File created: 13-May-88 14:13:53 **{ERINYES}<LISPUSERS>MEDLEY>PROMPTREMINDERS.;1**

previous date: 18-Nov-85 13:39:10 {ERINYES}<LISP>KOTO>LISPUSERS>PROMPTREMINDERS.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ PROMPTREMINDERSCOMS
       [(DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS PERIODIC.PROMPT.REMINDER))
        (FNS SETREMINDER SHOW.REMINDER ACTIVEREMINDERNAMES REMINDER.NEXTREMINDDATE REMINDER.EXPIRATIONDATE
             REMINDER.PERIOD REMINDERS.RESTART REMINDERS.WATCHDOG)
        (PROP ARGNAMES REMINDER.NEXTREMINDDATE REMINDER.EXPIRATIONDATE REMINDER.PERIOD)
        (FNS \PUTREMINDER \GETREMINDER \DELREMINDER)
        (FNS PERIODICALLYCHECKREMINDERS)
        (INITVARS (\PR.REMOVALS NIL))
        (INITVARS (DEFAULT.REMINDER.DURATION 60)
               (DEFAULT.REMINDER.WINKINGDURATION 10)
               (PERIODIC.PROMPT.REMINDERS NIL)
               (REMINDERSTREAM PROMPTWINDOW))
        (GLOBALVARS \PR.REMOVALS REMINDERSTREAM PERIODIC.PROMPT.REMINDERS DEFAULT.REMINDER.DURATION
               DEFAULT.REMINDER.WINKINGDURATION DEFAULT.REMINDER.PERIOD)
        (FILEPKGCOMS REMINDERS)
        (INITVARS (CLOSEREMINDERSTREAMFLG))
        [VARS (\REMINDER.EVENT (CREATE.EVENT 'PERIODIC.PROMPT.REMINDERS]
        (GLOBALVARS CLOSEREMINDERSTREAMFLG \REMINDER.EVENT)
        (ADDVARS (AFTERLOGOUTFORMS (REMINDERS.RESTART)))
        (P (REMINDERS.RESTART))
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                             (NLAML)
                                                                             (LAMA REMINDER.PERIOD
                                                                                   REMINDER.EXPIRATIONDATE
                                                                                   REMINDER.NEXTREMINDDATE])

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

[TYPERECORD PERIODIC.PROMPT.REMINDER (REMINDER.TIMEOUTBOX REMINDER.MESSAGE REMINDER.PERIOD
                                            REMINDER.WINKINGDURATION REMINDER.DURATION REMINDER.FINALTIME
                                            REMINDER.NAME)
       REMINDER.TIMEOUTBOX _ (SETUPTIMER 0 'SECONDS)
       (ACCESSFNS ([REMINDER.NEXTREMINDDATE [GDATE (AND (fetch REMINDER.TIMEOUTBOX of DATUM)
                                                          (ALTO.TO.LISP.DATE (fetch REMINDER.TIMEOUTBOX
                                                                                    of DATUM]
                          (replace REMINDER.TIMEOUTBOX of DATUM with (SETUPTIMER.DATE NEWVALUE
                                                                             (fetch REMINDER.TIMEOUTBOX of DATUM]
                   (REMINDER.EXPIRATIONDATE [GDATE (AND (fetch REMINDER.FINALTIME of DATUM)
                                                         (ALTO.TO.LISP.DATE (fetch REMINDER.FINALTIME
                                                                                   of DATUM]
                          (replace REMINDER.FINALTIME of DATUM with (AND NEWVALUE (SETUPTIMER.DATE NEWVALUE]
)
)

(DEFINEQ

(SETREMINDER
  [LAMBDA (NAME PERIOD MESSAGE INITIALDELAY EXPIRATION REMINDINGDURATION WINKINGDURATION)
                                                                (* lmm "23-Apr-85 14:06")
    (PROG ((RNAME (OR NAME (GENSYM)))
           REMINDER)
          [if [NULL (SETQ REMINDER (GETDEF RNAME 'REMINDERS NIL 'NOERROR]
              then

          (* Big time delay before first "reminding" to allow time for the completion of this function!)

