziemlich einfach:

```
# Hier läuft immer X :)
video/*;          xanim %s > /dev/null
image/*;          xv %s > /dev/null

# Wenn Netscape ständig läuft
text/html;        netscape -remote 'openURL(%s)'
```

Diese mailcap-Datei zeigt eine ganze Menge Beispiele:

```
# xanim wird zur Anzeige von Videos benutzt. Die Statusmeldungen, die
# xanim zeigt, werden nach /dev/null umgeleitet.
video/*;          xanim %s > /dev/null

# Einen laufenden Netscape-Browser fernsteuern
text/html;        netscape -remote 'openURL(%s)'; test=RunningNetscape

# Wenn Netscape nicht läuft, aber X läuft, starte Netscape, um das
# Objekt anzuzeigen
text/html;        netscape %s; test=RunningX

# wenn X nicht läuft, benutze lynx
text/html;        lynx %s

# Diese Version wandelt text/html in text/plain um
text/html;        lynx -dump %s; copiousoutput

# enriched.sh konvertiert text/enriched in text/html und benutzt dann
# lynx -dump zum Konvertieren in text/plain
text/enriched;    enriched.sh ; copiousoutput

# benutze enscript, um den Text zweiseitig auszudrucken
text/*;          more %s; print=enscript -2Gr %s

# Netscape führt eine Markierung ein, um sich selbst mitzuteilen, dass es
# jpeg-Bilder intern anzeigen soll
image/jpeg;xv %s; x-mozilla-flags=internal
```

```
# Benutze xv für das Anzeigen von Bildern, wenn X läuft
# Zusätzlich wird der Backslash benutzt, um die Zeile fortzusetzen und
# einen Editor für image/* zu setzen
image/*;xv %s; test=RunningX; \
    edit=xpaint %s

# Benutze die netpbm-Tools, um Bilder in eine Textdarstellung
# umzuwandeln
image/*; (anytopnm %s | pnmscale -ysize 80 46 | ppmtopgm | pgmtopbm |
pbmtoascii -1x2 ) 2>&1 ; copiousoutput

# Schicke excel-Tabellen an meinen NT-Rechner
application/ms-excel; open.pl %s
```

5.4 MIME-Autoview

Zusätzlich zur Möglichkeit, Mutt explizit zu sagen, dass es einen Anhang mit dem in der Mailcap-Datei definierten Anzeige-Programm darstellen soll, unterstützt Mutt die automatische Anzeige von MIME-Anhängen im Pager.

Dazu muss ein Anzeigeprogramm in der Mailcap-Datei angegeben werden, das mit der Option `copiousoutput` versehen ist, damit es als nicht-interaktiv erkannt werden kann. Üblicherweise wird der Eintrag für die Umwandlung in eine Textdarstellung verwendet, die dann im Pager angezeigt werden kann.

Jetzt kann das `auto_view`-Kommando in der `muttrc` verwendet werden, das die Content-Typen angibt, die automatisch angezeigt werden sollen.

Zum Beispiel könnte man `auto_view` so setzen:

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

Mutt könnte dann diese Mailcap-Einträge benutzen, um automatisch Anhänge dieser Typen anzuzeigen:

```
text/html; lynx -dump %s; copiousoutput; nametemplate=%s.html
text/enriched; enrich.sh ; copiousoutput
image/*; anytopnm %s | pnmscale -xsize 80 -ysize 50 | ppmtopgm | pgmtopbm | pbmtoascii ; \
copiousoutput
application/x-gunzip; gzcatt; copiousoutput
application/x-tar-gz; gunzip -c %s | tar -tf - ; copiousoutput
application/postscript; ps2ascii %s; copiousoutput
```

5.5 MIME Multipart/Alternative

Mutt versucht selbst zu entscheiden, welcher Teil eines multipart/alternative-Typs angezeigt werden soll. Zuerst überprüft Mutt die `alternative_order`-Liste, um festzustellen, ob einer der verfügbaren Typen bevorzugt wird. Die `alternative_order`-Liste enthält eine Reihe von MIME-Typen in der gewünschten Reihenfolge. Die üblichen MIME-Platzhalter können verwendet werden. Beispiel:

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

Als nächstes überprüft Mutt, ob für einen dieser Typen 5.4 (`auto_view`) definiert ist. Ist das nicht der Fall, sucht Mutt nach einem Text-Typ. Zuletzt sucht Mutt nach irgendeinem Typ, mit dem es umgehen kann.

6 Referenz

6.1 Optionen an der Kommandozeile

Mutt versucht die spool-Mailbox zu lesen, wenn es ohne Argumente aufgerufen wird. Es ist allerdings auch möglich, andere Mailboxen zu öffnen oder Nachrichten von der Kommandozeile zu verschicken.

-a	Datei an eine Nachricht anhängen
-b	Eine Bcc-Adresse ("Blinde" Kopie an) angeben
-c	Eine Cc-Adresse (Kopie an) angeben
-e	Ein Konfigurations-Kommando angeben, das ausgeführt wird, sobald die Initialisierungsdateien gelesen wurden
-f	Eine Mailbox, aus der gelesen werden soll
-F	Eine alternative Datei angeben, aus der Initialisierungskommandos gelesen werden
-h	Hilfe zu den Kommandozeilen-Optionen
-H	Angabe einer Entwurfs-Datei, aus der Body und Header gelesen werden.
-i	Datei, die in eine neue Nachricht eingefügt wird.
-m	den Standard-Mailbox-Typ festlegen
-n	die System-Muttrc ignorieren
-p	Eine zurückgestellte Nachricht bearbeiten
-R	Mailbox darf nur gelesen, aber nicht geschrieben werden
-s	Angabe eines Betreffs (In Anführungszeichen setzen, wenn es Leerzeichen enthält)
-v	Versionsnummer und Einstellungen bei der Kompilierung anzeigen
-x	Simulation des mailx(1)-Erstellungs-Modus
-y	Anzeige eines Menüs mit allen bekannten Mailboxen
-z	Sofort beenden, wenn es keine neuen Nachrichten gibt
-Z	Die erste Datei mit neuen Nachrichten öffnen. Wenn es keine neuen Nachrichten gibt, sofort beenden.

Um Nachrichten in einer Mailbox zu lesen:

```
mutt [-nz] [-F muttrc] [-m Typ] [-f Mailbox]
```

Um eine neue Nachricht zu erstellen:

```
mutt [-n] [-F muttrc] [-a Datei] [-c Adresse] [-i Dateiname] [-s Betreff] Adresse [ Adresse ... ]
```

Mutt unterstützt auch einen "batch-Mode", um vorbereitete Nachrichten zu verschicken. Dazu muss die gewünschte Datei einfach über stdin an Mutt weitergeleitet werden. Beispiel:

```
mutt -s "Datensatz für den zweiten Programmlauf" professor@grosse-uni.de < ~/lauf2.dat
```

Dieses Kommando schickt eine Nachricht mit dem Betreff "Datensatz für den zweiten Programmlauf" an "professor@grosse-uni.de". Der Body der Nachricht enthält den Inhalt der Datei "~/lauf2.dat".

6.2 Konfigurations-Kommandos

Mutt versteht die folgenden Kommandos:

- 4.12 (account-hook) *Muster Kommando* 3.2 (alias) *Kurzbezeichnung Adresse* [, Adresse, ...]
- 3.2 (unalias) [* | Taste ...]
- 5.5 (alternative_order) *MIME-Typ* [*MIME-Typ* ...]

- 5.4 (auto_view) *MIME-Typ* [*MIME-Typ ...*]
- 3.3 (bind) *Menü Taste Funktion*
- 3.4 (charset-hook) *Alias Zeichensatz*
- 3.7 (color) *Objekt Vordergrund Hintergrund* [*regulärer_Ausdruck*]
- 3.7 (uncolor) *Index Muster* [*Muster ...*]
- 3.21 (exec) *Funktion* [*Funktion ...*]
- 3.15 (fcc-hook) *Mutter Mailbox*
- 3.16 (fcc-save-hook) *Muster Mailbox*
- 3.5 (folder-hook) *Muster Kommando*
- 3.13 (hdr_order) *Header-Feld* [*Header-Feld ...*]
- 3.13 (unhdr_order) *Header-Feld* [*Header-Feld ...*]
- 3.4 (iconv-hook) *Zeichensatz lokaler_Zeichensatz*
- 3.8 (ignore) *Muster* [*Muster ...*]
- 3.8 (unignore) *Muster* [*Muster ...*]
- 3.9 (lists) *Adresse* [*Adresse ...*]
- 3.9 (unlists) *Adresse* [*Adresse ...*]
- 3.6 (macro) *Menü Taste Tastenfolge* [*Beschreibung*]
- 3.11 (mailboxes) *Dateiname* [*Dateiname ...*]
- 3.10 (mbox-hook) *Muster Mailbox*
- 3.18 (message-hook) *Muster Kommando*
- 3.7 (mono) *Objekt-Attribut* [*regulärer_Ausdruck*]
- 3.7 (unmono) *Index Muster* [*Muster ...*]
- 3.12 (my_hdr) *String*
- 3.12 (unmy_hdr) *Header-Feld* [*Header-Feld ...*]
- 3.19 (pgp-hook) *Muster Schlüssel-ID*
- 3.20 (push) *String*
- 3.23 (reset) *Variable* [*Variable ...*]
- 3.14 (save-hook) *regulärer_Ausdruck Dateiname*
- 3.22 (score) *Muster Wert*
- 3.22 (unscore) *Muster* [*Muster ...*]
- 3.17 (send-hook) *regulärer_Ausdruck Kommando*
- 3.23 (set) [no|inv] *Variable[=Wert]* [*Variable ...*]

- 3.23 (unset) *Variable* [*Variable ...*]
- 3.24 (source) *Dateiname*
- 3.9 (subscribe) *Adresse* [*Adresse ...*]
- 3.9 (unsubscribe) *Adresse* [*Adresse ...*]
- 3.23 (toggle) *Variable* [*Variable ...*]
- 3.25 (unhook) *Hook-Type*

6.3 Konfigurations-Variablen

6.3.1 abort_nosubject

Typ: Quad-Option

Voreinstellung: ask-yes (Ja mit Nachfrage)

Auf *yes* gesetzt bewirkt diese Variable, dass das Erstellen einer Nachricht abgebrochen wird, wenn kein Subject (Betreff) angegeben wurde. Auf *no* gesetzt werden Nachrichten ohne angegebenes Subject niemals abgebrochen.

6.3.2 abort_unmodified

Typ: Quad-Option

Voreinstellung: yes (gesetzt)

Wenn diese Variable auf *yes* gesetzt ist, wird die Erstellung einer Nachricht sofort abgebrochen, wenn der Body der Nachricht nicht verändert wurde (dies wird nur nach dem *ersten* Bearbeiten der Datei überprüft). Auf *no* gesetzt, wird die Erstellung nicht abgebrochen.

6.3.3 alias_file

Typ: Pfadname

Voreinstellung: "~/muttrc"

Die Standarddatei, in der Adressbucheinträge abgespeichert werden, die von der Funktion 2.3.4 (create-alias) angelegt werden.

Hinweis: Mutt wird diese Datei nicht automatisch einlesen; Du musst explizit den Befehl 3.24 (source) dafür verwenden.

6.3.4 alias_format

Typ: Zeichenkette

Voreinstellung: "%4n %t %-10a %r"

Diese Variable definiert das Format, in dem die Daten im Adressbuch angezeigt werden. Die folgenden printf(3)-ähnlichen Platzhalter sind verfügbar:

%a

Kurzname

%f

Markierung - z.Zt. nur ein "d" für einen zum Löschen markierten Alias

%n

Indexnummer

%r

Adresse, die dem Kurznamen zugeordnet ist

%t

Zeichen, das anzeigt, ob der Alias für die Benutzung markiert wurde

6.3.5 allow_8bit

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Kontrolliert, ob 8-bit-Zeichen beim Versenden von Mail mit Hilfe der Quoted-Printable- oder Base64-Kodierung zu 7-bit-Zeichen umgewandelt werden.

6.3.6 allow_ansi

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Kontrolliert, ob ANSI-Farbcodes in Nachrichten (und Farbmarkierungen in Richtext-Nachrichten) ausgewertet werden sollen. Nachrichten, die diese Codes enthalten, sind selten, aber falls diese Option gesetzt ist, werden diese Texte entsprechend eingefärbt. Beachte, dass dies die eigene Farbauswahl überschreiben und sogar ein Sicherheitsproblem darstellen kann, weil eine Nachricht eine Zeile wie "[– PGP Ausgabe folgt ...]" beinhalten und ihr dieselbe Farbe wie deine Farbwahl für Anhänge geben kann.

6.3.7 alternates

Typ: Regulärer Ausdruck

Voreinstellung: ""

Ein regulärer Ausdruck, der es erlaubt, *alternative* E-Mail-Adressen anzugeben, unter denen du erreichbar bist. Dies hilft Mutt bei der Frage, welche Nachrichten von dir oder an dich geschrieben wurden.

6.3.8 arrow_cursor

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, wird ein Pfeil ("->") zur Anzeige des aktuellen Eintrags benutzt, anstatt die ganze Zeile hervorzuheben. Bei langsamen Netzwerk- oder Modemverbindungen erhöht sich dadurch die Bediengeschwindigkeit, weil weniger auf dem Bildschirm neu aufgebaut werden muss, wenn man zur nächsten oder vorigen Nachricht wechselt.

6.3.9 `ascii_chars`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, benutzt Mutt einfache ASCII-Zeichen anstelle der üblichen *ACS*-Zeichen, um Threads und Attachment-Bäume darzustellen.

6.3.10 `askbcc`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, wird Mutt nach möglichen Blind-Carbon-Copy-Empfängern (Bcc) fragen, bevor die zu verschickende Nachricht editiert wird.

6.3.11 `askcc`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, wird Mutt nach möglichen Carbon-Copy-Empfängern (Cc) fragen, bevor die zu verschickende Nachricht editiert wird.

6.3.12 `attach_format`

Typ: Zeichenkette

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

Diese Variable beschreibt das Format des 'Anhänge'-Menüs. Die folgenden printf-ähnlichen Platzhalter werden akzeptiert:

%D

Anhang ist zum Löschen markiert

%d

Beschreibung des Anhangs

%e

MIME content-transfer-encoding

%f

Dateiname

%I

Platzierung (I=inline, A=attachment)

%m

Haupt-MIME-Typ

%M

MIME-Subtyp

%n

Nummer des Anhangs

%s

Größe

%t

Anhang ist markiert

%u

Anzeige der Löschmarkierung

%>X

macht den Rest der Zeichenkette rechtsbündig und füllt mit dem Zeichen "X" auf

%|X

füllt bis zum Ende der Zeile mit dem Zeichen "X" auf

6.3.13 `attach_sep`

Typ: Zeichenkette

Voreinstellung: "\n"

Dies bestimmt den Trenner, der zwischen Anhängen eingefügt wird, wenn Anweisungen (Speichern, Drucken, Weiterleiten (Pipen), etc.) auf eine Liste von markierten Attachments angewendet werden.

6.3.14 `attach_split`

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Wenn diese Variable nicht gesetzt ist, wird Mutt bei Anweisungen (Speichern, Drucken, Pipen, etc) auf eine Liste von markierten Anhängen die Attachments verbinden und sie als einen einzigen Anhang behandeln. Der in 6.3.13 (`$attach_sep`) festgelegte Trenner wird nach jedem Anhang eingefügt. Falls die Variable gesetzt ist, wird Mutt jeden Anhang einzeln behandeln.

6.3.15 `attribution`

Typ: Zeichenkette

Voreinstellung: "On %d, %n wrote:"

Diese Zeichenkette wird einer Nachricht, die in einer Antwort eingefügt wurde, vorangestellt. Eine vollständige Liste der möglichen printf()-artigen findet man im Abschnitt 6.3.83 (`$index_format`).

6.3.16 `autoedit`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable zusammen mit 6.3.40 (`$edit_headers`) gesetzt ist, wird Mutt Eingangsfragen des Sendemenüs überspringen und eine direkte Bearbeitung deiner Nachricht erlauben. Auf das Sendemenü kann noch zugegriffen werden, sobald die Bearbeitung der Nachricht abgeschlossen ist.

Siehe auch "6.3.45 (`$fast_reply`)".

6.3.17 auto_tag

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Falls diese Variable gesetzt ist, werden Funktionen im *index*-Menü, die eine Nachricht beeinflussen, auf alle markierten Nachrichten (sofern vorhanden) angewendet. Falls die Variable nicht gesetzt ist, muss zuerst die Tag-Prefix-Funktion (Voreinstellung: ";") verwendet werden, damit die nächste Funktion auf alle markierten Nachrichten angewendet wird.

6.3.18 beep

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Falls diese Variable gesetzt ist, wird Mutt piepsen, wenn ein Fehler auftritt.

6.3.19 beep_new

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Falls diese Variable gesetzt ist, wird Mutt jedesmal beim Eintreffen einer neuen Nachricht piepsen. Dies ist unabhängig vom Wert der Variable 6.3.18 (beep).

6.3.20 bounce_delivered

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Falls diese Variable gesetzt ist, wird Mutt eine Delivered-To Header-Zeile einfügen, wenn eine Nachricht "gebounced" wird. Postfix-Benutzer werden es hilfreich finden, die Variable auf "no" zu setzen.

6.3.21 charset

Typ: Zeichenkette

Voreinstellung: ""

Die Variable beschreibt den Zeichensatz, den dein Terminal zum Anzeigen und für die Eingabe von Text benutzt.

6.3.22 check_new

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Anmerkung: Diese Option wirkt nur bei Mailboxen des Typs *Maildir* und *MH*.

