```
;;
;; Copyright (c) 1983-1988, 1990, 1993, 2022 by Venue & Xerox Corporation.


(RPAQQ VTCHATCOMS
        [;; VT100 emulator

        (FNS VTCHAT.STATE VTCHAT.HANDLECHARACTER VTCHAT.SEQUENCE VTCHAT.DOCOMMAND)
        (FNS VTCHAT.ADDRESS VTCHAT.REVERSE.INDEX VTCHAT.ATTRIBUTES VTCHAT.DECLFONT VTCHAT.CLEARMODES VTCHAT.SAVE
            VTCHAT.RESTORE VTCHAT.SETMODE VTCHAT.SETMARGINS VTCHAT.REPORT VTCHAT.STATUS)
        (INITVARS (VTCHAT.DEBUGGING.FLG)
              (VTCHAT.TERM.IDENTITY.STRING "Ÿ[?1;0c"))
        (GLOBALVARS VTCHAT.DEBUGGING.FLG CHATFONT GRAPHICSFONT VTCHAT.TERM.IDENTITY.STRING)
        (ADDVARS (CHAT.DRIVERTYPES (VT100 VTCHAT.HANDLECHARACTER VTCHAT.STATE)))
        (DECLARE%: EVAL@COMPILE DONTCOPY (LOCALVARS . T)
              (FILES (LOADCOMP)
                    CHATDECLS)
              (RECORDS VT100SAVE VT100.STATE))
        (INITRECORDS VT100.STATE)
        (SYSRECORDS VT100.STATE)
        (DECLARE%: DONTEVAL@LOAD DOCOPY (FILES (SYSLOAD)
                                              VT100KP)

              ;; lmm 2/20/22:

              ;; (CAAR CHAT.DISPLAYTYPES) had "replace this string with NIL to prefer VT100 for chat", now replaced

              (ADDVARS (CHAT.DISPLAYTYPES (NIL NIL VT100]


;; VT100 emulator

(DEFINEQ

(VTCHAT.STATE
  [LAMBDA (CHAT.STATE)                                       ; Edited 15-Feb-90 18:44 by bvm
    (replace (CHAT.STATE TERM.IDENTITY.STRING) of CHAT.STATE with VTCHAT.TERM.IDENTITY.STRING)
    (replace (CHAT.STATE TERM.TAB.STOPS) of CHAT.STATE
        with '(8 16 24 32 40 48 56 64 72 80 88 96 104 112 120 128 136))
    (replace (CHAT.STATE WRAPMODE) of CHAT.STATE with T)
    (TERM.RESET.DISPLAY.PARMS CHAT.STATE)
    (replace (CHAT.STATE CLEARMODEFN) of CHAT.STATE with (FUNCTION VTCHAT.CLEARMODES))
    (TERM.HOME CHAT.STATE)
    (create VT100.STATE])


(VTCHAT.HANDLECHARACTER
  [LAMBDA (CHAR CHAT.STATE VT100.STATE)                      ; Edited 11-Aug-88 17:19 by drc:
    (DECLARE (GLOBALVARS VTCHAT.DEBUGGING.FLG \MACHINETYPE))
    (\DTEST CHAT.STATE 'CHAT.STATE)
    (\DTEST VT100.STATE 'VT100.STATE)

    ;; Here and/or below

    [COND
        (VTCHAT.DEBUGGING.FLG (COND
                                  ((ILESSP CHAR (CHARCODE SPACE))
                                   (PRINTOUT PROMPTWINDOW "[" CHAR "]"))
                                  (T (\OUTCHAR (GETSTREAM PROMPTWINDOW)
                                           CHAR]
    (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 (VT100.STATE DINGED) OF VT100.STATE))
                          (CL:FUNCALL INVERTWINDOWFN (FFETCH (CHAT.STATE WINDOW) OF CHAT.STATE))
                                                            ; Complement window
                          (FREPLACE (VT100.STATE DINGED) OF VT100.STATE WITH T]
        (COND
            ((FFETCH (VT100.STATE DINGED) OF VT100.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))
             (FREPLACE (VT100.STATE DINGED) OF VT100.STATE WITH NIL)))
        (COND
            ((EQ CHAR (CHARCODE ESC))
```

```
                    (FREPLACE (VT100.STATE AUTOLF) OF VT100.STATE WITH NIL)))
            (COND
               [(>= CHAR (CHARCODE SPACE))                                    ; Normal char
                 (COND
                    ((FFETCH (VT100.STATE ESCAPESEQUENCE) OF VT100.STATE)
                     (VTCHAT.SEQUENCE CHAT.STATE VT100.STATE CHAR))
                    (T (FREPLACE (VT100.STATE EATLF) OF VT100.STATE WITH (FREPLACE (VT100.STATE EATCRLF)
                                                                                  OF VT100.STATE WITH NIL))
                       (RETURN (COND
                                  ((AND (NEQ CHAR (CHARCODE DEL))
                                        (NOT (FFETCH (VT100.STATE EATTOCRLF) OF VT100.STATE)))
                                                                     ; Print the char
                                   (TERM.PRINTCHAR CHAT.STATE CHAR]
               (T (SELCHARQ CHAR
                        ((LF FF ^K)                                  ; Line Feed
