NAME

phimail – Paul's xbiff replacement

VERSION

Version 1.1 23 February 1997

SYNOPSIS

phimail [standard X options] [options]

DESCRIPTION

phimail is flexible mail alert tool in the spirit of xbiff. Most aspects of it are configurable, and since it's a script you can always copy the script and configure it as much as you like.

I used to call it 'pmail', but that turned out to conflict with the name of an existing application.

Left button in window calls mail reader.
Right button in window pops up mailbox info.
Middle button in window tells about phimail.
Ctrl-C or 'q' in window quits.

You can get the default options at your site with 'phimail -blah'

OPTIONS

-mf <mail file>

Specify which mail file to report on, the default is typically /var/spool/mail/yourname

-rf <rc file>

Specify which file to read on startup for options, the default is ~yourname/.phimailrc. This file is actually sourced by phimail, so this file must be in Tcl syntax. The default file is read before any command-line options are set, so command-line options override settings in the .phimailrc. If -rf <file> is specified, that file is sourced at that, as the command-line options are processed. So, any command-line options specified before the -rf option can be overridden by the specified file, but any command-line options after the -rf option will override anything previously set.

The rc file can be re-read by hitting 'r' in the phimail window.

Here's an example .phimailrc showing all current settable options:

```
#
# .phimailrc
#
set mailFile /var/spool/mail/[exec whoami]
set rcFile ~/.phimailrc
set cmdFile ~/.phimailcmd
set flagDownPic ~/lib/pics/flagdown_sm.gif
set flagUpPic ~/lib/pics/flagup_sm.gif
set boxOpenPic ~/lib/pics/boxopen_sm.gif
set mailCommand "xterm -e /usr/local/bin/pine -i"
set soundCommand ""
set beepOn 1
set mailCommand "xterm -e /usr/local/bin/pine -i"
set checkDelay 60
set globalBells 0
set fullParse 1
```

```
set flagUpUnread 1
set fancyBell 1
set palette ""
```

-cf <cmd file>

Specify which file of Tcl commands to execute upon reception of new mail. The purpose of this file was originally to allow the user to do regular expression matching in the phimail global variables lastFrom and lastSubject to match arbitrary actions (i.e., playing a specific sound file) with mail from particular people or with a particular subject. The default is ~yourname/.phimailcmd.

The cmd file can be forced to execute by hitting 'c' in the phimail window. This is useful for testing/debugging your command file.

Here's an example .phimailcmd showing some useful commands:

```
############################################################
# This is a file of commands that is sourced when
# new mail is received.
#
# The intent was originally to do regular-expression matching
# on the lastFrom and lastSubject variables to customize phimail's
# response to new mail from different people and with different
# subjects.
#
# You have the lastFrom and lastSubject variables
# to work with.  You may want to echo them
# if your regular expressions aren't working properly.
#
# You can define your own variables, if you need.  Just avoid
# stepping on any variables used by phimail itself.
#
# For more information on the regexp command of Tcl, do a
# 'man regexp'.  You should get all the info you need, assuming
# your system is set up correctly.
############################################################

# echo variables for debugging
#puts $lastFrom
#puts $lastSubject

############################################################
# some regular expression matching examples
############################################################

# here's a good one for playing a custom sound
# file for the from person
set personFrom ""
if {[regexp (\[a-zA-Z\])(@) $lastFrom match personFrom]
    || [regexp (\[a-zA-Z\]*) $lastFrom match personFrom]} {
#   puts "message received from '$personFrom'"
  if {[file exists $personFrom.au]} {exec cat $personFrom.au > /dev/audio}
}
```

```
# the most basic match for the person who sent the message
if {[regexp (.*my_friends_name.*) $lastFrom]} {
#   puts "message received from my_friends_name
  exec cat my_friends_name.au > /dev/audio
}

# a special alert for messages marked 'urgent'
if {[regexp -nocase (.*urgent.*) $lastSubject]} {
#   puts "urgent message!!!"
  exec cat alert.au > /dev/audio
}

# a message for all messages from a specific domain
if {[regexp (.*.caltech..*) $lastFrom]} {
#   puts "message received from caltech"
  exec cat caltech.au > /dev/audio
#   exec rsh some_machine "cat caltech.au > /dev/audio"
}
```

-fdp <flagdown picture>

Which picture to show when no mail is present (or no new mail is present).