Falls diese Variable *gesetzt* ist, wird Mutt kontrollieren, ob neue Nachrichten eingetroffen sind, während die Mailbox geöffnet ist. Vor allem bei Mailboxen des Typs *MH* kann dieser Vorgang recht lange dauern, weil bei jeder Datei in dem Verzeichnis überprüft wird, ob Mutt sie schon betrachtet hatte. Falls *check_new ungesetzt* ist, wird keine Überprüfung nach neuen Nachrichten durchgeführt, solange die Mailbox geöffnet ist.

6.3.23 collapse_unread

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Falls diese Variable *ungesetzt* ist, wird Mutt einen Thread nicht einfalten, falls er ungelesene Nachrichten beinhaltet.

6.3.24 uncollapse_jump

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable *gesetzt* ist, springt Mutt zur nächsten ungelesenen Nachricht (falls vorhanden), wenn der aktuelle Thread *aufgefaltet* wird.

6.3.25 compose_format

Typ: Zeichenkette

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

Diese Variable kontrolliert das Aussehen der Statuszeile, die im Compose-Menü angezeigt wird. Diese Zeichenkette ähnelt 6.3.215 (`$status_format`), hat aber ihre eigene Auswahl an printf-ähnlichen Platzhaltern:

`%a`

Gesamtanzahl der Anhänge

`%h`

Lokaler Rechnername

`%l`

Geschätzte Größe (in Bytes) der aktuellen Nachricht

`%v`

Version von Mutt

Siehe auch die Beschreibung der Optionen zu 6.3.215 (`$status_format`) für weitere Informationen, wie 6.3.25 (`$compose_format`) gesetzt wird.

6.3.26 confirmappend

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Falls diese Variable *gesetzt* ist, wird Mutt um eine Bestätigung bitten, wenn es Nachrichten an eine bestehende Mailbox anhängen soll.

6.3.27 confirmcreate

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Falls diese Variable *gesetzt* ist, wird Mutt um eine Bestätigung bitten, wenn es beim Abspeichern von Nachrichten eine neue Mailbox erzeugen soll.

6.3.28 connect_timeout

Typ: Zahl

Voreinstellung: 30

Diese Variable veranlasst Mutt, eine Netzverbindung (für IMAP oder POP) nach so vielen Sekunden abzubrechen, falls die Verbindung nicht aufgebaut werden kann. Ein negativer Wert lässt Mutt unbegrenzt auf die Verbindung warten.

6.3.29 copy

Typ: Quad-Option

Voreinstellung: yes (gesetzt)

Diese Variable kontrolliert, ob Kopien von Deinen versandten Nachrichten archiviert werden sollen oder nicht. Siehe auch 6.3.181 (record), 6.3.192 (save_name), 6.3.51 (force_name) und 3.15 (fcc-hook).

6.3.30 date_format

Typ: Zeichenkette

Voreinstellung: "%!a, %b %d, %Y at %I:%M:%S%p %Z"

Diese Variable bestimmt das Format des Datums, das von dem Platzhalter "%d" in 6.3.83 (\$index_format) ausgegeben wird. Es wird an den Aufruf *strftime* weitergereicht, um das Datum zu bearbeiten. Siehe auch die Manpage von *strftime(3)* für die korrekte Syntax.

Wenn das erste Zeichen in der Zeichenkette kein Ausrufezeichen ("!") ist, werden der Monat und die Wochentage in der Sprache ausgegeben, die in der Variablen 6.3.86 (locale) angegeben ist. Falls das erste Zeichen in der Zeichenkette ein Ausrufezeichen ist, wird es daraus entfernt, und die Namen der Monate und Wochentage in der restlichen Zeichenkette so ausgegeben, als wäre die Sprache *C* (US-Englisch) gesetzt.

6.3.31 default_hook

Typ: Zeichenkette

Voreinstellung: " f %s !~P | (~P ~C %s)"

Diese Variable kontrolliert, wie send-hooks, save-hooks and fcc-hooks interpretiert werden, wenn in ihnen ein einfacher regulärer Ausdruck anstelle eines Suchmusters angegeben wird. Die Hooks werden bei ihrer Deklaration ausgewertet, deshalb wird für einen Hook diejenige Einstellung von *default_hook* verwendet, die bei seiner Deklaration gültig ist. Mit der Voreinstellung wird getestet, ob die Nachricht entweder von jemanden ist, auf den der gegebene reguläre Ausdruck zutrifft, oder (falls sie von dir selbst ist, d.h. die From-Adresse auf den Wert von 6.3.7 (alternates) zutrifft), ob sie an jemanden geht, der in den Header-Zeilen "To" oder "Cc" steht und auf den der gegebene reguläre Ausdruck zutrifft.

6.3.32 delete

Typ: Quad-Option

Voreinstellung: ask-yes (Ja mit Nachfrage)

Diese Variable bestimmt, ob Nachrichten wirklich gelöscht werden, wenn eine Mailbox geschlossen oder synchronisiert wird. Falls sie auf *yes* gesetzt ist, werden zum Löschen markierte Nachrichten automatisch und ohne Nachfrage entfernt. Falls sie auf *no* gesetzt ist, werden zum Löschen markierte Nachrichten in der Mailbox belassen.

6.3.33 delete_untag

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Wenn diese Option gesetzt ist, entfernt Mutt die Markierung von Nachrichten, wenn sie zum Löschen vorge-merkt werden. Dies findet sowohl Anwendung, wenn man eine Nachricht ausdrücklich löscht, als auch, wenn sie in eine andere Mailbox verschoben wird.

6.3.34 digest_collapse

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Falls diese Option *gesetzt* ist, zeigt Mutts Anhang-Menü die Unterteile von einzelnen Nachrichten in einem Digest nicht an. Um sie zu sehen, drücke "v" in diesem Menü.

6.3.35 display_filter

Typ: Pfadname

Default: ""

Diese Variable bestimmt ein Programm zum Filtern von Nachrichten. Wenn eine Nachricht angezeigt wird, wird sie auf Standardausgabe an 6.3.35 (`$display_filter`) übergeben, und die gefilterte Nachricht wird von der Standardeingabe gelesen.

6.3.36 dotlock_program

Typ: Pfadname

Voreinstellung: `"/usr/local/bin/mutt_dotlock"`

Enthält den Pfadnamen der Binärdatei `mutt_dotlock` (8), die von Mutt verwendet werden soll.

6.3.37 dsn_notify

Typ: Zeichenkette

Voreinstellung: ""

Anmerkung: Du solltest dies nur benutzen, wenn du Sendmail 8.8.x oder neuer verwendest.

Diese Variable bestimmt, für welche Situationen eine Benachrichtigung angefordert werden soll. Die Zeichenkette besteht aus einer durch Kommata getrennten Liste (keine Leerzeichen!) aus einem oder mehreren der folgenden Ausdrücke: *never*, um nie eine Benachrichtigung anzufordern, *failure*, um eine Benachrichtigung bei Übertragungsfehlern anzufordern, *delay*, um eine Benachrichtigung bei Verzögerungen anzufordern, und *success*, um eine Benachrichtigung anzufordern, wenn die Zustellung erfolgreich war.

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

6.3.38 dsn_return

Typ: Zeichenkette

Voreinstellung: ""

Anmerkung: Du solltest dies nur benutzen, wenn du Sendmail 8.8.x oder neuer verwendest.

Diese Variable kontrolliert, wie viel von deiner Nachricht mit der DSN-Nachricht zurückgegeben wird. Sie kann entweder auf *hdrs* gesetzt werden, um nur die Kopfzeilen der Nachricht zurückzubekommen, oder auf *full*, um die komplette Nachricht zurückzubekommen.

Beispiel: `set dsn_return=hdrs`

6.3.39 duplicate_threads

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Diese Variable kontrolliert, ob Mutt, falls nach Threads sortiert wird, Nachrichten mit der selben Message-ID zusammen im Thread anzeigt. Mutt wird in einem solchen Fall durch ein Gleichheitszeichen im Threaddiagramm anzeigen, dass es diese Nachrichten für Duplikate hält.

6.3.40 edit_headers

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Diese Option erlaubt es, die Header-Zeilen der zu verschickenden Nachricht zusammen mit dem Rumpf zu bearbeiten.

6.3.41 editor

Typ: Pfadname

Voreinstellung: ""

Diese Variable gibt an, welchen Editor Mutt verwenden soll. Die Voreinstellung ist (so vorhanden) der Wert der Umgebungsvariablen EDITOR oder VISUAL, oder die Zeichenkette "vi".

6.3.42 encode_from

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, verwendet Mutt die Quoted-Printable-Kodierung für Nachrichten, die am Anfang einer Zeile die Zeichenkette "From " enthalten. Das ist nützlich, um zu verhindern, dass gewisse Mail-Transport- und Auslieferungs-Systeme sich an der Nachricht zu schaffen machen.

6.3.43 envelope_from

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn man diese Variable setzt, versucht Mutt, den *envelope*-Absender von Nachrichten aus dem "From:"-Header herzuleiten. Du solltest wissen, dass diese Information mittels der Kommandozeilen-Option "-f" an Sendmail übergeben wird; setze diese Variable also nicht, wenn du diese Sendmail-Option selbst verwenden möchtest oder wenn das Sendmail auf deiner Maschine diese Option nicht unterstützt.

6.3.44 `escape`

Typ: Zeichenkette

Voreinstellung: "~"

Das Escape-Zeichen, mit dem die Funktionen des eingebauten Editors angesprochen werden können.

6.3.45 `fast_reply`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Falls diese Variable gesetzt ist, verzichtet Mutt darauf, beim Antworten auf Nachrichten erst nach Empfängern und Betreff zu fragen und beim Weiterleiten (forward) von Nachrichten die Betreff-Zeile bestätigen zu lassen.

Anmerkung: Diese Variable hat keine Auswirkung, wenn die Variable 6.3.16 (`$autoedit`) gesetzt ist.

6.3.46 `fcc_attach`

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Diese Variable kontrolliert, ob Anhänge von ausgehenden Nachrichten zusammen mit dem Hauptteil deiner Nachricht abgespeichert werden sollen oder nicht.

6.3.47 `fcc_clear`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable *gesetzt* ist, werden FCCs (lokale Kopien) unverschlüsselt und unsigniert abgespeichert, selbst wenn die eigentliche Nachricht verschlüsselt und / oder signiert worden ist.

6.3.48 `folder`

Typ: Pfadname

Voreinstellung: "~/Mail"

Diese Variable bestimmt das Verzeichnis, in dem deine Mailboxen liegen. Ein '+' oder '=' am Anfang des Pfadnamens wird durch den Wert dieser Variable ersetzt. Beachte, dass du, falls du für diese Variable einen anderen Wert als die Voreinstellungen verwendest, sicherstellen musst, dass *folder* gesetzt wird, *bevor* du '+' oder '=' für andere Variablen benutzt, weil die Ersetzung bereits im "set"-Kommando erfolgt.

6.3.49 `folder_format`

Typ: Zeichenkette

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

Diese Variable erlaubt es dir, die Anzeige des Datei-Browsers deinem persönlichen Geschmack anzupassen. Diese Zeichenkette ist dem 6.3.83 (`$index_format`) ähnlich, hat aber ihre eigenen printf()-ähnlichen Platzhalter:

%C

Fortlaufende Nummerierung der Dateien

%d

Datum/Zeit der letzten Änderung der Mailbox

%f

Dateiname

%F

Dateizugriffsrechte

%g

Gruppenname (oder der numerische Wert der GID, falls der Name fehlt)

%l

Anzahl der Hardlinks

%N

N, falls die Mailbox neue Nachrichten hat, ansonsten nichts

%s

Größe in Bytes

%t

enthält * wenn die Datei markiert ist, ansonsten nichts

%u

Besitzername (oder der numerische Wert der UID, falls der Name fehlt)

%>X

macht den Rest der Zeichenkette rechtsbündig und füllt mit dem Zeichen "X" auf

%|X

füllt bis zum Ende der Zeile mit dem Zeichen "X" auf

6.3.50 `followup_to`

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Diese Variable kontrolliert, ob die Header-Zeile *Mail-Followup-To* beim Senden von Mail erstellt werden soll oder nicht. Falls sie *gesetzt* ist, wird Mutt diese Zeile erstellen, wenn du auf eine bekannte Mailingliste ("subscribe" oder 3.9 (lists)) antwortest.

Das Feld hat zwei Absichten. Erstens soll es verhindern, dass du doppelte Kopien von Antworten auf Nachrichten erhältst, die an Mailinglisten gehen. Zweitens soll es garantieren, dass du eine getrennte Antwort für alle Nachrichten erhältst, die an bekannte Mailinglisten gehen, auf denen du aber nicht eingeschrieben bist. Die Headerzeile wird bei eingeschriebenen Listen nur die Adresse der Liste enthalten; bei nicht abonnierten Mailinglisten dagegen sowohl die Adresse der Liste als auch deine eigene. Ohne diese Zeile wird eine Gruppen-Antwort auf deine Nachricht, die an eine abonnierte Liste ging, sowohl an die Liste als auch an deine Adresse geschickt, was zu zwei Kopien der selben Mail für dich führt.

6.3.51 force_name

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Diese Option ist der Variablen 6.3.192 (`$save_name`) ähnlich. Allerdings speichert Mutt bei Verwendung dieser Variablen auch dann eine Kopie der ausgehenden Nachrichten in einer Mailbox mit dem Benutzernamen der Empfängeradresse, wenn diese Mailbox noch nicht existiert.

Siehe auch die Variable 6.3.181 (`$record`).

6.3.52 forward_decode

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Diese Variable kontrolliert das Dekodieren umfangreicher MIME-Nachrichten nach `text/plain`, wenn eine Nachricht weitergeleitet wird. Die Kopfzeilen (Header) werden dann auch nach RFC2047 dekodiert. Diese Variable wird nur benutzt, wenn 6.3.103 (`mime_forward`) *ungesetzt* ist, sonst wird statt dessen 6.3.104 (`mime_forward_decode`) verwendet.

6.3.53 forward_format

Typ: Zeichenkette

Voreinstellung: "[%a: %s]"

Diese Variable bestimmt, nach welchem Muster beim Weiterleiten einer Nachricht eine Betreff-Zeile generiert wird. Sie bietet die gleichen Format-Platzhalter wie die Variable 6.3.83 (`$index_format`).

6.3.54 forward_quote

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Falls diese Variable *gesetzt* ist, werden weitergeleitete Nachrichten, die im Rumpf (Body) der Nachricht eingefügt werden (falls 6.3.103 (`mime_forward`) *ungesetzt* ist), unter Benutzung von 6.3.82 (`indent_string`) gequotet.

6.3.55 from

Typ: E-Mail-Adresse

Voreinstellung: ""

Wenn diese Variable *gesetzt* ist, sollte sie die Adresse enthalten, die du normalerweise im "From:"-Header verwendest. Sie wird überschrieben, wenn mittels `my_hdr` (auch von `send-hooks` aus) oder "reverse_name" eine andere Adresse *gesetzt* wird.

Voreinstellung ist der Inhalt der Umgebungsvariable `EMAIL`.

6.3.56 gecos_mask

Typ: Regulärer Ausdruck

Voreinstellung: "[^,]*"

Dieser reguläre Ausdruck wird von Mutt verwendet, um das GECOS-Feld eines Passwort-Eintrages einzulesen, wenn es einen Aliasnamen expandiert. In der Voreinstellung wird "`^[^,]*`" verwendet, das die Zeichenkette bis zu dem ersten "," zurückgibt. Wenn das GECOS-Feld Einträge der Form "Nachname, Vorname" enthält, solltest du `gecos_mask` auf `".*"` setzen.

Das kann nützlich sein, wenn du folgendes Problem beobachtest: Du schickst eine E-Mail an den User `stevf`, dessen voller Name "Steve Franklin" lautet. Wenn Mutt `stevf` durch "Franklin" `<stevf@foo.bar>` ersetzt, dann solltest du `gecos_mask` auf einen regulären Ausdruck setzen, der den ganzen Namen erfasst, damit Mutt nicht "Franklin", sondern "Franklin, Steve" verwendet.

6.3.57 `hdrs`

Typ: Wahrheitswert

Voreinstellung: `yes` (gesetzt)

Falls die Variable nicht gesetzt ist, werden Header-Zeilen, die normalerweise durch den Befehl 3.12 (`my_hdr`) hinzugefügt würden, nicht erstellt. Diese Variable *muß* auf `no` gesetzt werden, bevor eine neue Nachricht erstellt oder auf eine Nachricht geantwortet wird, um Wirkung zu zeigen. Falls sie gesetzt ist, werden die benutzerdefinierten Kopfzeilen zu jeder neuen Nachricht hinzugefügt.

6.3.58 `header`

Typ: Wahrheitswert

Voreinstellung: `no` (ungesetzt)

Falls diese Variable gesetzt ist, veranlasst sie Mutt dazu, die Kopfzeilen (Header) der Nachricht, auf die du antwortest, mit in den Editor zu laden. Die Variable 6.3.233 (`weed`) findet Anwendung.

6.3.59 `help`

Typ: Wahrheitswert

Voreinstellung: `yes` (gesetzt)

Falls diese Variable gesetzt ist, wird in der obersten Zeile des Bildschirms eine Kurzhilfe angezeigt, die die Belegung für die wichtigsten Funktionen der einzelnen Menüs beschreibt.

Anmerkung: Die Belegung wird nicht richtig angezeigt, wenn die Funktion an eine Tastensequenz statt an eine einzelne Taste gebunden ist. Es kann auch vorkommen, dass die Hilfezeile nicht aktualisiert wird, wenn eine Belegung geändert wird, während Mutt läuft. Da diese Variable sich hauptsächlich an neue Benutzer richtet, dürfte keine dieser Einschränkungen ein größeres Problem darstellen.

