

File created: 20-Jan-93 13:49:09 {DSK}<python>lde>lispcore>library>DMCHAT.;2

changes to: (RECORDS DM2500.STATE)

previous date: 11-Jun-90 15:39:12 {DSK}<python>lde>lispcore>library>DMCHAT.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1984, 1985, 1988, 1990, 1993 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ DMCHATCOMS

```
(  
  ;; DM2500 emulator, with some peculiar functions to handle its silly autocrlf properties  
  (FILES CHATTERMINAL)  
  (FNS DMCHAT.STATE DMCHAT.HANDLECHARACTER DMCHAT.HANDLE.WRAP DMCHAT.ADDRESS DMCHAT.CLEAR  
    DMCHAT.CLEARMODES DMCHAT.NEWLINE DMCHAT.RIGHT)  
  (ADDVARS (CHAT.DRIVERTYPES (DM2500 DMCHAT.HANDLECHARACTER DMCHAT.STATE)))  
  (VARIABLES CHAT.AUTOCRLF)  
  (DECLARE%: EVAL@COMPILE DONTCOPY (LOCALVARS . T)  
    (FILES (SOURCE)  
      CHATDECLS)  
    (RECORDS DM2500.STATE))  
  (INITRECORDS DM2500.STATE)  
  (SYSRECORDS DM2500.STATE)))
```

:: DM2500 emulator, with some peculiar functions to handle its silly autocrlf properties

(FILESLOAD CHATTERMINAL)

(DEFINEQ

(DMCHAT.STATE

```
[LAMBDA (CHAT.STATE) ; Edited 15-Feb-90 18:44 by bvm  
  (TERM.RESET.DISPLAY.PARMS CHAT.STATE)  
  (replace (CHAT.STATE CLEARMODEFN) of CHAT.STATE with (FUNCTION DMCHAT.CLEARMODES))  
  (TERM.HOME CHAT.STATE)  
  (create DM2500.STATE]))
```

