NAME

readmsg - extract messages from a mail folder

SYNOPSIS

readmsg [**-anhp**] [**-f** folder] [selection ...]

DESCRIPTION

Readmsg extracts selected mail messages from a mail folder. One helpful use of the program is when you are composing a response to a mail message in an external editor. In this case, you can run *readmsg* to pull a copy of the original message into the editing buffer. When you run *readmsg* from within *elm* (e.g. from a subshell escape or in an external editor while composing a mail reply) it behaves a bit differently from when you run it directly (e.g. from a shell command line). We will first describe its normal behavior, and then describe how it is different when you run it under *elm*. You tell *readmsg* which messages to extract with the *selection* argument. There are a couple of possible different ways to specify the *selection*.

1. A lone ‘‘*’’ means select all messages in the mailbox.

2. A list of message numbers may be specified. Values of ‘‘0’’ and ‘‘$’’ in the list both mean the last message in the mailbox. For example:

   ```
   readmsg 1 3 0
   ```

   extracts three messages from the folder: the first, the third, and the last.

3. Finally, the *selection* may be some text to match. This will select a mail message which *exactly* matches the specified text. For example,

   ```
   readmsg staff meeting
   ```

   extracts the message which contains the words ‘‘staff meeting.’’ Note that it will *not* match a message containing ‘‘Staff Meeting’’ the matching is case sensitive. Normally only the first message which matches the pattern will be printed. The **–a** option discussed in a moment changes this. The **-f** flag indicates that you’d rather use the folder specified rather than the default incoming mailbox. The specified *folder* can be a filename or a specification such as ‘‘=sentmail’’. The **-h** flag instructs the program to include the entire header of the matched message or messages when displaying their text. (default is to display the From: Date: and Subject: lines only) The **-n** flag instructs the program to exclude *all* headers. This is used mostly for extracting files mailed and such. The **-p** flag indicates that the program should put form-feeds (control-L) between message headers. The **-a** flag indicates that all messages which match the pattern specified on the command line should be printed, not just the first. If a pattern was not specified on the command line then this flag has no effect. When you run *readmsg* under *elm* (once again, say in the context of an external editor) the behavior will be different from that described above as follows.

1. The default mail folder will be the folder you are currently examining in *elm* and not necessarily your incoming mail folder.

2. You do not need to specify a *selection* on the command line. If you omit the *selection* then *readmsg* will extract the message(s) you have selected in Elm. If you have tagged any messages then this would be all of the tagged messages, otherwise it would be the message you are currently examining.

3. Normally the message numbers *readmsg* uses are in mailbox order. When you call *readmsg* under *elm* and do not override the folder selection with the **–f** option, then message numbers will be sorted as they are displayed on the *elm* message index screen.

EXAMPLES

First off, to use this from within **vi** to include the text of the current message, you could use the command:

:r !readmsg

(as you hit the ':' the editor will put you at the bottom of the screen with the ':' prompt). The space following ':r' is required.

Let's look at something more interesting, however;

Suppose you have the mail file;

```
From joe Jun 3 1986 4:45:30 MST
Subject: hello

Hey Guy!  Wanta go out and have a milk this evening?

Joe

From john Jun 3 1986 4:48:20 MST
Subject: Dinner at Eight
From: John Dinley <xyz!john>

Remember you should show up about eight, okay?

                        - John D -

From xxzyz!cron Jun 3 1986 5:02:43 MST

Cannot connect to server: blob
Job 43243 deleted from queue.
```

The following commands will result in;

```
$ readmsg 2
[ display the second message, from John ]

$ readmsg
[ an error, unless we're calling from elm ]

$ readmsg BLOB
[ no match - case sensitive! ]

$ readmsg -h connect to server
[ displays third message, including headers ]
```

FILES

/usr/mail/<username>	The incoming mail
$ELMSTATE	Status information from *elm*

AUTHOR

Elm Development Group

SEE ALSO

newmail(1L), elm(1L)

BUGS

The '*' metacharacter doesn't always work as expected!
Perhaps the pattern matching should be case insensitive?

It might be confusing that messages are sorted when running under *elm* with the current folder, and in mailbox order for all other cases.

BUG REPORTS TO

Syd Weinstein elm@DSI.COM (dsinc!elm)

COPYRIGHTS