6.3.60 `hidden_host`

Typ: Wahrheitswert

Voreinstellung: `no` (ungesetzt)

Falls diese Variable gesetzt ist, wird Mutt den Hostname-Teil von 6.3.67 (`hostname`) auslassen, wenn es den Domain-Teil von unvollständigen Adressen ergänzt. Diese Variable beeinflusst nicht die Erzeugung von Message-IDs, noch führt sie zum Abschneiden von First-Level-Domains.

6.3.61 `hide_limited`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Das Setzen dieser Variable veranlasst Mutt, im Thread-Baum nicht auf das Fehlen von Nachrichten hinzuweisen.

6.3.62 `hide_missing`

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Wenn diese Variable gesetzt ist, zeigt Mutt im Thread-Baum nicht die Abwesenheit von Nachrichten an, die aufgrund einer Begrenzung der angezeigten Nachrichten (mittels des Befehls "limit") verborgen sind.

6.3.63 `hide_top_limited`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, weist Mutt im Thread-Baum nicht auf fehlende Nachrichten am Anfang eines Threads hin. Beachte, dass diese Option keine Auswirkung hat, wenn 6.3.61 (`$hide_limited`) gesetzt ist.

6.3.64 `hide_top_missing`

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Wenn diese Variable gesetzt ist, weist Mutt im Thread-Baum nicht darauf hin, wenn am Anfang eines Threads aufgrund einer Begrenzung der angezeigten Nachrichten (Befehl "limit") Mails fehlen. Beachte, dass diese Option keine Auswirkung hat, wenn 6.3.62 (`$hide_missing`) gesetzt ist.

6.3.65 `history`

Typ: Zahl

Voreinstellung: 10

Diese Variable bestimmt die Größe des History-Puffers für Eingaben - angegeben ist die Anzahl der Zeichenketten, die Mutt sich "merkt". Der Puffer wird jedes Mal gelöscht, wenn die Variable gesetzt wird.

6.3.66 `honor_followup_to`

Typ: Quad-Option

Voreinstellung: yes (gesetzt)

Diese Variable setzt fest, ob eine Mail-Followup-To-Header-Zeile bei Gruppen-Antworten berücksichtigt wird.

6.3.67 `hostname`

Typ: Zeichenkette

Voreinstellung: ""

Diese Variable bestimmt den Hostnamen, der in lokalen E-Mail-Adressen nach dem "@" benutzt werden soll. Dies überschreibt die incompilierte Definition, die aus `/etc/resolv.conf` gewonnen wurde.

6.3.68 ignore_list_reply_to

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Diese Variable beeinflusst das Verhalten der Funktion *Reply*, wenn auf Nachrichten von Mailinglisten geantwortet wird. Falls sie gesetzt ist und im Feld "Reply-To:" die gleiche Adresse wie im Feld "To:" eingetragen ist, nimmt Mutt an, dass das "Reply-To:" von der Mailingliste gesetzt wurde, um automatisch Antworten an die Liste zu senden, und ignoriert diese Zeile. Um eine Antwort an die Mailinglist zu richten, wenn diese Option gesetzt ist, steht die Funktion *list-reply* zur Verfügung; *group-reply* schickt die Antwort sowohl an den Autor als auch an die Liste.

6.3.69 imap_authenticators

Typ: Zeichenkette

Voreinstellung: ""

Dies ist eine durch Strichpunkte getrennte Liste von Authentifizierungsmethoden, die Mutt zum Einloggen auf einem IMAP-Server anzuwenden versucht, in der Reihenfolge, wie Mutt sie ausprobieren soll. Authentifizierungsmethoden sind entweder "login" oder die rechte Seite einer IMAP "AUTH=xxx" Fähigkeitsanzeige, z.B. "digeset-md5", "gssapi" oder "cram-md5". Dieser Parameter unterscheidet nicht zwischen Groß- und Kleinschreibung. Falls diese Variable ungesetzt ist (Voreinstellung), wird Mutt alle möglichen verfügbaren Methoden durchprobieren, angefangen von der sichersten bis zur unsichersten.

Beispiel: `set imap_authenticators="gssapi:cram-md5:login"`

Anmerkung Mutt wird nur eine andere Authentifizierungsmethode versuchen, falls die vorhergehende nicht verfügbar ist. Falls eine Methode verfügbar ist, aber die Authentifizierung fehlschlägt, wird Mutt sich nicht mit dem IMAP-Server verbinden.

6.3.70 imap_delim_chars

Typ: Zeichenkette

Voreinstellung: "/."

Diese Variable enthält eine Liste von Zeichen, die du zum Trennen von Foldern bei der Anzeige von IMAP-Pfaden verwenden möchtest. Insbesondere hilft sie bei der Benutzung der "="-Abkürzung für deine *Folder*-Variable.

6.3.71 imap_force_ssl

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, wird Mutt immer SSL/TLS für die Verbindung zum IMAP-Server benutzen.

6.3.72 imap_home_namespace

Typ: Zeichenkette

Voreinstellung: ""

Du möchtest normalerweise deine persönlichen Folder zusammen mit deiner INBOX im IMAP-Browser sehen. Falls du etwas anderes siehst, kannst du diese Variable auf den IMAP-Pfad zu deinen Foldern setzen.

6.3.73 `imap_keepalive`

Typ: Zahl

Voreinstellung: 900

Diese Variable bestimmt die maximale Zeit in Sekunden, die Mutt abwarten wird, bevor er offene IMAP-Verbindungen pollt, um zu verhindern, dass der Server die Verbindungen schließt, bevor Mutt fertig ist. Die Voreinstellung liegt mitten in der durch RFCs spezifizierten Zeitspanne von 30 Minuten, bevor ein Server eine Verbindung schließen darf, aber praktisch wird hier und da gegen den RFC verstoßen. Reduziere diese Zahl, falls du von deinem IMAP-Server wegen Inaktivität rausgeworfen wirst.

6.3.74 `imap_list_subscribed`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Diese Variable bestimmt, ob beim Durchschauen der IMAP-Mailboxen nur nach eingetragenen Mailboxen oder nach allen Mailboxen gesucht wird. Dies lässt sich im IMAP-Browser mit dem Befehl *toggle-subscribed* ändern.

6.3.75 `imap_pass`

Typ: Zeichenkette

Voreinstellung: ""

Diese Zeichenkette beinhaltet das Passwort deines IMAP-Accounts. Falls die Variable ungesetzt ist, wird Mutt dich nach einem Passwort fragen, wenn du die Funktion `fetch-mail` aufrufst. **Warnung:** Du solltest diese Option nur verwenden, wenn du an einem sehr sicheren Computer arbeitest, weil der Systemverwalter deine `muttrc` lesen kann, auch wenn du von den Zugriffsrechten her der einzige bist, der die Datei lesen könnte.

6.3.76 `imap_passive`

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Falls diese Variable gesetzt ist, wird Mutt keine neuen IMAP Verbindungen öffnen, um nach neuen Nachrichten zu schauen, sondern nur über existierende IMAP Verbindungen auf neue Nachrichten prüfen. Dies ist sinnvoll, falls du beim Starten von Mutt nicht nach Benutzername und Passwort gefragt werden willst oder falls der Verbindungsaufbau langsam ist.

6.3.77 `imap_peek`

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Falls diese Variable gesetzt ist, vermeidet Mutt das Markieren deiner Mail als gelesen, wann immer du eine Nachricht vom Server holst. Dies ist normalerweise eine gute Sache, aber kann das Schließen von IMAP-Folders langsamer machen. Diese Option existiert, um Geschwindigkeits-Freaks zu beruhigen.

6.3.78 `imap_servernoise`

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Diese Variable veranlasst Mutt, Warnmeldungen des IMAP-Servers als Fehlermeldungen anzuzeigen. Weil diese Meldungen oft harmlos sind oder durch Konfigurationsprobleme auf dem Server verursacht werden, die außerhalb des Einflusses des Benutzers liegen, könnte der Wunsch entstehen, diese Meldungen zu unterdrücken.

6.3.79 `imap_user`

Typ: Zeichenkette

Voreinstellung: ""

Diese Variable enthält deinen Benutzernamen auf dem IMAP-Server.

Ihre Voreinstellung entspricht deinem Benutzernamen auf dem lokalen System.

6.3.80 `implicit_autoview`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable auf "yes" gesetzt ist, sucht Mutt bei *jedem* MIME-Anhang, für den keine interne Darstellung möglich ist, nach einem mailcap-Eintrag mit der Kennzeichnung `copiousoutput`. Falls ein solcher Eintrag gefunden wird, wird Mutt das dort definierte Programm benutzen, um den Rumpf der Nachricht in Textform zu konvertieren.

6.3.81 `include`

Typ: Quad-Option

Voreinstellung: ask-yes (Ja mit Nachfrage)

Diese Variable kontrolliert, ob eine Kopie der Nachricht(en), auf die du antwortest, deiner Antwort hinzugefügt wird oder nicht.

6.3.82 `indent_string`

Typ: Zeichenkette

Voreinstellung: "> "

Diese Variable bestimmt die Zeichenkette, die zitierten Zeilen aus der Nachricht, auf die du antwortest, vorangestellt wird. Es ist anzuraten, diese Zeichenkette nicht zu ändern, weil sich die fanatischeren unter den Netz-Bewohnern darüber aufregen.

6.3.83 `index_format`

Typ: Zeichenkette

Voreinstellung: "%4C %Z %b %d} %-15.15L (%4l) %s"

Diese Variable erlaubt es dir, die Anzeige des Nachrichtenindex deinem persönlichen Geschmack anzupassen.

"Formatierungs-Zeichenketten" ähneln den Zeichenketten, die in der C-Funktion `printf` benutzt werden, um die Ausgabe anzupassen (siehe deren Manpage für weitere Details). Die folgenden Platzhalter sind in Mutt definiert:

%a

Adresse des Autors

%b

Dateiname der Originalmailbox (denke an mailBox)

%B

Liste, an die der Brief gesendet wurde, oder sonst der Mailboxname (**%b**)

%c

Anzahl der Zeichen (Bytes) in der Nachricht

%C

aktuelle Nachrichtennummer

%d

Datum und Uhrzeit der Nachricht in dem vom "date_format" bestimmten Format, umgewandelt in die Zeitzone des Absenders

%D

Datum und Uhrzeit der Nachricht in dem vom "date_format" bestimmten Format, umgewandelt in die lokale Zeitzone

%f

vollständige From: Zeile (Adresse und Realname)

%F

Name des Autors oder des Empfängers, falls die Nachricht von dir ist

%i

Message-ID der aktuellen Nachricht

%l

Anzahl der Zeilen der Nachricht

%L

Wenn eine Adresse im To oder Cc Header-Feld einer Adresse entspricht, die der Benutzer mit dem Befehl "lists" angegeben hat, zeigt dieser Buchstabe "To <listen-name>" an, ansonsten verhält er sich wie **%F**.

%m

Gesamtanzahl der Nachrichten in der Mailbox

%M

Anzahl der versteckten Nachrichten, falls der Thread zusammengefaltet ist

%n

Vollständiger Name des Autors (oder die Adresse, falls er fehlt)

%O

(`._Originaler Save-Folder`) Wo Mutt eigentlich die Nachricht verstecken würde: List- oder Empfängername, falls es keine Liste ist

%s

Betreff der Nachricht

%S

Status der Nachricht (N/D/d/!/*/r)

%t

Header-Zeile 'To:' (Empfänger)

%T

das passende Zeichen aus der Zeichenkette 6.3.225 (`$to_chars`)

%u

Benutzername (Login) des Autors

%v

Vorname des Autors oder des Empfängers, falls die Nachricht von dir ist

%y

Feld "x-label:", falls vorhanden

%Y

Feld "x-label:", falls vorhanden, und (1) nicht Teil des Thread-Baumes, (2) an der Spitze des Threads oder (3) "x-label" unterscheidet sich vom "x-label" der Vorgängernachricht

%Z

Kennzeichnungen des Nachrichtenstatus

%{fmt}

Das Datum und die Uhrzeit der Nachricht werden in die Zeitzone des Absenders umgewandelt, und "fmt" wird von dem Systemaufruf "strftime" ausgewertet; ein führendes Ausrufezeichen schaltet die Locale-Einstellungen ab

%[fmt]

Das Datum und die Uhrzeit der Nachricht werden in die lokale Zeitzone umgewandelt, und "fmt" wird von dem Systemaufruf "strftime" ausgewertet; ein führendes Ausrufezeichen schaltet die Locale-Einstellungen ab

%(fmt)

lokales Datum und lokale Uhrzeit vom Empfangszeitpunkt der Nachricht. "fmt" wird von dem Systemaufruf "strftime" ausgewertet; ein führendes Ausrufezeichen schaltet die Locale-Einstellungen ab

%<fmt>

aktuelle lokale Uhrzeit. "fmt" wird von dem Systemaufruf "strftime" ausgewertet; ein führendes Ausrufezeichen schaltet die Locale-Einstellungen ab

%>X

macht den Rest der Zeichenkette rechtsbündig und füllt mit dem Zeichen "X" auf

%|X

füllt bis zum Ende der Zeile mit dem Zeichen "X" auf

Siehe auch: 6.3.225 (`$to_chars`).

6.3.84 `ispell`

Typ: Pfadname

Voreinstellung: `"/usr/bin/ispell"`

Diese Variable beschreibt, wie `ispell` (GNUs Rechtschreibüberprüfung) aufgerufen wird.

6.3.85 `keep_flagged`

Typ: Wahrheitswert

Voreinstellung: `no` (ungesetzt)

Falls diese Variable gesetzt ist, werden gelesene Nachrichten, die als wichtig markiert sind, nicht von der Spool-Mailbox in deine 6.3.94 (`$mbox`)-Mailbox verschoben. Ein Befehl vom Typ 6.3.94 (`$mbox`) hat ebenfalls keinen Einfluss.

6.3.86 `locale`

Typ: Zeichenkette

Voreinstellung: `"C"`

Die Variable beschreibt die Locale, die von *strftime(3)* zur Formatierung von Datum und Uhrzeit benutzt werden soll. Erlaubte Werte sind die Zeichenketten, die dein System für die locale-Variable `LC_TIME` akzeptiert.

6.3.87 `mail_check`

Typ: Zahl

Voreinstellung: `5`

Diese Variable bestimmt, wie oft (in Sekunden) Mutt auf neue Mail prüfen soll.

6.3.88 `mailcap_path`

Typ: Zeichenkette

Voreinstellung: `""`

Diese Variable bestimmt, in welchen Dateien nachgeschaut wird, wenn MIME-Nachrichten dargestellt werden sollen, die nicht direkt von Mutt unterstützt werden.

6.3.89 `mailcap_sanitize`

Typ: Wahrheitswert

Voreinstellung: `yes` (gesetzt)

Falls diese Variable gesetzt ist, wird Mutt die möglichen Zeichen in %-Auswertungen in der mailcap auf eine bekannte Auswahl von sicheren Zeichen beschränken. Dies ist eine sichere Einstellung, aber wir sind nicht ganz sicher, dass sie nicht die Ausführung einiger komplizierter MIME Regeln verhindert.

ÄNDERE DIESE EINSTELLUNG NICHT, WENN DU NICHT WIRKLICH SICHER WEISST, WAS DU TUST!

6.3.90 maildir_trash

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Das Setzen dieser Variable bewirkt das Abspeichern von zum Löschen markierten Nachrichten mit dem Flag (T)rashed. **Anmerkung:** Dies wirkt nur bei Mailboxen des Typs Maildir und hat keine Auswirkungen bei anderen Typen.

6.3.91 mark_old

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Diese Variable kontrolliert, ob Mutt einen Unterschied zwischen *neuen* Nachrichten und *alten ungelesenen* Nachrichten machen soll oder nicht. Standardmäßig markiert Mutt neue Nachrichten als alt, wenn du eine Mailbox verlässt, ohne sie gelesen zu haben. Das nächste Mal, wenn du Mutt startest, werden die Nachrichten im Index mit einem "O" angezeigt, das darauf hinweist, dass sie alt sind. Um Mutt dazu zu bringen, alle ungelesenen Nachrichten als neu zu betrachten, kannst du diese Variable auf "no" setzen.

6.3.92 markers

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Diese Variable kontrolliert die Anzeige von umgebrochenen Zeilen im internen Pager. Falls sie gesetzt ist, wird als Markierung am Anfang von umgebrochenen Zeilen ein "+" angezeigt. Siehe auch die Variable 6.3.205 (\$smart_wrap).

6.3.93 mask

Typ: Regulärer Ausdruck

Voreinstellung: "!^\.[^.]"

Dieser reguläre Ausdruck wird im Datei-Browser verwendet; auf Wunsch kann ihm der *Nicht*-Operator "!" vorangestellt werden. Nur Dateien, deren Namen auf diesen Ausdruck zutreffen, werden angezeigt. Groß- und Kleinschreibung wird unterschieden.

6.3.94 mbox

Typ: Pfadname

Voreinstellung: "/mbox"

Diese Variable bestimmt die Mailbox, in die gelesene Nachrichten aus deiner 6.3.213 (spool)-Mailbox verschoben werden.

6.3.95 mbox_type

Typ: Mailbox-Typ

Voreinstellung: mbox

Diese Variable bestimmt den Standard-Mailboxtyp, der beim Erstellen von neuen Mailboxen benutzt wird. Sie kann die Einstellungen mbox, MMDF, MH und Maildir haben.