                          (COND
                             ((NOT (FFETCH (VT100.STATE EATLF) OF VT100.STATE))
                              (TERM.DOWN CHAT.STATE))
                             (T (FREPLACE (VT100.STATE EATLF) OF VT100.STATE WITH NIL))))
                        (^I                                          ; Tab
                          (TERM.TAB CHAT.STATE))
                        (CR                                          ; Carriage return
                          (MOVETO (freplace (CHAT.STATE XPOS) of CHAT.STATE with 0)
                                  (FFETCH (CHAT.STATE YPOS) OF CHAT.STATE)
                                  (FFETCH (CHAT.STATE DSP) OF CHAT.STATE)))
                        (BS                                          ; Back space
                          (TERM.LEFT CHAT.STATE 1))
                        ((^X ^Z)                                     ; Cancel --resets modes
                          (VTCHAT.CLEARMODES CHAT.STATE VT100.STATE))
                        (^N                                          ; SO --- character set switch
                          (DSPFONT (FFETCH (VT100.STATE SOFONT) OF VT100.STATE)
                                   (FFETCH (CHAT.STATE WINDOW) OF CHAT.STATE)))
                        (^O                                          ; SI --- character set switch
                          (DSPFONT (FFETCH (VT100.STATE SIFONT) OF VT100.STATE)
                                   (FFETCH (CHAT.STATE WINDOW) OF CHAT.STATE)))
                        (ESC                                         ; Start of ESC sequence
                           (FREPLACE (VT100.STATE ESCAPESEQUENCE) OF VT100.STATE WITH 0)
                           (FREPLACE (VT100.STATE CSTERM) OF VT100.STATE WITH 48))
                        NIL])
```

(**VTCHAT.SEQUENCE**

```
  [LAMBDA (CHAT.STATE VT100.STATE CHAR)                              ; Edited 18-Dec-86 15:14 by amd

     ;; Here when an ESC has been seen

     (COND
        ([> CHAR (ffetch (VT100.STATE CSTERM) of (\DTEST VT100.STATE 'VT100.STATE]
         (VTCHAT.DOCOMMAND CHAT.STATE VT100.STATE CHAR))
        (T (LET ((ESCAPESEQUENCE (ffetch (VT100.STATE ESCAPESEQUENCE) of VT100.STATE)))
               (COND
                  ((EQ ESCAPESEQUENCE 0)
                   (SELCHARQ CHAR
                       ("(" (freplace (VT100.STATE CHARSET0) of VT100.STATE with T)
                            (freplace (VT100.STATE ESCAPESEQUENCE) of VT100.STATE with 1)
                            (freplace (VT100.STATE CSTERM) of VT100.STATE with (CHARCODE B)))
                       (")" (freplace (VT100.STATE CHARSET1) of VT100.STATE with T)
                            (freplace (VT100.STATE ESCAPESEQUENCE) of VT100.STATE with 1)
                            (freplace (VT100.STATE CSTERM) of VT100.STATE with (CHARCODE B)))
                       (printout PROMPTWINDOW "Bad ESCAPESEQUENCE--CHAR is " (CHARACTER CHAR)
                              T)))
                  ((OR (ffetch (VT100.STATE CHARSET0) of VT100.STATE)
                       (ffetch (VT100.STATE CHARSET1) of VT100.STATE))
                   (VTCHAT.DECLFONT CHAT.STATE VT100.STATE CHAR))
                  (T (LET ((PARAMARRAY (ffetch (VT100.STATE PARAMARRAY) of VT100.STATE))
                           TEMP)
                          (IF (AND (>= CHAR (CHARCODE 0))
                                   (<= CHAR (CHARCODE 9)))
                              THEN                                   ; Continue building current numeric argument
                                  [SETQ TEMP (+ (ITIMES 10 (ELT PARAMARRAY ESCAPESEQUENCE))
                                                (- CHAR (CHARCODE 0]
                                  (COND
                                     ((< TEMP MAX.SMALLP)
                                      (SETA PARAMARRAY ESCAPESEQUENCE TEMP)))
                              ELSEIF (EQ CHAR (CHARCODE ";"))
                              THEN                                   ; Param separator--start a new one