(DMCHAT.HANDLECHARACTER

```
[LAMBDA (CHAR CHAT.STATE DM2500.STATE) ; Edited 11-Aug-88 16:35 by drc:  
  (\DTEST CHAT.STATE 'CHAT.STATE)  
  (\DTEST DM2500.STATE 'DM2500.STATE)  
  (PROG NIL  
    [COND  
      ((EQ CHAR (CHARCODE BELL))  
        (RETURN (COND  
          ((NEQ \MACHINETYPE \DORADO) ; Modern machines have audible bells  
            (BOUT (ffetch (CHAT.STATE DSP) of CHAT.STATE)  
              7))  
          ((NOT (ffetch (DM2500.STATE DINGED) of DM2500.STATE))  
            (CL:FUNCALL INVERTWINDOWFN (ffetch (CHAT.STATE WINDOW) of CHAT.STATE))  
              ; Complement window  
            (replace (DM2500.STATE DINGED) of DM2500.STATE with T])  
          (COND  
            ((ffetch (DM2500.STATE DINGED) of DM2500.STATE) ; Last character was a bell, with which we complemented screen.  
              ; Now back to normal  
            (CL:FUNCALL INVERTWINDOWFN (ffetch (CHAT.STATE WINDOW) of CHAT.STATE))  
              (replace (DM2500.STATE DINGED) of DM2500.STATE with NIL)))  
          (COND  
            ([AND (ffetch (DM2500.STATE AUTOLF) of DM2500.STATE)  
              (OR (NEQ CHAR (CHARCODE CR))  
                (NOT (ffetch (DM2500.STATE EATTOCRLF) of DM2500.STATE])  
                ;; We last received a CR, so DM wants auto LF after it. However, we postpone doing so until the next char is received, so that we get  
                ;; scroll holding right  
              (TERM.DOWN CHAT.STATE)  
              (replace (DM2500.STATE AUTOLF) of DM2500.STATE with NIL)))  
            (COND  
              ((ffetch (DM2500.STATE ADDRESSING) of DM2500.STATE) ; In the middle of receiving an address command  
                (COND  
                  ((DMCHAT.ADDRESS CHAT.STATE DM2500.STATE CHAR)  
                    (RETURN)  
                  (RETURN  
                    [COND  
                      ((AND (>= CHAR (CHARCODE SPACE))  
                        (< CHAR (CHARCODE DEL))) ; Normal char  
                      (replace (DM2500.STATE EATLF) of DM2500.STATE with (replace (DM2500.STATE EATCRLF) of DM2500.STATE  
                        with NIL))  
                    (RETURN (COND  
                      ((NOT (ffetch (DM2500.STATE EATTOCRLF) of DM2500.STATE))  
                        ; Print the char
```

```

(COND
  ((ffetch (DM2500.STATE IDMODE) of DM2500.STATE)
   ; this is discouraged by the DM manual, but apparently EMACS
   ; does it, so might as well support it
   (TERM.ADDCHAR CHAT.STATE)))
  (TERM.PRINTCHAR CHAT.STATE CHAR (FUNCTION DMCHAT.HANDLE.WRAP]
;; At this point, we have a non-printing char, presumably some command (or cr, lf).
[COND
  ((ffetch (DM2500.STATE EATLF) of DM2500.STATE)
   (freplace (DM2500.STATE EATLF) of DM2500.STATE with NIL)
   (COND
    ((EQ CHAR (CHARCODE LF))
     (RETURN)
     ; Yes, it was a LF, so we're done.
    )
   )
  [COND
  ((ffetch (DM2500.STATE EATCRLF) of DM2500.STATE)
   (COND
    ((EQ CHAR (CHARCODE CR))
     (freplace (DM2500.STATE EATLF) of DM2500.STATE with T)
     (freplace (DM2500.STATE EATCRLF) of DM2500.STATE with NIL)
     (RETURN)
     ; There's the CR, next eat the lf
    )
    (T
     ; Intervening control characters do not stop the eating, except for
     ; a few inconsistent exceptions...
     (SELCHARQ CHAR
      ((^B ^\ ^^ ^_)
       (freplace (DM2500.STATE EATCRLF) of DM2500.STATE with NIL))
      NIL]
  (SELCHARQ CHAR
   (LF (COND
    ((ffetch (DM2500.STATE IDMODE) of DM2500.STATE)
     (TERM.ADDLINE CHAT.STATE))
    (T (TERM.DOWN CHAT.STATE))))
   (CR (freplace (DM2500.STATE EATOCRLF) of DM2500.STATE with NIL)
    (DMCHAT.NEWLINE CHAT.STATE DM2500.STATE T))
   (BS (COND
    ((ffetch (DM2500.STATE IDMODE) of DM2500.STATE)
     (TERM.DELCHAR CHAT.STATE))
    (T (TERM.LEFT CHAT.STATE))))
   (^W
    (TERM.ERASE.TO.EOL CHAT.STATE))
    ; Erase to end of line
   (^L
    (freplace (DM2500.STATE ADDRESSING) of DM2500.STATE with -1))
    ; Start of cursor address
   (^B
    (TERM.HOME CHAT.STATE)
    (DMCHAT.CLEARMODES CHAT.STATE DM2500.STATE))
    ; Homes cursor, cancels some modes
   (^X
    (DMCHAT.CLEARMODES CHAT.STATE DM2500.STATE)
    (freplace (CHAT.STATE ROLLMODE) of CHAT.STATE with NIL))
    ; Cancel --resets modes
   ((^^ ^_)
    (DMCHAT.CLEAR CHAT.STATE DM2500.STATE))
    ; Master Reset -- Clears screen, modes
   (^\
    (COND
    ((ffetch (DM2500.STATE IDMODE) of DM2500.STATE)
     (TERM.ADDCHAR CHAT.STATE))
    (T (DMCHAT.RIGHT CHAT.STATE DM2500.STATE))))
    ; Forward space
   (^Z
    (COND
    ((ffetch (DM2500.STATE IDMODE) of DM2500.STATE)
     (TERM.DELETELINE CHAT.STATE))
    (T (TERM.UP CHAT.STATE))))
    ; Up
   ((^N ^O)
    (TERM.MODIFY.ATTRIBUTES CHAT.STATE 'BRIGHT)
    (freplace (DM2500.STATE BRIGHTMODE) of DM2500.STATE with T))
    ; Enter blink mode, enter protected mode. Do both as embolden
   (^P
    (freplace (DM2500.STATE IDMODE) of DM2500.STATE with T))
    ; Enter insert/delete mode
   (^%)
    (freplace (CHAT.STATE ROLLMODE) of CHAT.STATE with T))
    ; Set roll mode
   NIL])