6.3.96 metoo

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Falls diese Variable ungesetzt ist, wird Mutt deine Adresse von der Liste der Empfänger entfernen, wenn du auf eine Nachricht antwortest.

6.3.97 menu_scroll

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Falls diese Variable *gesetzt* ist, werden Menüs eine Zeile auf- oder abwärts gescrollt, wenn du versuchst, die Bildschirmgrenzen zu überschreiten. Falls sie *ungesetzt* ist, wird der Bildschirm gelöscht und die nächste oder vorige Seite des Menüs wird dargestellt (nützlich für langsame Verbindungen, weil seltener ein Neuaufbau des Bildschirms nötig wird).

6.3.98 meta_key

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Falls diese Variable *gesetzt* ist, wird Mutt gezwungen, Tastendrucke, bei denen das höchste Bit (Bit 8) *gesetzt* ist, so zu interpretieren, als hätte der Benutzer die ESC Taste gedrückt und danach das Zeichen, das sich nach dem Entfernen des höchsten Bits ergibt. Falls die gedrückte Taste zum Beispiel einen ASCII Wert von 0xf4 hat, wird dies so behandelt, als hätte der Benutzer ESC und dann "x" gedrückt, weil das Ergebnis beim Entfernen des höchsten Bit von "0xf4" "0x74" ist, was dem ASCII Zeichen "x" entspricht.

6.3.99 mh_purge

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Falls diese Variable ungesetzt ist, wird Mutt das Verhalten von mh nachahmen und gelöschte Nachrichten in mh-Mailboxen in `,<alter Dateiname>` umbenennen, anstelle sie wirklich zu löschen. Falls die Variable *gesetzt* ist, werden die Nachrichtendateien einfach gelöscht.

6.3.100 mh_seq_flagged

Typ: Zeichenkette

Voreinstellung: "flagged"

Diese Variable bestimmt den Namen der MH-Folge, der zum Markieren von wichtigen Nachrichten benutzt wird.

6.3.101 `mh_seq_replied`

Typ: Zeichenkette

Voreinstellung: "replied"

Diese Variable bestimmt den Namen der MH-Folge, der zum Markieren von beantworteten Nachrichten benutzt wird.

6.3.102 `mh_seq_unseen`

Typ: Zeichenkette

Voreinstellung: "unseen"

Diese Variable bestimmt den Namen der MH-Folge, mit der noch nicht betrachtete Nachrichten markiert werden.

6.3.103 `mime_forward`

Typ: Quad-Option

Voreinstellung: no (ungesetzt)

Falls diese Variable gesetzt ist, wird die Nachricht, die du weiterleitest, als ein getrennter MIME Teil angehängt, anstelle sie in den eigentlichen Rumpf der Nachricht einzufügen. Dies ist nützlich beim Weiterleiten von MIME-Nachrichten, weil der Empfänger die Nachricht genau so ansehen kann, wie sie bei dir angekommen ist. Falls du von Nachricht zu Nachricht zwischen MIME und nicht-MIME hin- und herschalten willst, setze diese Variable auf den Wert "ask-no" oder "ask-yes".

Siehe auch 6.3.52 (`forward_decode`) und 6.3.104 (`mime_forward_decode`).

6.3.104 `mime_forward_decode`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Diese Variable kontrolliert das Dekodieren von umfangreichen MIME-Nachrichten nach text/plain beim Weiterleiten einer Nachricht, wenn 6.3.103 (`mime_forward`) *gesetzt* ist. Sonst wird 6.3.52 (`forward_decode`) benutzt.

6.3.105 `mime_forward_rest`

Typ: Quad-Option

Voreinstellung: yes (gesetzt)

Wenn mehrere Anhänge einer MIME-Nachricht aus dem Anhänge-Menü heraus weitergeleitet werden sollen, werden Attachments, die nicht vernünftig dekodiert werden konnten, als Anhang der neu erstellten Nachricht verschickt, sofern diese Variable gesetzt ist.

6.3.106 `mix_entry_format`

Typ: Zeichenkette

Voreinstellung: "%4n %c %-16s %a"

Diese Variable beschreibt den Aufbau einer Remailer-Zeile in dem Menü zur Auswahl einer Mixmaster-Kette. Die folgenden printf-ähnlichen Platzhalter werden unterstützt:

%n

Die laufende Nummer im Menü

%c

Fähigkeiten des Remailers

%s

Der Kurzname des Remailers

%a

Die E-Mail-Adresse des Remailers

6.3.107 mixmaster

Typ: Pfadname

Voreinstellung: "mixmaster"

Diese Variable enthält den Pfad zur Mixmaster-Binärdatei auf deinem System. Diese wird mit diversen Parametern aufgerufen, um die Liste der bekannten Remailer zusammenzustellen und um schließlich eine Nachricht über die Mixmaster-Kette zu verschicken.

6.3.108 move

Typ: Quad-Option

Voreinstellung: ask-no (Nein mit Nachfrage)

Diese Variable kontrolliert, ob du nach einer Bestätigung gefragt wirst, wenn Mutt vorhat, gelesene Nachrichten von von deiner Spool-Mailbox in deine 6.3.94 (\$mbox) Mailbox zu verschieben, oder wenn er aufgrund des 3.10 (mbox-hook)-Befehls Nachrichten verschiebt.

6.3.109 message_format

Typ: Zeichenkette

Voreinstellung: "%s"

Dies ist die Zeichenkette, die im "Anhang"-Menü für die Anzeige von Attachments des Typs message/rfc822 benutzt wird. Der Abschnitt über 6.3.83 (index_format) enthält eine komplette Liste der erlaubten Platzhalter.

6.3.110 pager

Typ: Pfadname

Voreinstellung: "builtin"

Diese Variable bestimmt, welchen Pager du zum Anschauen von Nachrichten verwenden möchtest. builtin bewirkt, dass der in Mutt eingebaute Pager verwendet wird, sonst sollte diese Variable den Pfadnamen zu dem externen Pager, den du verwenden möchtest, beinhalten.

Das Benutzen eines externen Pagers kann einige Nachteile haben: Zusätzliche Tastendrucke sind notwendig, weil du direkt aus dem Pager keine Mutt-Funktionen aufrufen kannst, und Änderungen an der Bildschirmgröße haben zur Folge, dass im Hilfe-Menü Zeilen, die größer als die Bildschirmbreite sind, schlecht formatiert werden.

6.3.111 pager_context

Typ: Zahl

Voreinstellung: 0

Diese Variable bestimmt die Anzahl der Zeilen, die beim Anzeigen der nächsten oder vorigen Seite nochmals ausgegeben werden, um die Orientierung zu behalten. Standardmäßig zeigt Mutt die Zeile nach der letzten sichtbaren ganz oben auf der nächsten Seite an (0 Zeilen Kontext).

6.3.112 pager_format

Typ: Zeichenkette

Voreinstellung: "-%Z- %C/%m: %-20.20n %s"

Diese Variable kontrolliert das Aussehen der Status-Zeile, die sowohl im internen als auch (vor der Nachricht) in einem externen Pager angezeigt wird. Die gültigen Platzhalter sind im Abschnitt 6.3.83 (index_format) aufgelistet.

6.3.113 pager_index_lines

Typ: Zahl

Voreinstellung: 0

Diese Variable bestimmt die Anzahl der Zeilen, die für einen Mini-Index zur Verfügung gestellt werden, wenn man sich im eingebauten Pager befindet. Die aktuelle Nachricht wird, falls sie nicht in der Nähe des Anfangs oder Endes der Mailbox ist, ungefähr nach einem Drittel dieses Mini-Index aufgeführt, um dem Leser einen Überblick über einige Nachrichten vor und nach der angezeigten zu geben. Dies ist z.B. nützlich um festzustellen, wie viele Nachrichten im aktuellen Thread noch übrig sind. Eine Zeile ist für die Statusanzeige des Index reserviert, sodass bei einem Wert von 6 für *pager_index_lines* nur in 5 Zeilen tatsächlich der Index angezeigt wird. Bei einem Wert von 0 wird kein Index gezeigt. Falls die Anzahl der Nachrichten in der aktuellen Mailbox geringer ist als der Wert von *pager_index_lines*, wird der Index nur so viele Zeilen wie nötig benutzen.

6.3.114 pager_stop

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, wird der interne Pager **nicht** zur nächsten Nachricht springen, wenn du am Ende einer Nachricht bist und die Funktion *next-page* (nächste Seite) aufrufst.

6.3.115 pgp_autosign

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Das Setzen dieser Variable wird Mutt veranlassen, immer einen Versuch zu starten, ausgehende Nachrichten mit PGP/MIME zu signieren. Dieses Verhalten kann unter Benutzung des *pgp-menu* geändert werden, falls eine Signierung einmal nicht nötig ist oder zusätzlich verschlüsselt werden soll.

6.3.116 pgp_autoencrypt

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Das Setzen dieser Variable wird Mutt veranlassen, immer einen Versuch zu starten, ausgehende Nachrichten mit PGP/MIME zu verschlüsseln. Dies ist wahrscheinlich nur in Verbindung mit dem Befehl *send-hook* nützlich. Dieses Verhalten kann unter Benutzung des *pgp-menu* geändert werden, falls Verschlüsselung einmal nicht nötig ist oder zusätzlich signiert werden soll.

6.3.117 `pgp_ignore_subkeys`

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Das Setzen dieser Variable lässt Mutt OpenPGP-Unterschlüssel ignorieren. Stattdessen erbt der Hauptschlüssel alle Fähigkeiten der Unterschlüssel. Setze diese Variable auf 'no', falls du interessante Schlüsselauswahlspiele spielen möchtest.

6.3.118 `pgp_entry_format`

Typ: Zeichenkette

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

Diese Variable erlaubt es dir, das Menü für die PGP-Schlüsselauswahl deinem persönlichen Geschmack anzupassen. Diese Zeichenkette ähnelt dem 6.3.83 (`$index_format`), hat aber ihre eigene Auswahl an printf()-ähnlichen Platzhaltern:

%n

Nummer

%k

Schlüssel-ID

%u

Benutzer-ID

%a

Algorithmus

%l

Schlüssellänge

%f

Markierungen

%c

Fähigkeiten

%t

Vertrauenswürdigkeit/Gültigkeit der Zuordnung von Schlüssel zu Benutzer-ID

%[<s>]

Datum des Schlüssels, wobei <s> ein strftime(3)-Ausdruck ist

6.3.119 `pgp_good_sign`

Typ: Regulärer Ausdruck

Voreinstellung: ""

Falls du dieser Variable einen regulären Ausdruck zuweist, wird eine PGP-Signatur nur dann als gültig angesehen, falls die Ausgabe von 6.3.134 (`$pgp_verify_command`) diesen Text beinhaltet. Benutze diese Variable, falls der Exit-Code dieses Befehls auch bei ungültigen Signaturen 0 ist.

6.3.120 `pgp_long_ids`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Falls diese Variable gesetzt ist, werden 64 bit PGP-Schlüssel-IDs benutzt. Ist sie ungesetzt, werden die normalen 32 bit Schlüssel-IDs verwendet.

6.3.121 `pgp_replyencrypt`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Falls diese Variable gesetzt ist, werden Antworten auf verschlüsselte Nachrichten automatisch verschlüsselt.

6.3.122 `pgp_replysign`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Falls diese Variable gesetzt ist, werden Antworten auf signierte Nachrichten automatisch signiert.

Anmerkung: Dies funktioniert nicht bei Nachrichten, die verschlüsselt **und** signiert sind!

6.3.123 `pgp_replysignencrypted`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Falls diese Variable gesetzt ist, werden Antworten auf verschlüsselte Nachrichten automatisch signiert. Dies macht in Kombination mit 6.3.121 (`$pgp_replyencrypt`) Sinn, weil es dir erlaubt, alle Nachrichten signieren zu lassen, die automatisch verschlüsselt werden. Damit kann man das in 6.3.122 (`$pgp_replysign`) angesprochene Problem umgehen, dass Mutt nicht in der Lage ist herauszufinden, ob eine verschlüsselte Nachricht auch signiert ist.

6.3.124 `pgp_retainable_sigs`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, besteht der Body (Rumpf) signierter und verschlüsselter Nachrichten aus verschachtelten multipart/signed und multipart/encrypted Teilen.

Das ist für Anwendungen wie verschlüsselte und signierte Mailing-Listen nützlich, wo die äußere Schicht (multipart/encrypted) leicht entfernt werden kann, während der innere multipart/signed-Teil erhalten bleibt.

6.3.125 pgp_show_unusable

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Falls diese Variable gesetzt ist, wird Mutt auch unbrauchbare Schlüssel im PGP-Schlüsselauswahlmenü anzeigen. Dies beinhaltet Schlüssel, die zurückgerufen wurden, abgelaufen oder vom Benutzer gesperrt worden sind.

6.3.126 pgp_sign_as

Typ: Zeichenkette

Voreinstellung: ""

Falls du mehr als ein Schlüsselpaar hast, erlaubt dir diese Option, den privaten Schlüssel zu bestimmen, den du benutzen willst. Es wird empfohlen, dass du die Schlüssel-ID zur Angabe deines Schlüssels benutzt (z.B. "0x00112233").

6.3.127 pgp_strict_enc

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Falls diese Variable gesetzt ist, wird Mutt PGP/MIME signierte Nachrichten automatisch in *quoted-printable* kodieren. Beachte bitte, dass das Ändern dieser Variable zu Problemen mit nicht überprüfbaren PGP Signaturen führen kann. Ändere dies also nur, wenn du weißt, was du tust.

6.3.128 pgp_timeout

Typ: Zahl

Voreinstellung: 300

Diese Variable gibt die Zeit in Sekunden an, nach der ein PGP-Mantra aus dem Speicher gelöscht wird, wenn es nicht mehr benutzt wurde.

6.3.129 pgp_verify_sig

Typ: Quad-Option

Voreinstellung: yes (gesetzt)

Wenn diese Variable auf "yes" gesetzt ist, versucht Mutt immer, PGP/MIME-Signaturen zu überprüfen. Wenn sie auf "ask-yes" bzw. "ask-no" (Ja bzw. Nein mit Nachfrage) gesetzt ist, fragt Mutt nach, ob es eine Signatur überprüfen soll oder nicht. Wenn sie auf "no" steht, versucht Mutt nie, PGP/MIME-Signaturen zu überprüfen.

6.3.130 pgp_sort_keys

Typ: Sortier-Reihenfolge

Voreinstellung: address

Gibt an, wie die Einträge im "pgp Schlüssel"-Menü sortiert werden. Folgende Werte sind gültig:

address

sortiert alphabetisch nach Benutzer-ID

keyid

sortiert alphabetisch nach Schlüssel-ID

date

sortiert nach Erstellungsdatum des Schlüssels

trust

sortiert nach Vertrauenswürdigkeit des Schlüssels

Wenn du eine umgekehrte Sortierreihenfolge bevorzugst, kannst du zusätzlich `reverse-` vor die Werte stellen.

6.3.131 `pgp_create_traditional`

Typ: Quad-Option

Voreinstellung: `no` (ungesetzt)

Diese Option bestimmt, ob Mutt unter bestimmten Umständen nach der "altmodischen Art" PGP-verschlüsselte oder -signierte Nachrichten erzeugt.

Man sollte dabei wissen, dass PGP/MIME automatisch verwendet wird, wenn die Nachricht einen anderen Zeichensatz als US-ASCII hat oder aus mehr als einem einzigen MIME-Teil besteht.

Man sollte ebenfalls wissen, dass **stark** davon **abgeraten** wird, nach der alten Methode PGP in Mails zu verwenden.

6.3.132 `pgp_decode_command`

Typ: Zeichenkette

Voreinstellung: `""`

Diese Zeichenkette gibt einen Befehl an, der verwendet wird, um `application/pgp` Anhänge zu dekodieren.

In den "PGP command"-Variablen kann eine eigene Auswahl printf-ähnlicher Platzhalter verwendet werden:

%p

Wird durch `PGPPASSFD=0` ersetzt, wenn ein Mantra erforderlich ist, sonst zu einer leeren Zeichenkette. Anmerkung: Kann mit einem `%-`-Konstrukt verwendet werden.

%f

Wird durch den Namen der Datei ersetzt, die die Nachricht enthält.

%s

Wird durch den Namen der Datei ersetzt, die den Signatur-Teil eines multipart/signed-Anhangs enthält, der überprüft werden soll.

%a

Der Wert von 6.3.126 (`$pgp_sign_as`)

%r

Eine oder mehrere Schlüssel-IDs

Beispiele, wie diese Variablen für die verschiedenen gebräuchlichen PGP-Versionen gesetzt werden können, sind in den `pgp*.rc` und `gpg.rc` Dateien im Unterverzeichnis `samples/` enthalten, das zusammen mit der Dokumentation auf deinem System installiert worden ist.

6.3.133 `pgp_getkeys_command`

Typ: Zeichenkette

Voreinstellung: ""

Dieser Befehl wird aufgerufen, wenn Mutt Informationen über öffentliche Schlüssel benötigt. `%r` ist der einzige Platzhalter, der in dieser Variablen Verwendung findet.

6.3.134 `pgp_verify_command`

Typ: Zeichenkette

Voreinstellung: ""

Dieser Befehl wird verwendet, um PGP/MIME-Signaturen zu überprüfen.

6.3.135 `pgp_decrypt_command`

Typ: Zeichenkette

Voreinstellung: ""

Dieser Befehl wird gebraucht, um mit PGP/MIME verschlüsselte Nachrichten zu dekodieren.

6.3.136 `pgp_clearsign_command`

Typ: Zeichenkette

Voreinstellung: ""

Dieser Befehl wird verwendet, um ein PGP-Anhang mit einer "Klartext-Signatur" nach der alten Methode zu erzeugen. Es wird **stark** davon **abgeraten**, dieses Format heute noch zu verwenden.