-fup <flagup picture>

Which picture to show when mail is present (or new mail is present).

-bop <boxopen picture>

Which picture to show when mail is being read.

-mc <mail reader command>

What command to call to read mail. If you use pine, for example, you might want to try:

xterm -e pine -i

For users of elm you may want to try something like:

xterm -geometry 80x36 -fn 10x20 -n Elm -T Elm -rv -cr cyan -e elm

For users of netscape mail, I can't help you. I thought the -remote option for netscape would provide a way, but there doesn't seem to be a way to bring up the mail window via a remote netscape command. If anyone knows how, let me know.

For users of regular old mail, we can only hope your PDP-11 lasts one more year. Try this:

xterm -e mail

-t <check delay, in secs>

How often to check the mail file for new mail.

-sc <new mail sound command>

What command to execute to play a sound when new mail arrives. For standard audio files, for example, try

cat holler.au > /dev/audio

Using a sound file disables standard beeping.

-beepOn <0,1>

Whether to beep when new mail arrives.

-globalBells <0,1>

Some people (i.e., myself) prefer bells disabled in general (with the 'xset b 0' command). However, I still want mail to beep. If you are of a similar mind, use -globalBells 0. If you don't disable bells globally, yet run with -globalBells 0, be aware that phimail will end up globally disabling bells for you.

-fancyBell <0,1>

Whether to play a sequence of bells, not just one. Eventually it might be nice to have a selection of possible tunes.

-fullParse <0,1>

Do a full parse of the mail file headers to gather info on unread messages, etc. This is the default. You may want to disable the full parse if you have a _very_ large mail file or phimail is somehow failing when parsing your mail file (it shouldn't, but it is one of the more platform-finicky parts of the code.) With -fullParse 0, phimail just uses time stamps to decide whether new mail is in or not. In addition, -fullParse 0 disables the mail box info pop-up, since none of that information is then known.

-flagUpUnread <0,1>

Whether you want the flagUp picture shown when you have unread mail. Otherwise, flagUp is only shown when you have new mail.

-palette <reds/greens/blues>

How many shades of red/green/blue you want to limit yourself to in the colormap. This is useful for reducing color usage in 8-bit displays. Try something like 5/5/4 or 6/6/5 if phimail is eating up too many colors in your colormap. I leave less shades of blue since the human eye is less sensitive to blues. If phimail doesn't seem to be getting enough colors even though you have enough colors (on a 24-bit display, for example), try '-visual truecolor' when running phimail. This is a standard wish option, and you can find out about other -visual choices with a 'man wish'.

-blah

Print out command line options and exit. (Unfortunately, wish grabs the -h or -help, so just type any illegal option to get help on usage.)

The standard X options are grabbed and processed by phimail, and aren't passed to phimail itself. Do 'man wish' to find out about these options, since they are common to all wish scripts.

standard X options:
-colormap: Colormap for main window
-display: Display to use
-geometry: Initial geometry for window

-name: Name to use for application
-sync: Use synchronous mode for display server
-visual: Visual for main window
--: Pass all remaining arguments through to script
-h:
-help:

COPYLEFT

AUTHOR

Paul LeMahieu
lemahieu@paradise.caltech.edu
http://www.paradise.caltech.edu/~lemahieu

SEE ALSO

wish(1), xbiff(1)

BUGS

phimail doesn't resize the image, but rather takes on the size of the mailbox image used.

The pop-up menu used to display the mailbox information doesn't line things up nicely.

phimail currently uses perl to preprocess the mailbox. This isn't bad, but it would probably be nicer to do everything strictly in tcl.