                  (SETQ REMINDER (create PERIODIC.PROMPT.REMINDER
                                        REMINDER.TIMEOUTBOX _ (SETUPTIMER 16000 NIL 'SECONDS]
          (replace REMINDER.MESSAGE of REMINDER with (OR MESSAGE RNAME))
          (replace REMINDER.DURATION of REMINDER with (OR (FIXP REMINDINGDURATION)
                                                           DEFAULT.REMINDER.DURATION))
          (replace REMINDER.WINKINGDURATION of REMINDER with (OR (FIXP WINKINGDURATION)
                                                                  DEFAULT.REMINDER.WINKINGDURATION))
          (PUTDEF RNAME 'REMINDERS REMINDER)

          (* This call is made even for "old" reminders, to get the action of MARKASCHANGED * Note also how the PERIOD was null
          during this time, so that it didn't reset the timer.)

          (replace REMINDER.PERIOD of REMINDER with PERIOD)
          [replace REMINDER.TIMEOUTBOX of REMINDER with (if (STRINGP INITIALDELAY)
                                                             then (SETUPTIMER.DATE INITIALDELAY
                                                                       (fetch REMINDER.TIMEOUTBOX of REMINDER))
                                                         else (SETUPTIMER (OR (FIXP INITIALDELAY)
                                                                              PERIOD 0)
                                                                    (fetch REMINDER.TIMEOUTBOX of REMINDER)
                                                                    'SECONDS]
          (if EXPIRATION
              then (REMINDER.EXPIRATIONDATE REMINDER (if (FIXP EXPIRATION)
                                                           then (IPLUS EXPIRATION (IDATE (
                                                                                         REMINDER.NEXTREMINDDATE
                                                                                          REMINDER)))
                                                         else (STRINGP EXPIRATION))
                          T))
          (NOTIFY.EVENT \REMINDER.EVENT)
          (RETURN RNAME])

(SHOW.REMINDER
  [LAMBDA (REMINDER)                                                   (* lmm "19-Apr-85 18:04")
    (PROG ((MESSAGE (fetch REMINDER.MESSAGE of REMINDER)))
          [if (LISTP MESSAGE)
              then (NLSETQ (EVAL MESSAGE))
            else (PRINTBELLS)
                 (DSPRESET REMINDERSTREAM)
                 (bind (FIRSTTIME _ T)
                       (LUACTION _ (COPYALL \LASTUSERACTION))
                       (VISIBLE _ NIL)
                       (DURATION _ (SETUPTIMER (fetch REMINDER.DURATION of REMINDER)
                                          NIL
                                          'SECONDS))
                    repeatuntil (TIMEREXPIRED? DURATION 'SECONDS)
                    do (bind (WINKING _ (SETUPTIMER (fetch REMINDER.WINKINGDURATION of REMINDER)
                                               NIL
                                               'SECONDS))
                          repeatuntil (TIMEREXPIRED? WINKING 'SECONDS)
                          do (if (SETQ VISIBLE (NOT VISIBLE))
                                  then (PRIN3 MESSAGE REMINDERSTREAM)
                                       (TERPRI REMINDERSTREAM)
                                       (if (NOT (EQUAL LUACTION \LASTUSERACTION))
                                           then (GO DONE))
                                else (DSPRESET REMINDERSTREAM))
                              (DISMISS 500]
      DONE])

(ACTIVEREMINDERNAMES
  [LAMBDA NIL                                                          (* JonL "29-NOV-82 16:58")
    (MAPCAR PERIODIC.PROMPT.REMINDERS (FUNCTION CAR])

(REMINDER.NEXTREMINDDATE
  [LAMBDA N                                                            (* lmm "19-Apr-85 18:07")

          (* * 1-arg case is only asking for information; multi-arg for update;
          3'rd arg, if non-null, says don't mark as changed.)

    (AND (IGEQ N 1)
         ([LAMBDA (DEF)
            (AND DEF ([LAMBDA (SDATE NEWDATE)
                        (if (EQ N 1)
                            then SDATE
                          else [SETQ NEWDATE (OR (STRINGP (ARG N 2))
                                                 (GDATE (ARG N 2]
                               (if [AND (NOT (IEQP (IDATE SDATE)
                                                   (IDATE NEWDATE)))
                                        (OR (ILEQ N 2)
                                            (NULL (ARG N 3]
                                   then (MARKASCHANGED (fetch REMINDER.NAME of DEF)
                                               'REMINDERS
                                               'CHANGED))
                               (replace REMINDER.NEXTREMINDDATE of DEF with NEWDATE)
                               (NOTIFY.EVENT \REMINDER.EVENT)
                      (fetch REMINDER.NEXTREMINDDATE of DEF]
          (if (type? PERIODIC.PROMPT.REMINDER (ARG N 1))
              then (ARG N 1)
            else (GETDEF (ARG N 1)
                         'REMINDERS NIL '(NOERROR NOCOPY])