6.3.137 `pgp_sign_command`

Typ: Zeichenkette

Voreinstellung: ""

Dieser Befehl wird verwendet, um die separate PGP-Signatur eines multipart/signed PGP/MIME Body-Teils zu erzeugen.

6.3.138 `pgp_encrypt_sign_command`

Typ: Zeichenkette

Voreinstellung: ""

Dieser Befehl wird verwendet, um einen Teil des Nachrichtenrumpfs (Body) gleichzeitig zu signieren und zu verschlüsseln.

6.3.139 `pgp_encrypt_only_command`

Typ: Zeichenkette

Voreinstellung: ""

Dieser Befehl wird verwendet, um einen Teil des Nachrichtenrumpfs (Bodys) zu verschlüsseln, ohne ihn zu signieren.

6.3.140 `pgp_import_command`

Typ: Zeichenkette

Voreinstellung: ""

Dieser Befehl wird verwendet, um einen Schlüssel aus einer Nachricht in den öffentlichen Schlüsselbund des Benutzers zu importieren.

6.3.141 `pgp_export_command`

Typ: Zeichenkette

Voreinstellung: ""

Dieser Befehl wird verwendet, um einen öffentlichen Schlüssel aus dem Schlüsselbund des Benutzers zu exportieren.

6.3.142 `pgp_verify_key_command`

Typ: Zeichenkette

Voreinstellung: ""

Dieser Befehl wird verwendet, um Schlüssel-Informationen aus dem Menü zur Schlüssel-Auswahl zu überprüfen.

6.3.143 `pgp_list_secring_command`

Typ: Zeichenkette

Voreinstellung: ""

Dieser Befehl wird verwendet, um den Inhalt des geheimen Schlüsselbundes auszugeben. Das Ausgabe-Format muss das gleiche sein wie das von `gpg -list-keys -with-colons` erzeugte.

Dieses Format wird auch vom Hilfsprogramm `pgpring` erzeugt, das mit `Mutt` ausgeliefert wird.

6.3.144 `pgp_list_pubring_command`

Typ: Zeichenkette

Voreinstellung: ""

Dieser Befehl wird verwendet, um den Inhalt des öffentlichen Schlüsselbundes auszugeben. Das Ausgabe-Format muss das gleiche sein wie das von `gpg -list-keys -with-colons` erzeugte.

Dieses Format wird auch vom Hilfsprogramm `pgpring` erzeugt, das mit `Mutt` ausgeliefert wird.

6.3.145 forward_decrypt

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Diese Variable kontrolliert den Umgang mit verschlüsselten Nachrichten beim Weiterleiten einer Nachricht. Falls sie gesetzt ist, wird die äußere Schicht der Verschlüsselung entfernt. Diese Variable wird nur benutzt, wenn 6.3.103 (mime_forward) *gesetzt* und 6.3.104 (mime_forward_decode) *ungesetzt* ist.

6.3.146 ssl_starttls

Typ: Quad-Option

Voreinstellung: yes (gesetzt)

Falls diese Variable gesetzt ist (Voreinstellung), wird Mutt versuchen, STARTTLS auf Servern zu benutzen, die diese Option anbieten. Falls diese Variable ungesetzt ist, wird Mutt nie versuchen, STARTTLS zu benutzen, egal ob der Server es anbietet oder nicht.

6.3.147 certificate_file

Typ: Pfadname

Voreinstellung: ""

Diese Variable gibt eine Datei an, in der Zertifikate abgelegt werden, denen du traust. Wenn Mutt auf ein unbekanntes Zertifikat trifft, wirst du gefragt, ob du es akzeptierst oder nicht. Wenn du es akzeptierst, kann das Zertifikat ebenfalls in dieser Datei abgelegt werden und weitere Verbindungen werden automatisch akzeptiert.

Du kannst auch von Hand CA-Zertifikate in diese Datei hinzufügen. Jedes Server-Zertifikat, das mit einem dieser CA-Zertifikate unterschrieben ist, wird automatisch akzeptiert.

Beispiel: set certificate_file=~/.mutt/certificates

6.3.148 ssl_usesystemcerts

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Falls diese Variable auf *yes* gesetzt ist, verwendet Mutt CA-Zertifikate im systemweiten "Zertifikatsverzeichnis" (*certificate store*), um zu kontrollieren, ob ein Serverzertifikat von einer vertrauenswürdigen CA unterschrieben wurde.

6.3.149 entropy_file

Typ: Pfadname

Voreinstellung: ""

Diese Variable weist auf die Datei, die Zufallsdaten enthält, mit der die SSL-Bibliotheksfunktionen initialisiert werden.

6.3.150 ssl_use_sslv2

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Diese Variable gibt an, ob im SSL-Anmelde-Vorgang versucht werden soll, SSLv2 zu verwenden.

6.3.151 `ssl_use_sslv3`

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Diese Variable gibt an, ob im SSL-Anmelde-Vorgang versucht werden soll, SSLv3 zu verwenden.

6.3.152 `ssl_use_tlsv1`

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Diese Variable gibt an, ob im SSL-Anmelde-Vorgang versucht werden soll, TLSv1 zu verwenden.

6.3.153 `pipe_split`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wird in Verbindung mit dem Befehl *pipe-message* und dem "tag-prefix"-Operator wirksam. Wenn Mutt eine Liste von markierten Nachrichten weiterleitet (piped) und diese Variable ungesetzt ist, verknüpft Mutt die einzelnen Mails und piped sie als einen einzigen Ordner. Wenn sie gesetzt ist, werden die Mails einzeln gepiped. In beiden Fällen leitet Mutt die Nachrichten in der Reihenfolge weiter, in der sie im Moment sortiert sind, und fügt nach jeder Mail das in 6.3.155 (`$pipe_sep`) definierte Trennzeichen an.

6.3.154 `pipe_decode`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wird in Verbindung mit dem *pipe-message*-Befehl wirksam. Solange diese Variable ungesetzt ist, wird Mutt die Nachrichten unverändert weiterleiten (pipen). Wenn sie gesetzt ist, entfernt Mutt zuvor bestimmte Header und versucht, die Mails nach PGP/MIME zu entschlüsseln.

6.3.155 `pipe_sep`

Typ: Zeichenkette

Voreinstellung: "\n"

Das Trennzeichen, das zwischen den einzelnen Mails eingefügt wird, wenn Mutt eine Liste von markierten Nachrichten an einen externen Unix-Befehl weiterleitet (piped).

6.3.156 `pop_authenticators`

Typ: Zeichenkette

Voreinstellung: ""

Dies ist eine durch Strichpunkte getrennte Liste von Authentifizierungsmethoden, die Mutt zum Einloggen auf einem POP-Server anzuwenden versucht, in der Reihenfolge, wie Mutt sie ausprobieren soll. Authentifizierungsmethoden sind entweder "user", "apop" oder ein SASL-Mechanismus, z.B. "digest-md5", "gssapi" oder "cram-md5". Dieser Parameter unterscheidet nicht zwischen Groß- und Kleinschreibung. Falls diese

Variable ungesetzt ist (Voreinstellung), wird Mutt alle möglichen verfügbaren Methoden durchprobieren, angefangen von der sichersten bis zur unsichersten.

Beispiel: `set pop_authenticators="digest-md5:apop:user"`

6.3.157 `pop_auth_try_all`

Typ: Wahrheitswert

Voreinstellung: `yes` (gesetzt)

Falls diese Variable gesetzt ist, wird Mutt alle verfügbaren Authentifizierungsmethoden versuchen. Falls sie ungesetzt ist, wird Mutt nur eine andere Authentifizierungsmethode versuchen, wenn die vorhergehende nicht verfügbar ist. Falls eine Methode verfügbar ist, aber die Authentifizierung fehlschlägt, wird Mutt sich nicht mit dem POP-Server verbinden.

6.3.158 `pop_checkinterval`

Typ: Zahl

Voreinstellung: `60`

Diese Variable bestimmt, wie oft (in Sekunden) Mutt über POP nach neuer Mail schauen soll.

6.3.159 `pop_delete`

Typ: Quad-Option

Voreinstellung: `ask-no` (Nein mit Nachfrage)

Wenn diese Variable gesetzt ist, wird Mutt erfolgreich von einem POP-Server geladene Nachrichten auf dem Server löschen, wenn er die Funktion `fetch-mail` verwendet. Wenn sie nicht gesetzt ist, wird Mutt die Nachrichten herunterladen, sie jedoch auch auf dem POP-Server belassen.

6.3.160 `pop_host`

Typ: Zeichenkette

Voreinstellung: `" "`

Der Name deines POP-Servers für die Funktion `fetch-mail`. Du kannst auch einen anderen Port, einen Usernamen und ein Passwort eingeben, z.B.:

```
[pop[s]://][username[:password]@]popserver[:port]
```

6.3.161 `pop_last`

Typ: Wahrheitswert

Voreinstellung: `no` (ungesetzt)

Wenn diese Variable gesetzt ist, versucht Mutt, mit Hilfe des POP-Befehls `"LAST"` nur ungelesene Nachrichten vom POP-Server abzuholen, wenn die Funktion `fetch-mail` verwendet wird.

6.3.162 `pop_reconnect`

Typ: Quad-Option

Voreinstellung: `ask-yes` (Ja mit Nachfrage)

Diese Variable bestimmt, ob Mutt versuchen soll, sich nach dem Verlust einer Verbindung erneut mit dem POP-Server zu verbinden.

6.3.163 pop_user

Typ: Zeichenkette

Voreinstellung: ""

Dein Login-Name auf dem POP3-Server.

Die Voreinstellung ist dein Login-Name auf dem lokalen System.

6.3.164 pop_pass

Typ: Zeichenkette

Voreinstellung: ""

Enthält das Passwort Deines POP-Kontos. Falls die Variable ungesetzt ist, wird Mutt dich nach deinem Passwort fragen, wenn du die POP-Mailbox öffnen willst. **Warnung:** Du solltest diese Option nur verwenden, wenn du dich auf einem ausreichend sicheren Rechner befindest, weil der Superuser deine muttrc auch dann lesen kann, wenn du eigentlich der einzige bist, der sie lesen kann.

6.3.165 post_indent_string

Typ: Zeichenkette

Voreinstellung: ""

Gleicht der Variablen 6.3.15 (\$attribution). Mutt hängt diese Zeichenkette an die zitierten Nachrichten an, auf die man antwortet.

6.3.166 postpone

Typ: Quad-Option

Voreinstellung: ask-yes (Ja mit Nachfrage)

Bestimmt, ob Nachrichten in der 6.3.167 (\$postponed)-Mailbox gespeichert werden oder nicht, wenn man sich entscheidet, sie nicht unmittelbar zu versenden.

6.3.167 postponed

Typ: Pfadname

Voreinstellung: "~/postponed"

Mutt erlaubt es, das Versenden der Nachricht, die man gerade editiert, für unbestimmte Zeit 2.6 (zurückzustellen.) Wenn man sich dafür entscheidet, eine Nachricht zurückzustellen, speichert Mutt sie in dem Ordner ab, den diese Variable angibt. Siehe auch die Variable 6.3.166 (\$postpone).

6.3.168 preconnect

Typ: Zeichenkette

Voreinstellung: ""

Diese Variable beinhaltet einen Befehl, der ausgeführt wird, falls Mutt keine Verbindung zum Server aufbauen kann. Dies ist nützlich, um eine sichere Verbindung aufzubauen, z.B. mit `ssh(1)`. Falls der Befehl einen Wert ungleich 0 zurückgibt, gibt Mutt den Verbindungsaufbau auf. Beispiel:

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

Mailbox 'foo' auf mailhost.net kann jetzt über '{localhost:1234}foo' erreicht werden.

Anmerkung: Damit dieses Beispiel funktioniert, musst du in der Lage sein, dich ohne Eingabe des Passwortes auf dem entfernten Rechner einzuloggen.

6.3.169 `print`

Typ: Quad-Option

Voreinstellung: ask-no (Nein mit Nachfrage)

Bestimmt, ob Mutt vor dem Drucken eine Bestätigung verlangt oder nicht. Diese Variable ist für Leute (wie mich) nützlich, die oft versehentlich "p" drücken.

6.3.170 `print_command`

Typ: Pfadname

Voreinstellung: "lpr"

Diese Variable enthält den Befehl, an den Nachrichten weitergeleitet (gepiped) werden sollen, um sie auszu-drucken.

6.3.171 `print_decode`

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Diese Variable wird in Verbindung mit dem Befehl `print-message` benutzt. Falls sie gesetzt ist, wird die Nachricht dekodiert, bevor sie an das externe Programm, das durch 6.3.170 (`$print_command`) bestimmt wird, weitergeleitet wird. Falls die Option nicht gesetzt ist, findet keine Bearbeitung der Nachricht vor dem Drucken statt. Letzteres kann dann nützlich sein, falls du einen guten Druckerfilter hast, der in der Lage ist, E-Mails ordentlich für den Ausdruck zu formatieren.

6.3.172 `print_split`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Diese Variable wird in Verbindung mit dem Befehl `print-message` benutzt. Falls sie gesetzt ist, wird der Befehl, der durch 6.3.170 (`$print_command`) bestimmt wird, einmal für jede zu druckende Nachricht aufgerufen. Falls sie ungesetzt ist, wird der Befehl nur einmal aufgerufen, und alle Nachrichten werden zu einer verbunden mit einem Formfeed als Nachrichtentrenner.

Diejenigen, die den Mail-Druck-Modus des Programms `enscript(1)` verwenden, möchten diese Option sicher setzen.

6.3.173 `prompt_after`

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Wenn diese Variable gesetzt ist und du einen *externen* 6.3.110 (Pager) verwendest, fragt Mutt dich nach einem Befehl, wenn du den Pager verlässt. Wenn sie nicht gesetzt ist, kehrt Mutt statt dessen direkt zum Index-Menü zurück.

6.3.174 `query_command`

Typ: Pfadname

Voreinstellung: ""

Diese Variable gibt den Befehl an, den Mutt für externe Adressenabfragen (query) benutzen soll. Der Wert sollte ein %s enthalten; dieser Platzhalter wird dann durch die Zeichenkette ersetzt, die der Benutzer bei der Abfrage eingibt. Siehe "4.5 (Abfrage (query))", um mehr Information darüber zu erhalten.

6.3.175 `quit`

Typ: Quad-Option

Voreinstellung: yes (gesetzt)

Diese Variable bestimmt, ob "quit" und "exit" Mutt tatsächlich beenden. Wenn sie auf "yes" gestellt ist, beenden sie Mutt, wenn sie auf "no" gestellt ist, sind sie wirkungslos und wenn sie auf "ask-yes" (Ja mit Nachfrage) oder "ask-no" (Nein mit Nachfrage) gestellt ist, wirst du um eine Bestätigung gebeten, wenn du Mutt verlassen willst.

6.3.176 `quote_regexp`

Typ: regulärer Ausdruck

Voreinstellung: "`^[\t]*[>:}#]+`"

Ein regulärer Ausdruck, der im internen Pager verwendet wird, um zitierte Abschnitte des Texts im Rumpf (Body) der Nachrichten zu erkennen.

Hinweis: Um die *quotedx*-Muster im internen Pager nutzen zu können, muss in dieser Variable ein regulärer Ausdruck stehen, auf den die Zitatzeichen am Anfang einer zitierten Zeile *genau* passen.

6.3.177 `read_inc`

Typ: Zahl

Voreinstellung: 10

Wenn diese Variable auf einen Wert größer 0 gesetzt ist, zeigt Mutt an, bei welcher Nachricht es gerade ist, wenn es eine Mailbox-Datei einliest. Diese Anzeige erfolgt, nachdem *read_inc* Nachrichten gelesen wurden (wenn die Variable z.B. auf 25 gesetzt ist, wird Mutt dem Benutzer mitteilen, wenn er Nachricht 25 einliest und das nächste Mal wenn er Nachricht 50 erreicht). Dieses Verhalten soll den Fortschritt anzeigen, wenn große Mailbox-Dateien gelesen werden, die einige Zeit benötigen.

Wenn sie auf 0 gesetzt ist, wird der Benutzer nur eine einziges Mal informiert, nämlich bevor Mutt die Mailbox-Datei liest.

Siehe auch die Variable 6.3.236 (`$write_inc`).

6.3.178 read_only

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, werden alle Ordner nur zum Lesezugriff geöffnet.

6.3.179 realname

Typ: Zeichenkette

Voreinstellung: ""

Diese Variable gibt an, welcher "echte" oder "persönliche" Name ("Realname") beim Versenden von Nachrichten benutzt werden soll.

In der Voreinstellung wird das Feld für "Benutzernamen oder Kommentare" in `/etc/passwd` genutzt. Diese Variable wird *nicht* verwendet, wenn der Benutzer bereits in der Variablen 6.3.55 (from) einen Namen gesetzt hat.

6.3.180 recall

Typ: Quad-Option

Voreinstellung: ask-yes (Ja mit Nachfrage)

Bestimmt, ob man beim Erstellen einer neuen Nachricht automatisch eine zurückgestellte Nachricht weiterbearbeiten will oder nicht. Siehe dazu auch 6.3.167 (postponed).

Diese Variable auf "yes" (Ja) zu setzen, ist in der Regel wenig sinnvoll und wird daher nicht empfohlen.

6.3.181 record

Typ: Pfadname

Voreinstellung: ""

Diese Variable gibt die Datei an, in der deine ausgehenden Nachrichten abgelegt werden sollen. (Dies ist die bevorzugte Methode, um eine Kopie deiner Nachrichten abzuspeichern; eine andere Möglichkeit wäre es, den 3.12 (my_hdr)-Befehl zu verwenden, um ein `Bcc:`-Feld zu erzeugen, das deine E-Mail-Adresse enthält.)