                                   (freplace (VT100.STATE ESCAPESEQUENCE) of VT100.STATE with (add ESCAPESEQUENCE
                                                                                                    1))
                                   (SETA PARAMARRAY ESCAPESEQUENCE 0])
```

(**VTCHAT.DOCOMMAND**

```
  [LAMBDA (CHAT.STATE VT100.STATE CHAR)                              ; Edited 18-Dec-86 15:08 by amd

     ;; Function called when an escape or control sequence has been terminated by CHAR

     (\DTEST CHAT.STATE 'CHAT.STATE)
     (PROG ([PARAMARRAY (ffetch (VT100.STATE PARAMARRAY) of (\DTEST VT100.STATE 'VT100.STATE]
            (ESCAPESEQUENCE (ffetch (VT100.STATE ESCAPESEQUENCE) of VT100.STATE))
```

```
            PARAM1 PARAM2 STAYESCAPE)
        [if (> ESCAPESEQUENCE 0)
          then (if (EQ (SETQ PARAM1 (ELT PARAMARRAY 1))
                       0)
                  then                                              ; Param = 0 is same as omitted
                      (SETQ PARAM1 NIL))
             (if (> ESCAPESEQUENCE 1)
               then (if (EQ (SETQ PARAM2 (ELT PARAMARRAY 2))
                            0)
                      then (SETQ PARAM2 NIL]
        (SELCHARQ CHAR
             (%[                                                    ; ESC-LeftBracket is the control sequence introducer
              (freplace (VT100.STATE CSTERM) of VT100.STATE with 64)
              (freplace (VT100.STATE ESCAPESEQUENCE) of VT100.STATE with 1)
              (SETQ STAYESCAPE T)
              (SETA PARAMARRAY 1 0))
             (7                                                     ; ESC 7 -> Save parameters
              (VTCHAT.SAVE CHAT.STATE VT100.STATE))
             (8                                                     ; ESC 8 -> Restore parameters
              (VTCHAT.RESTORE CHAT.STATE VT100.STATE))
             (A                                                     ; ESC [ Pn A -> Move up; param1 indicates how far
              (TERM.UP CHAT.STATE (OR PARAM1 1)))
             (B                                                     ; ESC [ Pn B -> Move down; param1 indicates how far
              (TERM.GODOWN CHAT.STATE (OR PARAM1 1)))
             (C                                                     ; ESC [ Pn C -> Move right; param1 indicates how far
              (TERM.RIGHT CHAT.STATE (OR PARAM1 1)))
             (D (if (EQ 0 ESCAPESEQUENCE)
                  then                                              ; ESC D -> index
                      (TERM.DOWN CHAT.STATE)
                  else                                              ; ESC [ Pn D -> cursor backwards
                      (TERM.LEFT CHAT.STATE (OR PARAM1 1))))
             (E                                                     ; ESC E -> Do CRLF
              (TERM.NEWLINE CHAT.STATE))
             ((H f)
               (if (AND (EQ CHAR (CHARCODE H))
                        (EQ ESCAPESEQUENCE 0))
                  then                                              ; ESC H -> Set tab at position
                      (TERM.SET.TAB CHAT.STATE (IQUOTIENT (ffetch (CHAT.STATE XPOS) of CHAT.STATE)
                                                         (ffetch (CHAT.STATE FONTWIDTH) of CHAT.STATE)))
                elseif (NEQ ESCAPESEQUENCE 0)
                  then                                              ; ESC [ Pn H -> Cursor addressing.  Default coord is 1
                      (VTCHAT.ADDRESS CHAT.STATE VT100.STATE (OR PARAM1 1)
                         (OR PARAM2 1))))
             (J                                                     ; Erase in display; param1 indicates mode
              (TERM.ERASE.IN.DISPLAY CHAT.STATE (OR PARAM1 0)))
             (K                                                     ; Erase in line; param1 indicates mode
              (TERM.ERASE.IN.LINE CHAT.STATE (OR PARAM1 0)))
             (M                                                     ; Reverse Index
              (VTCHAT.REVERSE.INDEX CHAT.STATE VT100.STATE))
             (Z                                                     ; What are you?