(REMINDER.EXPIRATIONDATE
  [LAMBDA N                                                            (* lmm "19-Apr-85 17:26")

          (* * 1-arg case is only asking for information; multi-arg for update;
          3'rd arg, if non-null, says don't mark as changed.)

    (PROG (DEF SDATE NEWDATE)
          (if [OR (NOT (IGEQ N 1))
                  (NULL (SETQ DEF (if (type? PERIODIC.PROMPT.REMINDER (ARG N 1))
                                     then (ARG N 1)
                                   else (GETDEF (ARG N 1)
                                              'REMINDERS NIL '(NOERROR NOCOPY]
              then (RETURN NIL))
          (SETQ SDATE (fetch REMINDER.EXPIRATIONDATE of DEF))    (* Note that SDATE must be either a STRINGP or NIL)
          (if (IGREATERP N 1)
              then [SETQ NEWDATE (OR (STRINGP (ARG N 2))
                                     (GDATE (ARG N 2]
                   (if [AND (NOT (EQUAL (IDATE SDATE)
                                        (OR (IDATE NEWDATE)
                                            -1)))
                            (OR (ILEQ N 2)
                                (NULL (ARG N 3]
                       then (MARKASCHANGED (fetch REMINDER.NAME of DEF)
                                  'REMINDERS
                                  'CHANGED))
                   (replace REMINDER.EXPIRATIONDATE of DEF with (SETQ SDATE NEWDATE)))
          (RETURN SDATE])

(REMINDER.PERIOD
  [LAMBDA N                                                      (* JonL "11-Jun-84 13:49")
    (AND (IGEQ N 1)
         ([LAMBDA (DEF PERIOD)
            (AND (PROG1 DEF                                      (* Comment PPLossage))
                 ([LAMBDA (PERIOD NEWPERIOD)
                    (if (IEQP N 1)
                        then PERIOD
                      else (OR (IGEQ (SETQ NEWPERIOD (FIX (ARG N 2)))
                                     1)
                               (ERRORX (LIST 27 NEWPERIOD)))
                           (if (NOT (IEQP PERIOD NEWPERIOD))
                               then (MARKASCHANGED (fetch REMINDER.NAME of DEF)
                                           'REMINDERS
                                           'CHANGED))
                           (replace REMINDER.PERIOD of DEF with NEWPERIOD]
                  (fetch REMINDER.PERIOD of DEF]
          (if (type? PERIODIC.PROMPT.REMINDER (ARG N 1))
              then (ARG N 1)
            else (GETDEF (ARG N 1)
                       'REMINDERS NIL '(NOERROR NOCOPY])

(REMINDERS.RESTART
  [LAMBDA NIL                                                    (* lmm "20-Apr-85 12:32")
    (DEL.PROCESS 'REMINDERS.WATCHDOG)
    (OR (FIND.PROCESS 'REMINDERS.WATCHDOG)
        (ADD.PROCESS '(REMINDERS.WATCHDOG)
               'RESTARTABLE
               'HARDRESET
               'NAME
               'REMINDERS.WATCHDOG])

(REMINDERS.WATCHDOG
  [LAMBDA NIL                                                    (* lmm "20-Apr-85 12:15")
    (bind DELAY do (PERIODICALLYCHECKREMINDERS)
                   (AWAIT.EVENT \REMINDER.EVENT
                          (for PR in PERIODIC.PROMPT.REMINDERS when (\TIMER.TIMERP (fetch REMINDER.TIMEOUTBOX
                                                                                     of (CADR PR)))
                             smallest (ALTO.TO.LISP.DATE (fetch REMINDER.TIMEOUTBOX of (CADR PR)))
                             finally (RETURN (if $$EXTREME
                                                 then (IMIN (ITIMES 30 60 1000)
                                                            (IMAX 0 (ITIMES 1000 (IDIFFERENCE $$EXTREME (IDATE])
)