```

DMCHAT.HANDLE.WRAP

```

[LAMBDA (CHAT.STATE)
;; Called when a character is printed in the last column of the screen
(LET ((DM2500.STATE (ffetch (CHAT.STATE TERM.STATE) of CHAT.STATE)))
 (COND
  (CHAT.AUTOCRLF
   (DMCHAT.NEWLINE CHAT.STATE DM2500.STATE))
   ; This is standard behavior--do auto crlf
  (T
   (freplace (DM2500.STATE EATOCRLF) of DM2500.STATE with T))
   ; An alternate mode some might like--flush everything til crlf.

```

DMCHAT.ADDRESS

```

[LAMBDA (CHAT.STATE DM2500.STATE CHAR)
(* ejs%: "12-May-85 15:26")
;; In the middle of doing absolute address, which is {^L, xpos, ypos}. Return T (meaning we handled the character) unless CHAR implies a
;; cancellation of the address, in which case caller must handle CHAR

```

```
(\DTEST DM2500.STATE 'DM2500.STATE)
(SELCHARQ CHAR
  (^X ^^ ^_) ; Cancel it, return NIL
  (freplace (DM2500.STATE ADDRESSING) of DM2500.STATE with NIL)
  NIL)
(^L ; Restarting the address in the middle of the address is legal
  (freplace (DM2500.STATE ADDRESSING) of DM2500.STATE with -1)
  T)
(LET ((ADDRESSING (ffetch (DM2500.STATE ADDRESSING) of DM2500.STATE))
      (NEXTPOS (LOGXOR CHAR 96)))
  (COND
    ((< ADDRESSING 0) ; Accept first (x) position
     (freplace (DM2500.STATE ADDRESSING) of DM2500.STATE with NEXTPOS))
    (T ; Accept second position and go there
     (TERM.MOVETO CHAT.STATE ADDRESSING NEXTPOS)
     (freplace (DM2500.STATE ADDRESSING) of DM2500.STATE with NIL)))
  T])
```

DMCHAT.CLEAR

```
[LAMBDA (CHAT.STATE DM2500.STATE SETROLL) (* ejs%: "12-May-85 17:13")
  [CLEARW (ffetch (CHAT.STATE WINDOW) of (\DTEST CHAT.STATE 'CHAT.STATE)
  (DMCHAT.CLEARMODES CHAT.STATE DM2500.STATE)
  (AND SETROLL (freplace (CHAT.STATE ROLLMODE) of CHAT.STATE with T))
  (TERM.HOME CHAT.STATE)])
```

DMCHAT.CLEARMODES

```
[LAMBDA (CHAT.STATE DM2500.STATE) ; Edited 15-Feb-90 18:37 by bvm
  (\DTEST CHAT.STATE 'CHAT.STATE)
  (DSPFONT (ffetch (CHAT.STATE PLAINFONT) of CHAT.STATE)
  (ffetch (CHAT.STATE DSP) of CHAT.STATE)) ; Restore normal font
  [freplace (DM2500.STATE BRIGHTMODE) of DM2500.STATE with (freplace (DM2500.STATE BLINKMODE) of DM2500.STATE
  with (freplace (DM2500.STATE IDMODE) of DM2500.STATE
  with (freplace (DM2500.STATE ADDRESSING)
  of DM2500.STATE with NIL))
  (freplace (CHAT.STATE ROLLMODE) of CHAT.STATE with T)])
```