Der Wert von 6.3.181 (`$record`) wird von den Variablen 6.3.51 (`$force_name`) und 6.3.192 (`$save_name`), sowie von dem Befehl 3.15 (fcc-hook) überschrieben.

6.3.182 reply_regexp

Typ: Regulärer Ausdruck

Voreinstellung: `^(re([\[0-9\]]+)*|aw):[\t]*`

Ein regulärer Ausdruck, der verwendet wird, um in Diskussionsfäden (Threads) und beim Antworten Nachrichten zu erkennen, die bereits Antworten sind. Die Voreinstellung erfasst das englische "Re:" und das deutsche "Aw:".

6.3.183 reply_self

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable nicht gesetzt ist und du auf eine deiner eigenen Nachrichten antwortest, nimmt Mutt an, dass du den Empfängern der ursprünglichen Nachricht antworten willst, nicht dir selbst.

6.3.184 `reply_to`

Typ: Quad-Option

Voreinstellung: ask-yes (Ja mit Nachfrage)

Wenn diese Variable auf "ask-yes" gesetzt ist und du auf eine Nachricht antwortest, fragt Mutt dich, ob es die Adresse benutzen soll, die im Header-Feld "Reply-To:" (Antworten an) angegeben ist. Wenn du mit "Nein" antwortest, wird es statt dessen die Adresse aus dem Header-Feld "From:" benutzen. Diese Einstellung ist nützlich, wenn man Mailing-Listen liest, die in das Header-Feld "Reply-To:" die Listen-Adresse setzen und du dem Autor einer Nachricht eine persönliche Antwort schicken möchtest.

6.3.185 `resolve`

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Wenn diese Variable gesetzt ist, wird der Cursor automatisch auf die nächste (wenn möglich nicht gelöschte) Nachricht vorgerückt, wenn ein Befehl ausgeführt wurde, der die aktuelle Nachricht verändert.

6.3.186 `reverse_alias`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Diese Variable bestimmt, ob Mutt den "persönlichen" Namen aus deinem Adressbuch im Index-Menü anzeigen soll, wenn es einen Adressbucheintrag findet, der auf den Absender einer Nachricht passt. Wenn du zum Beispiel den folgenden Eintrag hast:

```
alias otto abd30425@irgendwo.de (Otto Normaluser)
```

und du eine Nachricht empfangst, die folgende Kopfzeile enthält:

```
From: abd30425@irgendwo.de
```

würde als Absender im Index-Menü "Otto Normaluser" angezeigt statt "abd30425@irgendwo.de". Das ist nützlich, wenn die E-Mail-Adresse einer Person schwer lesbar ist (wie bei den alten Compuserve-Adressen).

6.3.187 `reverse_name`

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Es kann passieren, dass du auf einem bestimmten Rechner Nachrichten empfangst, sie auf einen anderen Rechner verschiebst und von dort aus auf einige der Nachrichten antwortest. Wenn diese Variable gesetzt ist, wird die vorgegebene *From*:-Zeile (der Absender) deiner Antwort aus deiner Adresse auf dem Rechner ermittelt, auf dem du die Nachrichten empfangen hast, auf die du antwortest. Wenn die Variable nicht gesetzt ist, wird die *From*:-Zeile deine Adresse auf dem momentanen Rechner verwenden.

6.3.188 reverse_realname

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Diese Variable verfeinert das Verhalten des Features 6.3.187 (`reverse_name`). Wenn sie gesetzt ist, benutzt Mutt die Adressen von eingehenden Nachrichten so wie sie sind, also einschließlich des möglicherweise darin enthaltenen Realnamen. Wenn sie nicht gesetzt ist, überschreibt Mutt den Realnamen darin grundsätzlich mit dem Wert der Variablen 6.3.179 (`realname`).

6.3.189 rfc2047_parameters

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, dekodiert Mutt RFC 2047-kodierte MIME-Parameter. Du weißt, dass du diese Variable setzen willst, wenn Mutt dir vorschlägt, einen Anhang unter einem Namen wie `=?iso-8859-1?Q?file=5F=E4=5F991116=2Ezip?=` abzuspeichern.

Wenn diese Variable interaktiv gesetzt wird, hat dies keine Auswirkung, bis man die Mailbox gewechselt hat.

Diese Option existiert, weil diese Art die RFC 2047-Kodierung zu verwenden vom Standard ausdrücklich verboten ist, aber "in freier Wildbahn" trotzdem immer wieder angetroffen wird. Es sei darauf hingewiesen, dass das Setzen dieser Variable *nicht* dazu führt, dass Mutt diese Art der Kodierung selbst *vornimmt*. Statt dessen verwendet Mutt in jedem Fall die in RFC 2231 beschriebene Kodierung.

6.3.190 save_address

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, nimmt Mutt die vollständige Adresse des Absenders, wenn es beim Abspeichern einer Nachricht einen Dateinamen für die Mailbox vorschlägt. Wenn 6.3.192 (`save_name`) oder 6.3.51 (`force_name`) ebenfalls gesetzt ist, ändert sich dadurch auch die Wahl des fcc-Ordners (des Ordners, in dem Kopien eigener Nachrichten gespeichert werden).

6.3.191 save_empty

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Wenn diese Variable auf "no" gesetzt ist, entfernt Mutt Mailbox-Dateien, die keine Nachrichten mehr enthalten, wenn es sie schließt (die einzige Ausnahme ist die 6.3.213 (Posteingangs-Mailbox), die niemals entfernt wird). Wenn sie gesetzt ist, löscht Mutt grundsätzlich keine Mailboxen.

Anmerkung: Dies betrifft nur mbox- und MMDF-Ordner, Mutt löscht keine MH- oder Maildir-Verzeichnisse.

6.3.192 save_name

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Diese Variable bestimmt, wie Kopien ausgehender Nachrichten abgespeichert werden. Wenn sie gesetzt ist, überprüft Mutt, ob eine Mailbox-Datei existiert, die zur Adresse des Empfängers passt (es tut dies,

indem es im 6.3.48 (folder)-Verzeichnis eine Mailbox mit dem *Benutzernamen* (username part) aus der Empfängeradresse als Dateiname sucht). Wenn eine solche Mailbox-Datei existiert, wird die ausgehende Nachricht in ihr gespeichert, ansonsten wird die Nachricht in der von 6.3.181 (record) angegebenen Mailbox abgelegt.

Siehe auch die Variable 6.3.51 (\$force_name).

6.3.193 score

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Wenn diese Variable auf "no" gesetzt ist, findet kein 6.3.193 (Scoring) (Bewertung von Nachrichten) statt. Das kann nützlich sein, um Scoring gezielt für ganz bestimmte Ordner abzustellen, wenn Variablen wie "6.3.194 (\$score_threshold_delete)" und ähnliche verwendet werden.

6.3.194 score_threshold_delete

Typ: Zahl

Voreinstellung: -1

Nachrichten, die mit einer Punktzahl (Score) bewertet wurden, die dem Wert dieser Variable entspricht oder darunter liegt, werden von Mutt automatisch zum Löschen vorgemerkt. Da Punktzahlen in Mutt immer größer oder gleich Null sind, wird bei der Voreinstellung der Variablen nie eine Nachricht aufgrund ihrer Bewertung gelöscht.

6.3.195 score_threshold_flag

Typ: Zahl

Voreinstellung: 9999

Nachrichten, die mit einer Punktzahl (Score) bewertet wurden, die mindestens dem Wert dieser Variablen entsprechen, werden automatisch als wichtig markiert.

6.3.196 score_threshold_read

Typ: Zahl

Voreinstellung: -1

Nachrichten, deren Punktzahl (Score) höchstens diesem Wert entspricht, werden von Mutt automatisch als bereits gelesen gekennzeichnet. Da Punktzahlen in Mutt immer mindestens Null sind, wird bei der Voreinstellung der Variablen nie eine Nachricht automatisch als gelesen markiert.

6.3.197 send_charset

Typ: Zeichenkette

Voreinstellung: "us-ascii:iso-8859-1:utf-8"

Diese Variable enthält eine Liste von Zeichensätze für ausgehende Nachrichten. Mutt benutzt den ersten Zeichensatz, in den der Text exakt konvertiert werden kann. Falls dein "6.3.21 (\$charset)" nicht iso-8859-1 ist und Empfänger möglicherweise kein UTF-8 verstehen, ist es sinnvoll, in die Liste einen anderen, weit verbreiteten Zeichensatz (wie z.B. iso-8859-2, koi8-r oder iso-2022-jp) entweder anstelle oder nach "iso-8859-1" einzufügen.

6.3.198 sendmail

Typ: Pfadname

Voreinstellung: `"/usr/sbin/sendmail -oem -oi"`

Gibt das Programm (mit den nötigen Parametern) an, das Mutt zum Versenden von Nachrichten verwenden soll. Mutt erwartet, dass das angegebene Programm zusätzliche Parameter als Empfängeradressen interpretiert.

6.3.199 sendmail_wait

Typ: Zahl

Voreinstellung: 0

Bestimmt, wie viele Sekunden lang Mutt abwarten soll, dass der 6.3.198 (sendmail)-Prozess sich beendet, bevor es aufgibt und die Auslieferung in den Hintergrund verschiebt.

Mutt interpretiert den Wert dieser Variablen wie folgt:

>0

Anzahl von Sekunden, während der auf Sendmail gewartet werden soll

0

Mutt wartet in jedem Fall, bis Sendmail sich beendet hat

<0

Sendmail wird immer ohne zu warten im Hintergrund ausgeführt

Beachte, dass die Ausgabe des Kind-Prozesses bei jedem anderen Wert als 0 in eine temporäre Datei geschrieben wird. Wenn ein Fehler auftritt, wirst du darüber informiert, wo du die Ausgabe finden kannst.

6.3.200 shell

Typ: Pfadname

Voreinstellung: `""`

Der Befehl, der benutzt werden soll, um eine Subshell aufzurufen. In der Voreinstellung wird die Login-Shell des Benutzers verwendet, die in `/etc/passwd` festgelegt wurde.

6.3.201 sig_dashes

Typ: Wahrheitswert

Voreinstellung: `yes` (gesetzt)

Wenn diese Variable gesetzt ist, wird vor deiner 6.3.203 (Signatur) (signature) eine Zeile eingefügt, die `"- "` enthält. Es wird **nachdrücklich** empfohlen, dass du diese Funktion nicht ausschaltest, wenn deine Signatur mehr als nur deinen Namen enthält. Der Grund dafür ist, dass viele Software-Pakete anhand von `"- \n"` Signaturen erkennen können. Mutt hast zum Beispiel die Fähigkeit, die Signatur im eingebauten Pager in einer anderen Farbe darzustellen.

6.3.202 sig_on_top

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Falls die Variable gesetzt ist, wird die Signatur vor dem gequoteten oder weitergeleiteten Text eingefügt. Es wird **nachdrücklich** empfohlen, dass du diese Variable nicht setzt, außer du weißt wirklich, was du tust, und bist bereit, Streitereien mit den Netiquette-Wächtern zu erdulden.

6.3.203 signature

Typ: Pfadname

Voreinstellung: "~/signature"

Gibt den Namen der Datei an, die deine Signatur enthält. Sie wird an alle ausgehenden Nachrichten angehängt. Wenn der Dateiname mit einem Piping-Zeichen ("|") endet, nimmt Mutt an, dass der Dateiname ein Shell-Befehl ist, von dessen stdout (Standard-Ausgabe) die Signatur eingelesen werden soll.

6.3.204 simple_search

Typ: Zeichenkette

Voreinstellung: "~f %s | ~s %s"

Gibt an, wie Mutt eine einfache Suche in ein echtes Suchmuster umwandeln soll. Ein Suchausdruck ist eine einfache Suche, wenn er keinen der ~-Operatoren verwendet. Siehe 4.2 (Suchmuster) (patterns), um mehr Informationen über Suchmuster zu erhalten.

Wenn du zum Beispiel als Such- oder Begrenzungsmuster einfach otto eingibst, wird Mutt es in den Wert umwandeln, den diese Variable angibt. Im Falle der Voreinstellung wäre das:

```
~f otto | ~s otto
```

6.3.205 smart_wrap

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Regelt die Anzeige von Zeilen im internen Pager, die länger als die Bildschirmbreite sind. Wenn diese Variable gesetzt ist, werden zu lange Zeilen zwischen zwei Wörtern umgebrochen. Wenn sie nicht gesetzt ist, werden die Zeilen einfach am Bildschirmrand umgebrochen. Siehe auch die Variable 6.3.92 (\$markers).

6.3.206 smileys

Typ: regulärer Ausdruck

Voreinstellung: "(>From)|(:[-^]?[[](><}{|/DP))"

Der *pager* benutzt diese Variable, um ein paar der üblichen irrtümlichen Treffer des 6.3.176 (quote_regexp)-Suchausdrucks abzufangen, vor allem Smileys :-) am Anfang einer Zeile.

6.3.207 sleep_time

Typ: Zahl

Voreinstellung: 1

Diese Variable bestimmt die Wartezeit in Sekunden, um bestimmte Informationsmeldungen anzuzeigen, während man von Mailbox zu Mailbox springt und nach dem Löschen von Nachrichten in der aktuellen Mailbox. Die Voreinstellung wartet eine Sekunde, daher schaltet ein Wert von 0 diese Option ab.

6.3.208 sort

Typ: Sortier-Kriterium

Voreinstellung: date (Zeitpunkt)

Gibt an, wie die Nachrichten im *Index*-Menü sortiert werden sollen. Gültige Werte sind

```
date oder date-sent (Zeitpunkt des Absendens)
date-received (Zeitpunkt des Eintreffens)
from (Absender)
mailbox-order (unsortiert)
score (Wertung)
size (Größe)
subject (Betreff)
threads (Gesprächsfäden)
to (Adressat)
```

Du kannst zusätzlich den Präfix reverse- davor setzen, um eine umgekehrte Sortierreihenfolge zu erreichen (Beispiel: set sort=reverse-date-sent).

6.3.209 sort_alias

Typ: Sortier-Kriterium

Voreinstellung: alias (Kurzname)

Gibt an, wie die Einträge im "alias"-Menü (Adressbuch) sortiert werden. Folgende Werte sind gültig:

```
address (alphabetische Sortierung nach E-Mail-Adresse)
alias (alphabetische Sortierung nach Kurzname)
unsorted (Reihenfolge wie beim Einlesen)
```

6.3.210 sort_aux

Typ: Sortier-Kriterium

Voreinstellung: date (Zeitpunkt)

Wenn Mutt nach Gesprächsfäden (Threads) sortiert, kontrolliert diese Variable, wie Threads untereinander sortiert werden und wie die einzelnen Teile innerhalb der Gesprächsfäden sortiert werden sollen. Für diese Variable sind die gleichen Werte gültig wie für 6.3.208 (sort), außer threads (wenn man das angibt, verwendet Mutt ganz einfach date-sent). Du kannst zusätzlich zu dem Präfix reverse- auch last- vor die Kriterien stellen; last- muss dabei jedoch nach reverse- stehen. Das Präfix last- sorgt dafür, dass Nachrichten gleicher Schachtelungstiefe (etwa Threads untereinander) im Vergleich zu ihren "Geschwistern" nach ihrem neuesten "Nachkommen" einsortiert werden, wobei der Rest von sort_aux als Sortierkriterium benutzt wird. Ein Beispiel: set sort_aux=last-date-received würde bedeuten, dass ein Gesprächsfaden, in dem eine neue Nachricht empfangen wird, als letzter angezeigt wird (bzw. als erster, wenn man set sort=reverse-threads angegeben hat). Anmerkung: Für eine Umkehrung von 6.3.208 (\$sort) ist 6.3.210 (\$sort_aux) wieder umgekehrt (was nicht der korrekte Weg ist, aber beibehalten wurde, um existierende Konfigurationen nicht zu stören).

6.3.211 sort_browser

Typ: Sortier-Kriterium

Voreinstellung: alpha (alphabetisch)

Gibt an, wie die Einträge in Datei-Listen sortiert werden sollen. In der Voreinstellung werden die Einträge alphabetisch sortiert. Gültige Werte sind:

```
alpha (alphabetisch)
date (nach Datum)
size (nach Größe)
unsorted (unsortiert)
```

Zusätzlich kannst du die Präfix reverse- davor stellen, um eine umgekehrte Sortierreihenfolge zu erreichen (Beispiel: set sort_browser=reverse-date).

6.3.212 sort_re

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Diese Variable wird nur dann benötigt, wenn nach Gesprächsfäden sortiert wird und 6.3.217 (strict_threads) nicht gesetzt ist. In diesem Fall ändert es die Heuristik, die Mutt verwendet, um Nachrichten nach dem Betreff zu Gesprächsfäden zusammenzufassen. Wenn sort_re gesetzt ist, hält Mutt eine Nachricht aufgrund des Betreffs nur dann für ein "Kind" einer anderen Nachricht, wenn der Betreff der Kind-Nachricht mit einer Zeichenkette anfängt, auf die das Suchmuster in 6.3.182 (reply_regexp) zutrifft. Wenn sort_re nicht gesetzt ist, fügt Mutt sie unabhängig davon dem Gesprächsfaden hinzu, solange die nicht von 6.3.182 (reply_regexp) erfassten Teile des Betreffs bei beiden Nachrichten identisch sind.

