

百万人のMutt

~Mutt活用講座~

滝澤陸史

taki@cyber.email.ne.jp

第11回 複数アカウントの設定



皆さんはメールアドレスのアカウントをいくつお持ちでしょうか？ 筆者の場合は、私的なもので2個、仕事で使っているもので1個あります。転送されて送られてくるアドレスも含めればさらに2、3個あります。今回は、このように複数のアカウントを使い分けている場合に、それらをMuttでどのように扱ったらよいかを紹介합니다。

本題に入る前に、Muttの最新状況をお知らせします。執筆時点での最新版は1.3.27iです。これは1.4の最終ベータ版となる予定です。Muttの開発者たちの間で再現できないバグが1つあり、1.4の公開の障害となっていたのですが、それもなんとか修正されました。1.4が公開されない理由はもうないはずですが、その前に「念のため」ということで1.3.28iが出るかもしれません。一方、開発系のブランチとして1.5.0がすでに始まっていて、こちらにはS/MIMEの機能が含まれています。執筆時点ではFTPサイトには1.5.0は置かれておらず、anonymous CVSからのみ取得可能となっています。

【表1】\$HOME/.muttディレクトリのファイル構成

ファイル名	内容
foo	fooのアカウント用
bar	barのアカウント用
common	共通の設定

【リスト1】使用する転送アドレスを指定する

```
set alternates="~foo@example.org$"
```

【リスト3】設定ファイルfooの例

```
source ~/.mutt/common
source /usr/local/doc/mutt/samples/mutt-ja.rc
# name and address
set realname="Taro Itsutsubashi"
set from="foo@example.org"
set hostname="example.org"
# folder
set spoolfile=~/.Maildir/foo
set folder=~/.Mail/foo
```

🦷 転送アドレス

まずはウォーミングアップとして転送アドレスを処理する例を紹介します。

ここで扱う転送アドレスは、身元を隠すための怪しいアドレスではなく、メインで使うメールアドレスを変更した際に旧アドレスにきたメールを新アドレスに転送するように設定したものと、サーバの管理をする際にpostmasterやwebmaster宛にきたメールをエイリアスで自分のアドレスに転送するように設定したものです。そのため、転送されたメールに返信する際に、原則として転送元のアドレスを差出人(From)としては用いないこととします。

\$alternates

転送されたメッセージを読む分には、特にMuttで設定することもありません。しかし、そのままでは転送アドレスが自分のアドレスであることをMuttは認識しないため、インデックス画面における\$to_charsで設定できるフラグやパターン修飾子の「~P」(大文字のP:あなたからのメッセージ)と「~p」(小文字のP:あなた宛のメッセージ)には使えません。そのため、転送アドレスが自分のアドレスであることをMuttに認識させるために、設定変数\$alternatesにアドレスを指定する必要があります。

ります。\$alternatesの設定方法は次の例のようにメールアドレスを記述します。

```
set alternates="foo@example.org"
```

ここで記述するのは正規表現なので厳密にはリスト1のように記述した方が間違いはないでしょう。複数のアドレスを指定する場合にはリスト2の例のように記述します。

🦷 設定ファイルの使い分け

複数アカウントを使い分ける方法として、最も簡単なのは「アカウントごとに設定ファイルを用意する方法」です。設定ファイルを用意したら、Muttの起動オプション「-F」で使用するアカウント用の設定ファイルを指定すればよいでしょう。ただし、別のアカウントのメールを読みたい場合にはその都度Muttを起動し直す必要があります*1。

ここで「foo」と「bar」という2つのアカウントを使い分ける例を紹介します*2。表1に示すようにfooとbar用の設定ファイルを用意します。この2つの設定ファイルにはそれぞれ名前やメールアドレスなどの情報やメールを格納するディレクトリなどの情報を記述します。必要に応じてPOP/IMAPサーバの設定も記述します。このときに共通の設定はcommonというファイルに記述し、fooとbarの両方からsourceコマンドでcommonを読み込むようにします。

【リスト2】複数の転送アドレスを使用する場合の設定

```
set alternates="(~foo@example.org$|^bar@example.net$)"
```

【リスト4】設定ファイルbarの例

```
source ~/.mutt/common
source /usr/local/doc/mutt/samples/mutt-en.rc
# name and address
set realname="Taro Itsutsubashi"
set from="bar@example.net"
set hostname="example.net"
# folder
set spoolfile=~/.Maildir/bar
set folder=~/.Mail/bar
```

*1 文字符号化方式が同じであれば、他のアカウントのメールもフォルダを開けば読むことができます。しかし、返信を行うときにアカウントが異なるためFromやFccが現在のアカウントのものが適応されることに注意してください。 *2 本誌2001年7月号の連載記事のコラム「Muttも歩けば棒に当たる」で紹介した設定ファイルの使用例を修正したものです。