(PUTPROPS REMINDER.NEXTREMINDDATE ARGNAMES (REMINDER NEWVALUE))

(PUTPROPS REMINDER.EXPIRATIONDATE ARGNAMES (REMINDER NEWVALUE))

(PUTPROPS REMINDER.PERIOD ARGNAMES (REMINDER NEWVALUE))

(DEFINEQ

(\PUTREMINDER
  [LAMBDA (NAME FILEPKGTYPE DEF)                                 (* lmm "19-Apr-85 17:31")
    (DECLARE (GLOBALVARS PERIODIC.PROMPT.REMINDERS))
    (PROG ((OLDDEF (ASSOC NAME PERIODIC.PROMPT.REMINDERS))
           (PERIOD (fetch REMINDER.PERIOD of DEF)))
          (replace REMINDER.NAME of DEF)
          (MARKASCHANGED NAME FILEPKGTYPE (if (NULL OLDDEF)
                                              then (/SETTOPVAL 'PERIODIC.PROMPT.REMINDERS
                                                          (CONS (LIST NAME DEF)
                                                                PERIODIC.PROMPT.REMINDERS))
                                                   'DEFINED
                                            elseif (EQUAL (CDDR DEF)
                                                        (CDDR (CADR OLDDEF)))
                                             then

          (* Blaaag! Notice how the CDDR depends upon PERIODIC.PROMPT.REMINDER being a TYPERECORD so as to skip
          checking the REMINDER.TIMEOUTBOX)

                                                  (RETURN)
                                           else (/RPLACA (CDR OLDDEF)
                                                       DEF)
                                                'CHANGED))
          [AND (FIXP PERIOD)
               (replace REMINDER.TIMEOUTBOX of DEF with (SETUPTIMER PERIOD (fetch REMINDER.TIMEOUTBOX of DEF)
                                                            'SECONDS]
          (NOTIFY.EVENT \REMINDER.EVENT))
    NAME])
```

(\GETREMINDER

```
  [LAMBDA (NAME TYPE)                                             (* JonL "21-NOV-82 17:11")
    (CADR (ASSOC NAME PERIODIC.PROMPT.REMINDERS])
```

(\DELREMINDER

```
  [LAMBDA (NAME FILEPKGTYPE)                                      (* JonL "26-FEB-83 12:24")
    [if (OR (NULL NAME)
           (NOT (LITATOM NAME))
           (NEQ FILEPKGTYPE 'REMINDERS))
        then (ERRORX (LIST 27 (if (EQ FILEPKGTYPE 'REMINDERS)
                                 then NAME
                               else FILEPKGTYPE]
    (PROG ((OLDDEF (ASSOC NAME PERIODIC.PROMPT.REMINDERS)))
          (if OLDDEF
              then (MARKASCHANGED NAME FILEPKGTYPE 'DELETED)
                  (/SETTOPVAL 'PERIODIC.PROMPT.REMINDERS (REMOVE OLDDEF PERIODIC.PROMPT.REMINDERS))
                  (RETURN T])

)

(DEFINEQ
```

(PERIODICALLYCHECKREMINDERS

```
  [LAMBDA (RESETP)                                                (* lmm "19-Apr-85 18:28")
    (RESETLST
        (SETQ \PR.REMOVALS)
        [RESETSAVE NIL '(PROGN (MAPC \PR.REMOVALS (FUNCTION (LAMBDA (X)
                                                              (DELDEF (CAR X)
                                                                     'REMINDERS]
        (PROG (ACTED REMINDER Reminder'sTimer Reminmder'sPeriod Reminder'sExpiration MESSAGE FLASHFLG)
              [for X in PERIODIC.PROMPT.REMINDERS
                 do (SETQ REMINDER (CADR X))
                    (if [AND (SETQ Reminder'sExpiration (fetch REMINDER.FINALTIME of REMINDER))
                            (OR (NOT (\TIMER.TIMERP Reminder'sExpiration))
                                (TIMEREXPIRED? Reminder'sExpiration 'SECONDS]
                        then                                    (* An expiration date was set, and he has expired!)
                            (push \PR.REMOVALS X)
                      elseif [OR (NULL (SETQ Reminder'sTimer (fetch REMINDER.TIMEOUTBOX of REMINDER)))
                               (AND Reminder'sTimer (NOT (\TIMER.TIMERP Reminder'sTimer))
                                    (PROG1 T
                                        (replace REMINDER.TIMEOUTBOX of REMINDER with (SETQ Reminder'sTimer)))]
                        then

          (* This guy is just an old "one-shot" reminder which has already fired off, but is being kept around for the benefit of
          Denber.wbst)

                            (if (NULL Reminder'sExpiration)
                                then

          (* If he has a non-null expiration field, then the previous clause will eventuall delete him.)

