
PROMPTREMINDERS

To be periodically reminded of things

By: JonL White

Revised by: Larry Masinter (Masinter.pa@Xerox.com), subsequently by Becky Burwell
(Burwell.PA@Xerox.COM)

INTRODUCTION

PROMPTREMINDERS implements a facility which schedules events to be performed, or messages to be flashed in a prompt window. Events can be periodic or once-only. The showing of a message in a prompt window has the extra facility of flashing a message, and stopping only when there has been a recent response (mouse or keyboard movement) from the user.

If the MESSAGE given for the reminder (see description of the function SETREMINDER below) is a listp, then when the reminder "goes off", that listp will be EVAL'd rather than any of the "winking", "flashing", or "hassling" mentioned above.

The global variable REMINDERSTREAM holds the stream where the message is to be displayed; if not set by the user, it defaults to PROMPTWINDOW. After the message has been displayed, the window (if indeed REMINDERSTREAM holds a window) will be closed, depending on the value of CLOSEREMINDERSTREAMFLG.

REMINDERS is a file package type, so that they may be easily saved on files, and so that the general typed-definition facilities may be used. On any file which uses the REMINDERS filepkgcom, it is advisable to precede this command with a command

```
(FILES (SYSLOAD COMPILED FROM LISPUSERS) PROMPTREMINDERS)
```

since this package is not in the initial Lisp loadup. When initially defining a reminder, it is preferable for the user to call SETREMINDER rather than PUTDEF; but HASDEF is the accepted way to ask if some name currently defines a "reminder", and DELDEF is the accepted way to cancel an existing "reminder".

EXAMPLES

```
(SETREMINDER NIL (ITIMES 30 60) "Have you done a CLEANUP recently?")
```

the user wants to be reminded every 30 minutes that he ought to save his work

```
(SETREMINDER 'WOOF NIL "Call home about dinner plans."  
"8-Jan-83 4:00PM")
```

he merely wants to be told once, at precisely 4:00PM to call home

```
(SETREMINDER NIL 600
  ' (PROGN (AND (FIND.PROCESS 'LISTFILES) (add FREQ 1))
    (add TOTAL 1)))
```

checks every 10 minutes to see if there is a process called LISTFILES

FUNCTIONS

(SETREMINDER NAME *PERIOD MESSAGE INITIALDELAY EXPIRATION*) [Function]

This will create and install a "reminder" with the name NAME (NIL given for a name will be replaced by a gensym), which will be executed every PERIOD number of seconds by winking the string MESSAGE into the prompt window; if MESSAGE is null, then NAME is winked; if MESSAGE is a listp, then it is EVAL'd and no "winking" takes place. "Winking" means alternately printing the message and clearing the window in which it was printed, at a rate designed to attract the eye's attention.

The first such execution will occur at PERIOD seconds after the call to SETREMINDER unless INITIALDELAY is non-NIL, in which case that time will be used; a numeric value for INITIALDELAY is interpreted as an offset in seconds from the time of the call to SETREMINDER, and a stringp value is an absolute date/time string.

If PERIOD is null, then the reminder is to be run precisely once. If EXPIRATION is non-null, then a fixp means that that number of seconds after the first execution, the timer will be deleted; a stringp means a precise date/time at which to delete the timer.

Optional 6th and 7th arguments -- called REMINDINGDURATION and WINKINGDURATION -- permit one to vary the amount of time spent in one cycle of the wink/flash loop, and the amount of time spent winking before initiating a "flash". The attention-attracting action will continue for REMINDINGDURATION seconds (default: the value of the global variable DEFAULT.REMINDER.DURATION which is initialized to 60), or until some keyboard action takes place.

Type-ahead does not release the winking. In case the user fails to notice the winking, then every WINKINGDURATION seconds (default: the value of the global variable DEFAULT.REMINDER.WINKINGDURATION which is initialized to 10) during the "reminding", the whole display videocolor will be wagged back and forth a few times, which effects a most obnoxious stimulus.

SETREMINDER returns the name (note above when NIL is supplied for the name).

(ACTIVEREMINDERNAMES) [Function]

ACTIVEREMINDERNAMES returns the list of active reminders.

(REMINDER.NEXTREMINDDATE *NAME DATE*) [Function]

REMINDER.NEXTREMINDDATE returns (and optionally sets) the date&time (in DATE format) at which the reminder is next to be executed.

(REMINDER.EXPIRATIONDATE *NAME DATE*) [Function]

REMINDER.EXPIRATIONDATE returns (and optionally sets) the date&time (in DATE format) at which the reminder will be automatically deleted.

(REMINDER.PERIOD *NAME SECONDS*)

[Function]

REMINDER.PERIOD returns (and optionally sets) the period (in seconds) at which the reminder gets rescheduled.

(SHOWDEF *name* 'REMINDERS)

[Function]

will show a reminder in a pretty format, etc.