リスト3～リスト5に、設定ファイルfoo、bar、commonの例をそれぞれ示します。後は次のように設定ファイルを指定してMuttを起動すればよいでしょう。

```
mutt -F ~/.mutt/foo
```

起動時にロケールや環境変数を設定し直したり、新たにターミナルを起動したりする場合は、リスト6のようなシェルスクリプトを作ってMuttを起動します。

フックコマンドの活用

設定ファイルを使い分ける方法では異なるアカウントのメールを読む場合にはMuttを起動し直さなければなりません。どうせなら、Muttを1度起動したらそのまま他のアカウントのメールも扱えた方が便利です。そこで、フックコマンドを活用することで1つの設定ファイルで複数アカウントを使う方法を紹介합니다。なお、フックコマンドに関しては本誌2001年9月号の記事「第3回：フック」をご覧ください。

フォルダ構成

複数のアカウントを使い分けるには、アカウントごとにフォルダを用意してfolder-hookで設定を変えるのが簡単でしょう。そのため、fooとbarという2つのアカウントに対して図1のようなフォルダ構成にしてみます。

メールアドレス等の設定

名前は各アカウントで共通なので次のように通常の設定を行います。

```
set realname="Taro Itsutsubashi"
```

【リスト5】設定ファイルcommonの例

```
# folder
set mbox_type=Maildir
set mbox_type=Maildir
set folder=~ /Mail
set tmpdir=~ /tmp
set postponed=+postponed
set alias_file=~ /.mutt/alias"
source ~/.mutt/alias
source ~/.mutt/color
```

【リスト6】mutt-fooスクリプト

```
#!/bin/sh
LANG=ja_JP.eucJP
COLORFGBG="white:black"
export LANG COLORFGBG
mlterm -f white -b black -g 80x40 -E euc-jp \
-e mutt -F ${HOME}/.mutt/foo "$@" &
```

メールアドレスに関してはアカウントごとに異なるためfolder-hookで使い分けます。ここでは、デフォルトではfooの設定が適用されるようにし、barの下のフォルダを開いたときにbarの設定が適用されるようにします。

```
folder-hook . 'set \
from="foo@example.org" \
hostname="example.org"'
folder-hook bar 'set \
from="bar@example.net" \
hostname="example.net"'
```

フォルダの設定

まず、各アカウントで共通であるフォルダの設定を次のように行います。

```
set mbox_type=Maildir
set folder=~ /Mail
set tmpdir=~ /tmp"
set postponed=+postponed"
```

次に\$spoolfileの設定を行います。デフォルトではfooのスプールに設定し、barの下のフォルダを開いているときにはbarのスプールに設定します。

```
folder-hook . \
"set spoolfile=~ /Maildir/foo"
folder-hook bar \
"set spoolfile=~ /Maildir/bar"
```

このfolder-hookの設定のみだと、Muttの起動時にはスプールが設定されていないことになるので、Muttのデフォルトの/var/spool/mailを開こうとします。これを防ぐために起動時に開くスプールの設定を行います。ここでは、次のようにfooのスプールに設定します。

```
set spoolfile=~ /Maildir/foo
```

残りの\$mboxと\$recordの設定を行います。

デフォルトではfooのフォルダを設定し、barのフォルダを開いたときにはbarのフォルダを設定します。

```
folder-hook . 'set \
mbox=+foo/mbox \
record=+foo/record'
folder-hook bar 'set \
mbox=+bar/mbox \
record=+bar/record'
```

なお、folder-hookを使わずに次のようにmbox-hookとfcc-hookを使っても同じ動作をするようになります。ただし、\$mboxや\$recordが設定されているわけではないため、「>」(\$mboxファイルの参照)、「<」(\$recordファイルの参照)といったショートカットキーが使えなくなります。

```
mbox-hook Maildir/foo =foo/mbox
mbox-hook Maildir/bar =bar/mbox
fcc-hook "~f foo@example.org" \
foo/record
fcc-hook "~f bar@example.net" \
bar/record
```

メールボックスの設定

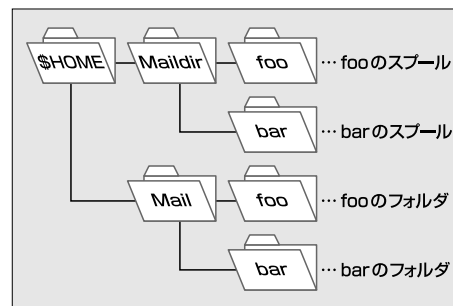
procmailやmaildropなどで振り分けを行っている場合には、振り分け先のフォルダをmailboxesコマンドで指定します。これはアカウントごとに使い分ける必要がないため、すべてのアカウントのフォルダを指定しましょう。

```
mailboxes 'echo $HOME/Maildir/*' \
'echo $HOME/Mail/foo/list/*' \
'echo $HOME/Mail/bar/list/*'
```