DMCHAT.NEWLINE

```
[LAMBDA (CHAT.STATE DM2500.STATE EXPLICIT) (* ejs%: "12-May-85 15:12")
  ;; Do a CRLF. EXPLICIT = T means a CR was received, NIL means we did autowraparound
  (\DTEST CHAT.STATE 'CHAT.STATE)
  (MOVETO (freplace (CHAT.STATE XPOS) of CHAT.STATE with 0)
  (ffetch (CHAT.STATE YPOS) of CHAT.STATE)
  (ffetch (CHAT.STATE DSP) of CHAT.STATE)) ; Do only the CR part now, saving the LF for when next char
  (freplace (DM2500.STATE AUTOLF) of (\DTEST DM2500.STATE 'DM2500.STATE) with T) ; arrives
  (COND
    (EXPLICIT ; That WAS a cr, so eat the following lf
     (freplace (DM2500.STATE EATLF) of DM2500.STATE with T))
    (T ; Just wrapping, eat chars til crlf
     (freplace (DM2500.STATE EATCRLF) of DM2500.STATE with T)))
```

DMCHAT.RIGHT

```
[LAMBDA (CHAT.STATE DM2500.STATE) (* ejs%: "12-May-85 15:31")
  (LET [(XPOS (+ (ffetch (CHAT.STATE XPOS) of (\DTEST CHAT.STATE 'CHAT.STATE))
  (ffetch (CHAT.STATE FONTWIDTH) of CHAT.STATE))
      (COND
        ((< XPOS (ffetch (CHAT.STATE TTYWIDTH) of CHAT.STATE))
         (MOVETO (freplace (CHAT.STATE XPOS) of CHAT.STATE with XPOS)
         (ffetch (CHAT.STATE YPOS) of CHAT.STATE)
         (ffetch (CHAT.STATE DSP) of CHAT.STATE)))
        (T ; Auto crlf
         (DMCHAT.NEWLINE CHAT.STATE DM2500.STATE))
      )
```

```
(ADDTOVAR CHAT.DRIVERTYPES (DM2500 DMCHAT.HANDLECHARACTER DMCHAT.STATE))
```

```
(CL:DEFVAR CHAT.AUTOCRLF T
  "If true, dm2500 emulator performs automatic CRLF when it reaches the right edge of the display.")
```

```
(DECLARE%: EVAL@COMPILE DONTCOPY
```

```
(DECLARE%: DOEVAL@COMPILE DONTCOPY
```

```
(LOCALVARS . T)
```

```
)
```

```
(FILESLOAD (SOURCE)
  CHATDECLS)
```

```
(DECLARE%: EVAL@COMPILE
```

```
(DATATYPE DM2500.STATE ((DINGED FLAG)
                        (EATLF FLAG)
                        (EATCRLF FLAG)
                        (EATTOCRLF FLAG)
                        (AUTOLF FLAG)
                        ADDRESSING
                        (IDMODE FLAG)
                        (BLINKMODE FLAG)
                        (BRIGHTMODE FLAG)))
)
(/DECLAREDATATYPE 'DM2500.STATE ' (FLAG FLAG FLAG FLAG FLAG POINTER FLAG FLAG FLAG)
  ;; ---field descriptor list elided by lister---
  ' 4)
)
(/DECLAREDATATYPE 'DM2500.STATE ' (FLAG FLAG FLAG FLAG FLAG POINTER FLAG FLAG FLAG)
  ;; ---field descriptor list elided by lister---
  ' 4)
)
(ADDTOVAR SYSTEMRECLST (DATATYPE DM2500.STATE ((DINGED FLAG)
                                                (EATLF FLAG)
                                                (EATCRLF FLAG)
                                                (EATTOCRLF FLAG)
                                                (AUTOLF FLAG)
                                                ADDRESSING
                                                (IDMODE FLAG)
                                                (BLINKMODE FLAG)
                                                (BRIGHTMODE FLAG)))
)
(PUTPROPS DMCHAT COPYRIGHT ("Venue & Xerox Corporation" 1984 1985 1988 1990 1993))
```

FUNCTION INDEX

DMCHAT.ADDRESS2 DMCHAT.CLEARMODES3 DMCHAT.HANDLECHARACTER ..1 DMCHAT.RIGHT3
DMCHAT.CLEAR3 DMCHAT.HANDLE.WRAP2 DMCHAT.NEWLINE3 DMCHAT.STATE1

VARIABLE INDEX

CHAT.AUTOCRLF3 CHAT.DRIVERTYPES3 SYSTEMRECLST4

RECORD INDEX

DM2500.STATE4