6.3.213 spoolfile

Typ: Pfadname

Voreinstellung: ""

Wenn deine Posteingangs-Datei (spool-Mailbox) an einem unüblichen Ort ist, sodass Mutt sie nicht automatisch findet, kannst du mit dieser Variable ihren Pfad angeben. Mutt setzt diese Variable automatisch auf den Wert der Umgebungsvariablen \$MAIL, wenn du sie nicht ausdrücklich setzt.

6.3.214 status_chars

Typ: Zeichenkette

Voreinstellung: "-*%A"

Bestimmt die Zeichen, die vom Platzhalter "%r" in "6.3.215 (status_format)" verwendet werden. Der erste Buchstabe wird eingesetzt, wenn die Mailbox unverändert ist. Der zweite wird verwendet, wenn die Mailbox verändert wurde und neu synchronisiert werden muss. Der dritte kommt zum Einsatz, wenn die Mailbox nur zum Lesen geöffnet ist oder sie nicht zurückgeschrieben wird, wenn man sie verlässt (du kannst mit dem Befehl toggle-write, in der Voreinstellung auf "%" gebunden, festlegen, ob Änderungen an einer Mailbox-Datei auf Platte geschrieben werden sollen oder nicht). Das vierte Zeichen wird verwendet, um anzuzeigen, dass der aktuelle Ordner im attach-message (Nachricht anhängen)-Modus geöffnet wurde (bestimmte Befehle wie Erstellen einer neuen Nachricht, antworten, weiterleiten u.s.w. können in diesem Modus nicht ausgeführt werden).

6.3.215 status_format

Typ: Zeichenkette

Voreinstellung: "-%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)—"

Bestimmt das Format der Status-Zeile, die im *Index*-Menü angezeigt wird. Diese Zeichenkette ähnelt dem 6.3.83 (\$index_format), bietet jedoch seine eigenen printf()-ähnlichen Platzhalter:

%b

Anzahl der Mailbox-Dateien mit neuen Nachrichten *

%d

Anzahl der gelöschten Nachrichten *

%h

Name des lokalen Rechners (local hostname)

%f

der volle Pfadname der aktuellen Mailbox-Datei

%F

Anzahl der markierten Nachrichten *

%l

Größe (in Byte) der aktuellen Mailbox-Datei *

%L

Größe (in Byte) der angezeigten Nachrichten (d.h. auf die das aktuelle Begrenzungsmuster zutrifft) *

%m

Anzahl der Nachrichten in der Mailbox-Datei *

%M

Anzahl der angezeigten Nachrichten (d.h. auf die das aktuelle Begrenzungsmuster zutrifft) *

%n

Anzahl der neuen Nachrichten in der Mailbox-Datei *

%o

Anzahl der alten ungelesenen Nachrichten

%p

Anzahl der zurückgestellten Nachrichten *

%P

Position des Cursors im Index (in Prozent)

%r

verändert/nur-lesen/nicht-schreiben/Nachrichten-anhängen-Indikator, gemäß 6.3.214 (\$status_chars)

%s

aktueller Sortiermodus (6.3.208 (\$sort))

%S

aktuelle Hilfs-Sortiermethode (ref id=sort_aux" name="\$sort_aux">)

%t

Anzahl der ausgewählten Nachrichten *

%u

Anzahl der ungelesenen Nachrichten *

%v

Versions-Anzeige von Mutt

%V

aktuelles Begrenzungsmuster (falls aktiv) *

%>X

richte den Rest der Zeichenkette rechtsbündig aus und fülle mit "X" auf

%|X

fülle bis zum Ende der Zeile mit "X" auf

* = kann wahlweise ausgegeben werden, wenn es ungleich Null ist

Einige der genannten Platzhalter können verwendet werden, um eine Zeichenkette nur dann auszugeben, wenn ihr Wert ungleich Null ist. Zum Beispiel möchtest du die Zahl der markierten Nachrichten möglicherweise nur dann sehen, wenn überhaupt Nachrichten markiert sind, weil Null nicht gerade aussagekräftig ist. Um eine Zeichenkette aufgrund eines der obigen Platzhalter optional auszugeben, wird folgende Konstruktion verwendet

`%(?<sequence_char>?(?<optional_string>?)`

wobei *sequence_char* ein Zeichen aus der obigen Tabelle und *optional_string* diejenige Zeichenkette ist, die du ausgegeben haben möchtest, wenn *status_char* ungleich Null ist. *optional_string* **darf** neben normalem Text auch andere Platzhalter enthalten; es ist jedoch **nicht** erlaubt, optionale Zeichenketten zu schachteln.

Es folgt ein Beispiel, das vorführt, wie man die Anzahl der neuen Nachrichten in einer Mailbox-Datei optional ausgeben kann: `%(?n?%n neue Nachrichten.?`

Zusätzlich kann man zwischen zwei Zeichenketten umschalten; dabei wird die erste ausgegeben, wenn ein Wert Null ist, die zweite, wenn er ungleich Null ist. Dazu wird folgende Konstruktion verwendet: `%(?<sequence_char>?(?<if_string>&<else_string>?)`

Du kannst außerdem erzwingen, dass die Ersetzung eines Platzhalters klein geschrieben wird, indem du dem entsprechenden Zeichen einen Unterstrich (_) voranstellst. Wenn du zum Beispiel den Namen des lokalen Rechners klein geschrieben anzeigen lassen willst, kannst du folgendes verwenden: `%_h`

6.3.216 status_on_top

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, wird die Status-Zeile statt in der Nähe des unteren Randes in der ersten Zeile des Bildschirms ausgegeben.

6.3.217 strict_threads

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, wird Mutt beim 6.3.208 (Sortieren) nach Gesprächsfäden (Threads) nur die Header-Felder "In-Reply-To" und "References" auswerten. In der Voreinstellung werden Nachrichten mit dem gleichen Betreff (Subject) in "Pseudo-Gesprächsfäden" zusammengefasst. Unter Umständen ist dies nicht wünschenswert, wie zum Beispiel in einer persönlichen Mailbox, in der es verschiedene voneinander unabhängige Nachrichten mit dem Betreff "Hallo" geben könnte, die in diesem Fall zusammengefasst würden.

6.3.218 suspend

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Wenn diese Variable *nicht gesetzt* ist, lässt Mutt sich nicht unterbrechen, wenn man als Benutzer die "Unterbrechen"-Taste ("suspend key", normalerweise Ctrl-Z) des Terminals drückt. Das ist nützlich, wenn du Mutt mit einem Kommando wie `xterm -e mutt` in einem `xterm` laufen lässt.

6.3.219 text_flow

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn die Variable gesetzt ist, erstellt Mutt Anhänge des Typs `text/plain; format=flowed`. Dieses Format wird von mancher Mailersoftware leichter verarbeitet und sieht eigentlich wie normaler Text aus. Um dieses Feature benutzen zu können, benötigst du einen Editor, der dieses Format unterstützt.

Beachte, dass 6.3.82 (`$indent_string`) ignoriert wird, wenn diese Option gesetzt ist.

6.3.220 thread_received

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Mit dieser Variable benutzt Mutt die Empfangszeit anstelle der Sendezeit, um Nachrichten nach dem Betreff in Threads anzuordnen.

6.3.221 thorough_search

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Betrifft die `~b`- und `~h`-Such-Operatoren, die weiter oben im Abschnitt 4.2 (Suchmuster) (`patterns`) beschrieben werden. Wenn diese Variable gesetzt ist, werden die Kopfdaten (Header) und Anhänge der zu durchsuchenden Nachrichten vor dem Suchen dekodiert. Wenn sie nicht gesetzt ist, werden die Nachrichten so durchsucht, wie sie im Ordner stehen.

6.3.222 tilde

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Wenn diese Variable gesetzt ist, wird der interne Pager die Leerzeilen bis zur Unterkante des Bildschirms mit jeweils einer Tilde (˘) auffüllen.

6.3.223 timeout

Typ: Zahl

Voreinstellung: 600

Diese Variable bestimmt, *wie viele Sekunden lang* Mutt im Hauptmenü auf einen Tastenanschlag wartet, bevor es aufgrund der Zeitüberschreitung überprüft, ob neue Mail angekommen ist. Bei einem Wert kleiner oder gleich Null wird Mutt nichts aufgrund einer Zeitüberschreitung tun.

6.3.224 tmpdir

Typ: Pfadname

Voreinstellung: ""

Diese Variable ermöglicht es festzulegen, wo Mutt die temporären Dateien anlegen soll, die zum Anzeigen und Erstellen von Nachrichten notwendig sind.

6.3.225 to_chars

Typ: Zeichenkette

Voreinstellung: "+TCFL"

Bestimmt das Zeichen, das zur Kennzeichnung von Nachrichten verwendet wird, die an dich adressiert sind. Das erste Zeichen wird verwendet, wenn die Nachricht nicht an deine Adresse gerichtet ist (Voreinstellung: Leerzeichen). Das zweite wird eingesetzt, wenn du der einzige Empfänger einer Nachricht bist (Voreinstellung: +). Das dritte kommt zum Einsatz, wenn deine Adresse im Header-Feld "To" erscheint, du jedoch nicht der einzige Empfänger der Nachricht bist (Voreinstellung: T). Der vierte Buchstabe wird verwendet, wenn deine Adresse im Kopf-Feld "Cc" angegeben ist und du nicht der einzige Empfänger bist. Der fünfte Buchstabe wird verwendet, um Mail zu kennzeichnen, die *von dir* versandt wurde. Der sechste Buchstabe dient zur Kennzeichnung von Mails, die die an eine Mailingliste gingen, auf der du eingeschrieben bist (Voreinstellung: L).

6.3.226 tunnel

Typ: Zeichenkette

Voreinstellung: ""

Das Setzen dieser Variable veranlasst Mutt, eine Pipe zu einem Befehl anstelle eines Raw-Sockets zu öffnen. Du kannst dies benutzen, um eine authentifizierte Verbindung zu deinem IMAP/POP3-Server aufzubauen. Beispiel:

```
tunnel="ssh -q mailhost.net /usr/local/libexec/imapd"
```

Anmerkung: Damit dieses Beispiel funktioniert, musst du dich auf dem entfernten Rechner ohne Passworteingabe einloggen können.

6.3.227 use_8bitmime

Typ: Wahrheitswert

Voreinstellung: no (ungesetzt)

Warnung: Setze diese Variable nicht, es sei denn du benutzt eine Version von Sendmail, die den Parameter `-B8BITMIME` unterstützt (so zum Beispiel `sendmail 8.8.x`) oder du wirst unter Umständen nicht mehr in der Lage sein, Nachrichten zu versenden.

Wenn diese Variable *gesetzt* ist, ruft Mutt den Befehl 6.3.198 (`$sendmail`) mit dem Parameter `-B8BITMIME` auf, wenn er 8-bit Nachrichten verschickt, um das ESMTP-Interface zu aktivieren.

6.3.228 use_domain

Typ: Wahrheitswert

Voreinstellung: `yes` (gesetzt)

Wenn diese Variable *gesetzt* ist, wird Mutt alle lokalen Adressen (solche, die keinen `"@host"`-Teil haben) durch den Wert der Variablen 6.3.67 (`$hostname`) ergänzen. Wenn sie *nicht gesetzt* ist, werden keine Adressen ergänzt.

6.3.229 use_from

Typ: Wahrheitswert

Voreinstellung: `yes` (gesetzt)

Wenn diese Variable *gesetzt* ist, erzeugt Mutt das Header-Feld `'From:'`, wenn er Nachrichten verschickt. Wenn es *nicht gesetzt* ist, wird das Header-Feld `'From:'` nicht erzeugt, es sei denn der Benutzer setzt explizit eins, indem er den Befehl 3.12 (`my_hdr`) verwendet.

6.3.230 user_agent

Typ: Wahrheitswert

Voreinstellung: `yes` (gesetzt)

Wenn diese Variable *gesetzt* ist, fügt Mutt ausgehenden Nachrichten eine `"User-Agent"`-Header-Zeile hinzu, in der steht, mit welcher Version von Mutt die Nachricht erstellt wurde.

6.3.231 visual

Typ: Pfadname

Voreinstellung: `""`

Gibt den visuellen Editor an, der aufgerufen werden soll, wenn im eingebauten Editor der Befehl `~v` aufgerufen wird.

6.3.232 wait_key

Typ: Wahrheitswert

Voreinstellung: `yes` (gesetzt)

Bestimmt, ob Mutt nach den Befehlen `shell-escape`, `pipe-message`, `pipe-entry`, `print-message` und `print-entry` auf einen Tastendruck warten soll.

Die Variable wird ebenfalls verwendet, wenn man mit der automatischen Ansicht (5.4 (`auto.view`)) einen Anhang betrachtet, vorausgesetzt, dass der zugehörige mailcap-Eintrag den Parameter `needsterminal` gesetzt hat und das externe Programm interaktiv ist.

Wenn sie gesetzt ist, wartet Mutt immer auf einen Tastendruck. Wenn sie nicht gesetzt ist, wartet Mutt nur dann auch eine Taste, wenn der externe Befehl einen Statuscode ungleich Null zurückgegeben hat.

6.3.233 weed

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Wenn diese Variable gesetzt ist, ignoriert Mutt unerwünschte Header ("header weeding"), wenn man Nachrichten anzeigt, weiterleitet, ausdrückt oder darauf antwortet.

6.3.234 wrap_search

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Bestimmt, ob Suchen am Ende einer Mailbox "umbrechen" sollen.

Wenn diese Variable gesetzt ist, beginnt die Suche nach Erreichen der letzten (bzw. ersten) Nachricht wieder von vorn (bzw. hinten). Wenn sie nicht gesetzt ist, endet die Suche dort.

6.3.235 wrapmargin

Typ: Zahl

Voreinstellung: 0

Diese Variable kontrolliert den Rand, der auf der rechten Seite übrig bleibt, wenn Mutts Pager intelligent die Zeilen umbricht (smart wrapping).

6.3.236 write_inc

Typ: Zahl

Voreinstellung: 10

Wenn Mutt eine Mailbox abspeichert, erfolgt nach jeder *write_inc*-ten Nachricht eine Ausgabe, um den Fortschritt anzuzeigen. Wenn diese Variable auf 0 gesetzt ist, erscheint nur eine einzelne Anzeige, bevor die Mailbox geschrieben wird.

Siehe auch die Variable 6.3.177 (*\$read_inc*).

6.3.237 write_bcc

Typ: Wahrheitswert

Voreinstellung: yes (gesetzt)

Bestimmt, ob Mutt auch die Kopfzeile "Bcc" ausgibt, wenn es Nachrichten auf das Versenden vorbereitet. Möglicherweise wollen Benutzer von Exim diese Möglichkeit nutzen.

6.4 Funktionen

Es folgt eine Liste der verfügbaren Funktionen, aufgeführt nach dem Menü, in dem sie aufgerufen werden können. Neben der Standard-Tastaturbindung wird auch eine Erklärung gegeben, was die jeweilige Funktion tut. Die Tastaturbindungen dieser Funktionen können über den Befehl 3.3 (bind) geändert werden.

6.4.1 allgemein (generic)

Das *allgemeine* Menü ist kein wirkliches Menü, sondern definiert gemeinsame Funktionen (wie zum Beispiel zur Navigation), die in allen Menüs außer dem *Pager* und dem *Editor* verfügbar sind. Eine Änderung der Einstellungen für dieses Menü wird (mit den genannten Ausnahmen) die Standard-Tastaturbindungen für alle Menüs beeinflussen.

bottom-page		L	Zum Ende der Seite
current-bottom	ungebunden		Bewege Eintrag zum unteren Ende des Bildschirms
current-middle	ungebunden		Bewege Eintrag zur Bildschirmmitte
current-top	ungebunden		Bewege Eintrag zum Bildschirmanfang
enter-command		:	Gib ein muttrc-Kommando ein
exit		q	Menü verlassen
first-entry		=	Gehe zum ersten Eintrag
half-down]	Gehe 1/2 Seite nach unten
half-up		[Gehe 1/2 Seite nach oben
help		?	Diese Darstellung
jump		Zahl	Springe zu einer Index-Nummer
last-entry		*	Springe zum letzten Eintrag
middle-page		M	Gehe zur Seitenmitte
next-entry		j	Gehe zum nächsten Eintrag
next-line		>	Gehe eine Zeile nach unten
next-page		z	Gehe zur nächsten Seite
previous-entry		k	Gehe zum vorigen Eintrag
previous-line		<	Gehe eine Zeile nach oben
previous-page		Z	Gehe zur vorigen Seite
refresh		~L	Erzeuge Bildschirmanzeige neu
search		/	Suche mit regulärem Ausdruck
search-next		n	Suche nächsten Treffer
search-opposite	ungebunden		Suche nächsten Treffer in umgekehrter Richtung
search-reverse		ESC /	Suche rückwärts nach regulärem Ausdruck
select-entry		EINGABE	Wähle den aktuellen Eintrag aus
shell-escape		!	Rufe Kommando in Shell auf
tag-entry		t	Markiere aktuellen Eintrag (Schalter)
tag-prefix		;	Wende nächste Funktion auf markierte Nachrichten an
top-page		H	Springe zum Anfang der Seite

6.4.2 Index

bounce-message		b	Versende Nachricht erneut an anderen Empfänger
change-folder		c	Öffne eine andere Mailbox
change-folder-readonly	ESC	c	Öffne eine andere Mailbox im Nur-Lesen-Modus
check-traditional-pgp	ESC	P	Überprüfe die Nachricht auf klassisches PGP
clear-flag		W	Entferne einen Status-Indikator
copy-message		C	Kopiere Nachricht in Datei/Mailbox
create-alias		a	Erzeuge Adressbucheintrag für Absender
decode-copy	ESC	C	Erzeuge decodierte Kopie der Nachricht in einer Datei/Mailbox
decode-save	ESC	s	Speichere die Nachricht decodiert in einer