文字符号化方式の設定

使用する複数のアカウントがすべて同じ文字

【図1】フォルダ構成



【リスト7】送信用プログラムの設定

```
set sendmail="/usr/bin/nullmailer-inject"
```

【リスト8】メール送信時に実行されるコマンド

```
nullmailer-inject -f foo@example.org
```

符号化方式を用いるのであれば、アカウントごとに使い分ける必要はないため、通常の設定通り行います。

アカウントごとにさまざまな文字符号化方式を用いる場合には、先月号(2002年4月号)の

記事「第10回:多言語メール環境」を構築してください。

送信の設定

複数アカウントが使える送信用のプログラム

【リスト9】アカウント別の署名を設定する設定

```
folder-hook . \
set signature=~/.mutt/signature-foo'
folder-hook bar \
set signature=~/.mutt/signature-bar'
```

【リスト10】POP3サーバから直接メッセージを取得する設定

```
folder-hook . ' \
et pop_host=pop://foo@pop.example.org \
et pop_authenticators="apop"
folder-hook bar ' \
et pop_host=pop://bar@pop.example.net \
et pop_authenticators="apop"
```

【リスト11】アカウント切り替えマクロの例

```
macro index \cab ":set \
from=foo@example.org \
hostname=example.org \
mbox=+foo/mbox \
record=+foo/record \
signature=~/.mutt/signature-foo\n"
macro index \caf ":set \
from=bar@example.net \
hostname=example.net \
mbox=+bar/mbox \
record=+bar/record \
signature=~/.mutt/signature-bar\n"
```

を \$sendmail に設定します。ここでは nullmailer を使うことにします(リスト7)。

次に、\$sendmail で指定したプログラムに「-f」オプションで envelope from を指定するようにする設定を行います。

```
set envelope_from=yes
```

このように設定することにより、fooのアカウントであれば、メールの送信時にリスト8に示すコマンドが実行されます。逆に、この設定を行わないと、アカウントごとの envelope from が付かなくなるため、アカウントの使い分けはできなくなります。

署名の設定

アカウントごとに異なる署名を付ける場合には、リスト9のように folder-hook を設定します。

POP3の設定

POP3サーバから直接メッセージを取得する場合には、リスト10のように folder-hook を設定します。

アカウント切り替えマクロ

他のアカウントのメッセージを転送しようとする場合、そのメッセージのあるフォルダを開くと、folder-hookによって、アカウントの設定もそのフォルダのものに変わってしまいます。そのため、元のアカウントに戻す設定を行いたいところですが、手で設定するのは大変なので、アカウントの設定を変えるマクロを用意します(リスト11)。このリストの例では「Ctrl+A」「b」でbarの設定に切り替わり、「Ctrl+A」「f」でfooの設定に切り替わります。このアカウントの設定変更は、次にフォルダを変更して folder-hook が実行されるまで有効です。



最後に

連載コラムの「Muttも歩けば棒に当たる」は今回からお休みをいただくことになりました。とはいえ、不定期ながら継続はしていきたいと考えています。

Resource

- Mutt Japanese Edition
<http://www.emailab.org/mutt/>

おまけのTips 設定コマンドの補完機能

Muttでファイル名やメールアドレスを入力する際に、Tabキーを押すと、ファイル名やエイリアスの補完機能が働きます。このことは皆さんご存知かと思えます。

その一方、あまり知られていないのが、設定コマンドの補完機能です。この機能は、マニュアルにも載っていない*3事項なので、ここで紹介します。

「:(機能enter-command)を入力すると、設定コマンドの入力モードになりますが、このときTabキーを押すと、設定コマンドや設定変数の補完機能が働きます。設定変数名の綴りをはっきりとは覚えていなかったり、長い設定変数名を入力したりするとき役に立つでしょう。

例えば、Muttの起動中に「設定変数 \$pager_index_linesの値を変えたい」と思ったとします。このときには、まずインデックスやページで、

```
:set pager
```

と入力します。次にTabキーを数回押して、目的のものが出てくるのを待ちます。

*3 筆者が読み落としていなければ、

```
:set pager_index_lines
```

後は次のように値を入力し、Enterキーを押せば設定されます。

```
:set pager_index_lines=5
```

なお、設定変数名の接頭語として「?」を付けると、現在設定されている値が表示されますので、設定変数の値を調べるときにもこの機能は役に立つでしょう。先ほど示した例でいえば、次のようにまず入力します。

```
:set ?pager
```

同じく、Tabキーを数回押して目的のものが出てくるのを待ちます。

```
:set ?pager_index_lines
```

Enterキーを押せば、現在の値が表示されます。

```
pager_index_lines=10
```

(滝澤隆史)