                                     (push \PR.REMOVALS X))
                      else (SETQ Reminmder'sPeriod (FIXP (fetch REMINDER.PERIOD of REMINDER)))
                           (if RESETP
                               then

          (* Reset timers upon startup after SYSOUT or LOGOUT as if the call to SETUPTIMER were made upon startup)

                                   [if Reminmder'sPeriod
                                      then (replace REMINDER.TIMEOUTBOX of REMINDER with (SETUPTIMER
                                                                                           Reminmder'sPeriod
                                                                                           Reminder'sTimer
                                                                                           'SECONDS]
                             elseif (TIMEREXPIRED? Reminder'sTimer 'SECONDS)
                               then

          (* When a reminder's timer has expired, then flash it at the luser, or "run" it as a form to be eval'd.)

                                   (if Reminmder'sPeriod
                                          then

         (* For periodic reminders, set the next reminder time now, just in case he quits out of this with a ^D)

                                              (replace REMINDER.TIMEOUTBOX of REMINDER with (SETUPTIMER
                                                                                             Reminmder'sPeriod
                                                                                             Reminder'sTimer
                                                                                             'SECONDS))
                                        elseif (NULL Reminder'sExpiration)
                                          then

         (* After having "fired off" a one-shot reminder, then delete it, unless of course there is a future expiration date.)

                                              (push \PR.REMOVALS X)
                                        else                         (* This is to prevent a "kept, one-shot" reminder from firing off
                                                                     continuously.)
                                            (replace REMINDER.TIMEOUTBOX of REMINDER with NIL))
                                      (SETQ ACTED T)
                                      (SHOW.REMINDER REMINDER)
                                      (if Reminmder'sPeriod
                                          then                       (* Make the next reminder timeout more current.)
                                              (replace REMINDER.TIMEOUTBOX of REMINDER with (SETUPTIMER
                                                                                             Reminmder'sPeriod
                                                                                             Reminder'sTimer
                                                                                             'SECONDS]
             (AND ACTED CLOSEREMINDERSTREAMFLG (CLOSEW REMINDERSTREAM))))])

)

(RPAQ? \PR.REMOVALS NIL)

(RPAQ? DEFAULT.REMINDER.DURATION 60)

(RPAQ? DEFAULT.REMINDER.WINKINGDURATION 10)

(RPAQ? PERIODIC.PROMPT.REMINDERS NIL)

(RPAQ? REMINDERSTREAM PROMPTWINDOW)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \PR.REMOVALS REMINDERSTREAM PERIODIC.PROMPT.REMINDERS DEFAULT.REMINDER.DURATION
       DEFAULT.REMINDER.WINKINGDURATION DEFAULT.REMINDER.PERIOD)
)

[PUTDEF 'REMINDERS 'FILEPKGCOMS '((TYPE DESCRIPTION "Periodic PROMPT Reminders" GETDEF \GETREMINDER DELDEF
                                 \DELREMINDER PUTDEF \PUTREMINDER]

(RPAQ? CLOSEREMINDERSTREAMFLG )

(RPAQ \REMINDER.EVENT (CREATE.EVENT 'PERIODIC.PROMPT.REMINDERS))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS CLOSEREMINDERSTREAMFLG \REMINDER.EVENT)
)

(ADDTOVAR AFTERLOGOUTFORMS (REMINDERS.RESTART))

(REMINDERS.RESTART)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA REMINDER.PERIOD REMINDER.EXPIRATIONDATE REMINDER.NEXTREMINDDATE)
)

(PUTPROPS PROMPTREMINDERS COPYRIGHT ("Xerox Corporation" 1982 1983 1984 1985 1988))
```

{MEDLEY}<lispusers>PROMPTREMINDERS.;1 28-Jun-2024 18:34:03

-- Listed on 30-Jun-2024 13:14:42 --

FUNCTION INDEX

VARIABLE INDEX

PROPERTY INDEX

RECORD INDEX