		Datei/Mailbox (verschieben)
delete-message	d	Lösche die aktuelle Nachricht
delete-pattern	D	Lösche Nachrichten nach Muster
delete-subthread	ESC d	Lösche alle Nachrichten im Diskussionsfadenteil
delete-thread	^D	Lösche alle Nachrichten im Diskussionsfaden
display-address	@	Zeige komplette Absenderadresse
display-toggle-weed	h	Zeige Nachricht an und schalte zwischen allen/wichtigen Headern um
display-message	EINGABE	Zeige Nachricht an
edit	e	Editiere "rohe" Nachricht
edit-type	^E	Editiere den Content-Type der Nachricht
exit	x	Menü verlassen
extract-keys	^K	Extrahiere öffentliche PGP-Schlüssel
fetch-mail	G	Hole Nachrichten von POP-Server
flag-message	F	Markiere Nachricht als wichtig
forget-passphrase	^F	Entferne PGP-Mantra aus Speicher
forward-message	f	Leite Nachricht mit Kommentar weiter
group-reply	g	Antworte an alle Empfänger
limit	l	Wähle anzuzeigende Nachrichten mit Muster aus
list-reply	L	Antworte an Mailing-Listen
mail	m	Erzeuge neue Nachricht
mail-key	ESC k	Verschicke öffentlichen PGP-Schlüssel
next-new	TAB	Springe zur nächsten neuen Nachricht
next-subthread	ESC n	Springe zum nächsten Diskussionsfadenteil
next-thread	^N	Springe zum nächsten Diskussionsfaden
next-undeleted	j	Springe zur nächsten ungelöschten Nachricht
next-unread	ungebunden	Springe zur nächsten ungelesenen Nachricht
parent-message	P	Springe zur Bezugsnachricht im Diskussionsfaden
pipe-message		Bearbeite (pipe) Nachricht/Anhang mit Shell-Kommando
previous-new	ESC TAB	Springe zur vorigen neuen Nachricht
previous-page	Z	Gehe zur vorigen Seite
previous-subthread	ESC p	Springe zum vorigen Diskussionsfadenteil
previous-thread	^P	Springe zum vorigen Diskussionsfaden
previous-undeleted	k	Springe zur letzten ungelöschten Nachricht
previous-unread	ungebunden	Springe zur vorigen ungelesenen Nachricht
print-message	p	Drucke aktuellen Eintrag
query	Q	Externe Adressenabfrage
quit	q	Speichere Änderungen in Mailbox und beende das Programm
read-subthread	ESC r	Markiere den aktuellen Diskussionsfadenteil als gelesen
read-thread	^R	Markiere den aktuellen Diskussionsfaden als gelesen
recall-message	R	Bearbeite eine zurückgestellte Nachricht
reply	r	Beantworte Nachricht
save-message	s	Speichere Nachricht/Anhang in Datei (verschieben)
set-flag	w	Setze Statusindikator einer Nachricht
show-version	V	Zeige die Versionsnummer und das Datum von Mutt
show-limit	ESC l	Zeige derzeit aktives Begrenzungsmuster

sort-mailbox	o	Sortiere Nachrichten
sort-reverse	O	Sortiere Nachrichten in umgekehrter Reihenfolge
sync-mailbox	\$	Speichere Änderungen in Mailbox
tag-pattern	T	Markiere Nachrichten nach Muster
tag-thread	ESC t	Markiere aktuellen Diskussionsfaden
toggle-new	N	Setze/entferne den "neu"-Indikator einer Nachricht
toggle-write	%	Schalte Sichern von Änderungen ein/aus
undelete-message	u	Entferne Löschkmarkierung vom aktuellen Eintrag
undelete-pattern	U	entferne Löschkmarkierung nach Muster
undelete-subthread	ESC u	Entferne Löschkmarkierung von allen Nachrichten im Diskussionsfadenteil
undelete-thread	^U	Entferne Löschkmarkierung von allen Nachrichten im Diskussionsfaden
untag-pattern	^T	Entferne Markierung nach Muster
view-attachments	v	Zeige MIME-Anhänge

6.4.3 Pager

bottom	ungebunden	Springe zum Ende der Nachricht
bounce-message	b	Versende Nachricht erneut an anderen Empfänger
change-folder	c	Öffne eine andere Mailbox
change-folder-readonly	ESC c	Öffne eine andere Mailbox im Nur-Lesen-Modus
check-traditional-pgp	ESC P	Überprüfe die Nachricht auf klassisches PGP
copy-message	C	Kopiere Nachricht in Datei/Mailbox
create-alias	a	Erzeuge Adressbucheintrag für Absender
collapse-parts	v	Schalte Anzeige von Unter-teilen an/aus
delete-message	d	Lösche die aktuelle Nachricht
delete-subthread	ESC d	Lösche alle Nachrichten im Diskussionsfadenteil
delete-thread	^D	Lösche alle Nachrichten im Diskussionsfaden
display-address	@	Zeige komplette Absenderadresse
display-toggle-weed	h	Zeige Nachricht an und schalte zwischen allen/wichtigen Headern um
edit	e	Editiere "rohe" Nachricht
enter-command	:	Gib ein muttrc-Kommando ein
exit	i	Kehre zum Hauptmenü zurück
extract-keys	^K	Extrahiere öffentliche PGP-Schlüssel
flag-message	F	Markiere Nachricht als wichtig
forget-passphrase	^F	Entferne PGP-Mantra aus Speicher
forward-message	f	Leite Nachricht mit Kommentar weiter
group-reply	g	Antworte an alle Empfänger
half-up	ungebunden	Gehe 1/2 Seite nach oben
half-down	ungebunden	Gehe 1/2 Seite nach unten
help	?	Diese Darstellung
list-reply	L	Antworte an Mailing-Listen
mail	m	Erzeuge neue Nachricht
mail-key	ESC k	Verschicke öffentlichen PGP-Schlüssel
mark-as-new	N	Setze/entferne den "neu"-Indikator einer Nachricht
next-line	EINGABE	Gehe eine Zeile nach unten

next-entry	J	Gehe zum nächsten Eintrag
next-new	TAB	Springe zur nächsten neuen Nachricht
next-page	LEERZEICHEN	Gehe zur nächsten Seite
next-subthread	ESC n	Springe zum nächsten Diskussionsfadenteil
next-thread	^N	Springe zum nächsten Diskussionsfaden
next-undeleted	j	Springe zur nächsten ungelöschten Nachricht
next-unread	ungebunden	Springe zur nächsten ungelesenen Nachricht
parent-message	P	Springe zur Bezugsnachricht im Diskussionsfaden
pipe-message		Bearbeite (pipe) Nachricht/Anhang mit Shell-Kommando
previous-line	RÜCKTASTE	Gehe eine Zeile nach oben
previous-entry	K	Gehe zum vorigen Eintrag
previous-new	ungebunden	Springe zur vorigen neuen Nachricht
previous-page	-	Gehe zur vorigen Seite
previous-subthread	ESC p	Springe zum vorigen Diskussionsfadenteil
previous-thread	^P	Springe zum vorigen Diskussionsfaden
previous-undeleted	k	Springe zur letzten ungelöschten Nachricht
previous-unread	ungebunden	Springe zur vorigen ungelesenen Nachricht
print-message	p	Drucke aktuellen Eintrag
quit	Q	Speichere Änderungen in Mailbox und beende das Programm
read-subthread	ESC r	Markiere den aktuellen Diskussionsfadenteil als gelesen
read-thread	^R	Markiere den aktuellen Diskussionsfaden als gelesen
recall-message	R	Bearbeite eine zurückgestellte Nachricht
redraw-screen	^L	Erzeuge Bildschirmanzeige neu
reply	r	Beantworte Nachricht
save-message	s	Speichere Nachricht/Anhang in Datei
search	/	Suche mit regulärem Ausdruck
search-next	n	Suche nächsten Treffer
search-opposite	ungebunden	Suche nächsten Treffer in umgekehrter Richtung
search-reverse	ESC /	Suche rückwärts nach regulärem Ausdruck
search-toggle	\	Schalte Suchtreffer-Hervorhebung ein/aus
shell-escape	!	Rufe Kommando in Shell auf
show-version	V	Zeige Versionsnummer
skip-quoted	S	Übergehe zitierten Text
sync-mailbox	\$	Speichere Änderungen in Mailbox
tag-message	t	Markiere aktuellen Eintrag
toggle-quoted	T	Schalte Anzeige von zitiertem Text ein/aus
top	^	Springe zum Nachrichtenanfang
undelete-message	u	Entferne Löschkmarkierung vom aktuellen Eintrag
undelete-subthread	ESC u	Entferne Löschkmarkierung von allen Nachrichten im Diskussionsfadenteil
undelete-thread	^U	Entferne Löschkmarkierung von allen Nachrichten im Diskussionsfaden
view-attachments	v	Zeige MIME-Anhänge

6.4.4 Adressbuch (aliases)

search	/	Suche mit regulärem Ausdruck
search-next	n	Suche nächsten Treffer
search-reverse	ESC /	Suche rückwärts nach regulärem Ausdruck

6.4.5 Adressenabfrage (query)

create-alias	a	Erzeuge Adressbucheintrag für Absender
mail	m	Erzeuge neue Nachricht
query	Q	Externe Adressenabfrage
query-append	A	Hänge neue Abfrageergebnisse an vorhandene an
search	/	Suche mit regulärem Ausdruck
search-next	n	Suche nächsten Treffer
search-opposite	ungebunden	Suche nächsten Treffer in umgekehrter Richtung
search-reverse	ESC /	Suche rückwärts nach regulärem Ausdruck

6.4.6 Anhänge (attach)

bounce-message	b	Versende Nachricht erneut an anderen Empfänger
decode-copy	ESC C	Erzeuge decodierte Kopie der Nachricht in einer Datei/Mailbox
decode-save	ESC s	Speichere die Nachricht decodiert in einer Datei/Mailbox (verschieben)
delete-entry	d	Lösche
display-toggle-weed	h	Zeige Nachricht an und schalte zwischen allen/wichtigen Headern um
edit-type	^E	Editiere den Content-Type der Nachricht
extract-keys	^K	Extrahiere öffentliche PGP-Schlüssel
forward-message	f	Leite Nachricht mit Kommentar weiter
group-reply	g	Antworte an alle Empfänger
list-reply	L	Antworte an Mailing-Listen
pipe-entry		Bearbeite (pipe) Nachricht/Anhang mit Shell-Kommando
print-entry	p	Drucke aktuellen Eintrag
reply	r	Beantworte Nachricht
resend-message	ESC e	Versicke eine Nachricht erneut und erhalte die MIME-Struktur
save-entry	s	Speichere Nachricht/Anhang in Datei
undeleete-entry	u	Entferne Löschemarkierung vom aktuellen Eintrag
view-attach	EINGABE	Zeige Anhang, wenn nötig via Mailcap
view-mailcap	m	Erzwinge Mailcap für Anhang
view-text	T	Zeige Anhang als Text an

6.4.7 Nachricht erstellen (compose)

attach-file	a	Hänge Datei(en) an diese Nachricht an
attach-message	A	Hänge Nachricht(en) an diese Nachricht an
attach-key	ESC k	Hänge öffentlichen PGP-Schlüssel an
copy-file	C	Speichere Nachricht/Anhang in Datei

detach-file	D	Lösche
display-toggle-weed	h	Zeige Nachricht an und schalte zwischen allen/wichtigen Headern um
edit-bcc	b	Editiere die BCC-Liste
edit-cc	c	Editiere die CC-Liste
edit-description	d	Editiere die Beschreibung des Anhangs
edit-encoding	^E	Editiere die Codierung des Anhangs
edit-fcc	f	Wähle Datei, in die die Nachricht kopiert werden soll
edit-from	ESC f	Editiere das From-Feld
edit-file	^X e	Editiere die anzuhängende Datei
edit-headers	E	Editiere Nachricht (einschließlich Kopf)
edit-message	e	Editiere Nachricht
edit-mime	m	Editiere Anhang mittels Mailcap
edit-reply-to	r	Editiere Reply-To-Feld
edit-subject	s	Editiere Betreff dieser Nachricht (Subject)
edit-to	t	Editiere Empfängerliste (To)
edit-type	^T	Editiere Typ des Anhangs
filter-entry	F	Filtere Anhang mit Shell-Kommando
forget-passphrase	^F	Entferne PGP-Mantra aus Speicher
ispell	i	Rechtschreibprüfung via ispell
new-mime	n	Erzeuge neues Attachment via mailcap
pgp-menu	p	Zeige PGP-Optionen
pipe-entry		Bearbeite (pipe) Nachricht/Anhang mit Shell-Kommando
postpone-message	P	Stelle Nachricht zum späteren Versand zurück
print-entry	l	Drucke aktuellen Eintrag
rename-file	R	Benenne angehängte Datei um
send-message	y	Versicke Nachricht
toggle-unlink	u	Wähle, ob Datei nach Versand gelöscht wird
view-attach	EINGABE	Zeige Anhang, wenn nötig via Mailcap
write-fcc	w	Schreibe Nachricht in Mailbox

6.4.8 Nachricht zurückstellen (postpone)

delete-entry	d	Lösche aktuelle Nachricht
undelete-entry	u	Entferne Löschmoderung vom aktuellen Eintrag

6.4.9 Datei aus Liste auswählen (browser)

change-dir	c	Wechsle Verzeichnisse
check-new	TAB	Überprüfe Mailboxen auf neue Nachrichten
[Anmerkung: 2x TAB?]		
enter-mask	m	Gib Dateimaske ein
search	/	Suche mit regulärem Ausdruck
search-next	n	Suche nächsten Treffer
search-reverse	ESC /	Suche rückwärts nach regulärem Ausdruck
select-new	N	Wähle eine neue Datei in diesem Verzeichnis aus
sort	o	Sortiere Nachrichten
sort-reverse	O	Sortiere Nachrichten in umgekehrter Reihenfolge

toggle-mailboxes	TAB	Schalte zwischen Mailboxen und allen Dateien um
view-file	LEERZEICHEN	Zeige Datei an
subscribe	s	bestelle die aktuelle Mailbox (nur IMAP)
unsubscribe	u	bestelle die aktuelle Mailbox ab (nur IMAP)
toggle-subscribed	T	Schalte zwischen Ansicht aller und bestellter Mailboxen um (nur IMAP)

6.4.10 PGP

view-name	%	Zeige Nutzer-ID zu Schlüssel an
verify-key	c	Verifiziere öffentlichen PGP-Schlüssel

6.4.11 Editor

backspace	RÜCKTASTE	Lösche Zeichen vor dem Cursor
backward-char	^B	Bewege Cursor ein Zeichen nach links
backward-word	ESC b	Bewege Cursor zum vorhergehende Wort
bol	^A	Springe zum Zeilenanfang
buffy-cycle	LEERZEICHEN	Rotiere unter den Eingangs-Mailboxen
capitalize-word	ESC c	Schreibe das erste Zeichen im Wort groß
complete	TAB	Vervollständige Dateinamen oder Kurznamen
complete-query	^T	Vervollständige Adresse mit Abfrage
delete-char	^D	Lösche das Zeichen unter dem Cursor
eol	^E	Springe zum Ende der Zeile
forward-char	^F	Bewege Cursor ein Zeichen nach rechts
forward-word	ESC f	Bewege Cursor zum nächsten Wort
history-down	ungebunden	Gehe in der Liste früherer Eingaben nach unten
history-up	ungebunden	Gehe in der Liste früherer Eingaben nach oben
kill-eol	^K	Lösche bis Ende der Zeile
kill-eow	ESC d	Lösche vom Cursor bis Wortende
kill-line	^U	Lösche Zeile
kill-word	^W	Lösche Wort vor Cursor
quote-char	^V	Übernehme nächste Taste unverändert
transpose-chars	ungebunden	Tausche Zeichen unter Cursor mit vorherigem
upcase-word	ESC u	Schreibe alle Zeichen im aktuellen Wort groß

7 Diverses

7.1 Anerkennungen

Kari Hurttta <kari.hurttta@fmi.fi> hat damals in den ELM-ME Tagen den ursprünglichen Code zum MIME-Parsing mitentwickelt.

Die folgenden Menschen waren bei der Entwicklung von Mutt sehr hilfreich:

Vikas Agnihotri <vikasa@writeme.com>,
 Francois Berjon <Francois.Berjon@aar.alcatel-althom.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>,
Nickolay 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 Über dieses Dokument

Dieses Dokument wurde in SGML geschrieben und anschließend unter Verwendung des Pakets *sgml-tools* in das vorliegende Format konvertiert.

7.3 Über die vorliegende Übersetzung

Dieses Handbuch wurde in Teamarbeit ins Deutsche übersetzt. Als Übersetzer beteiligt waren (in alphabetischer Reihenfolge):

Matthias Friedrich <mafr@topmail.de>,
Thomas Schultz <tststs@gmx.de>,
Gottfried Schwieters <Gottfried.Schwieters@gmx.de>,

Stephan Seitz <*nur-ab-sal@gmx.de*>

Die Umstellung von Version 1.2.5 auf 1.4 haben Thomas Schultz und Stephan Seitz vorgenommen. Um die Gesamtleitung und anfallende kleinere Aktualisierungen kümmert sich Thomas Schultz.

An dieser Stelle möchten wir uns auch bei all denen bedanken, die unsere Übersetzung geduldig Korrektur gelesen haben. Fehler lassen sich natürlich trotzdem nicht ausschließen - wenn du einen findest, lass es uns bitte wissen, damit wir ihn in der nächsten Versionen beheben können.