              (TERM.IDENTIFY.SELF CHAT.STATE))
             (=                                                     ; Enter keypad application mode
              (VTCHAT.KPAPPLMODE CHAT.STATE VT100.STATE T))
             (>                                                     ; Leave keypad application mode
              (VTCHAT.KPAPPLMODE CHAT.STATE VT100.STATE NIL))
             (c                                                     ; What are you?
              (TERM.IDENTIFY.SELF CHAT.STATE))
             ((h l)                                                 ; Set or clear modes
               (VTCHAT.SETMODE CHAT.STATE VT100.STATE PARAMARRAY ESCAPESEQUENCE (EQ CHAR (CHARCODE h))))
             (m                                                     ; Set char attributes
               (VTCHAT.ATTRIBUTES CHAT.STATE VT100.STATE PARAMARRAY ESCAPESEQUENCE))
             (n                                                     ; Status report
               (VTCHAT.STATUS CHAT.STATE VT100.STATE (OR PARAM1 0)))
             (r                                                     ; Set scrolling margins.  Default is whole screen
              [VTCHAT.SETMARGINS CHAT.STATE VT100.STATE (OR PARAM1 1)
                 (OR PARAM2 (LET ((FONTHEIGHT (ffetch (CHAT.STATE FONTHEIGHT) of CHAT.STATE)))
                             (ITIMES (IQUOTIENT (ffetch (CHAT.STATE TTYHEIGHT) of CHAT.STATE)
                                               FONTHEIGHT)
                                    FONTHEIGHT])
             (x (VTCHAT.REPORT CHAT.STATE VT100.STATE))
             (\)
             (PROGN
                     ;; (COND ((ZEROP ESCAPESEQUENCE) (printout PROMPTWINDOW 'ESC ' (CHARACTER CHAR) T)) (T (printout
                     ;; PROMPTWINDOW 'ESC[' PARAM1 ';' PARAM2 ' ' (CHARACTER CHAR) T)))

                 NIL))
        (OR STAYESCAPE (freplace (VT100.STATE ESCAPESEQUENCE) of VT100.STATE with NIL])
)


(DEFINEQ
```

(**VTCHAT.ADDRESS**
```
  [LAMBDA (CHAT.STATE VT100.STATE ROW COLUMN)                       ; Edited 18-Dec-86 15:06 by amd

    ;; Do absolute positioning

    [COND
        ((fetch (VT100.STATE RELORIGIN) of VT100.STATE)
```

```
            (LET ([TOPMARGIN (ffetch (CHAT.STATE TOPMARGIN) of (\DTEST CHAT.STATE 'CHAT.STATE]
                  (BOTTOMMARGIN (ffetch (CHAT.STATE BOTTOMMARGIN) of CHAT.STATE)))
                 (add ROW TOPMARGIN)
                 (COND
                    ((< ROW TOPMARGIN)
                     (SETQ ROW TOPMARGIN))
                    ((> ROW BOTTOMMARGIN)
                     (SETQ ROW BOTTOMMARGIN]
             (TERM.MOVETO CHAT.STATE (SUB1 COLUMN)
                 (SUB1 ROW))
         T])
```

(**VTCHAT.REVERSE.INDEX**
```
   [LAMBDA (CHAT.STATE VT100.STATE)                                  (* ejs%: "18-Nov-85 12:58")
      (LET* ([YPOS (ffetch (CHAT.STATE YPOS) of (\DTEST CHAT.STATE 'CHAT.STATE]
             (FONTHEIGHT (ffetch (CHAT.STATE FONTHEIGHT) of CHAT.STATE))
             (TOPMARGIN (ffetch (CHAT.STATE TOPMARGIN) of CHAT.STATE))
             (TOPLINE (- TOPMARGIN FONTHEIGHT)))
            (COND
               ((< YPOS TOPLINE)
                (MOVETO (ffetch (CHAT.STATE XPOS) of CHAT.STATE)
                        (freplace (CHAT.STATE YPOS) of CHAT.STATE with (IMIN TOPLINE (+ YPOS FONTHEIGHT)))
                        (ffetch (CHAT.STATE DSP) of CHAT.STATE)))
               (T (TERM.SCROLLDOWN CHAT.STATE TOPMARGIN])
```

(**VTCHAT.ATTRIBUTES**
```
   [LAMBDA (CHAT.STATE VT100.STATE ATTRARRAY ATTRCOUNT)             ; Edited 18-Dec-86 15:06 by amd

      ;; Function to do character attribute setting

      (TERM.MODIFY.ATTRIBUTES CHAT.STATE (for I from 1 to ATTRCOUNT bind A when [SETQ A (CASE (ELT ATTRARRAY I)
                                                                             (0 'NORMAL)
                                                                             (1 'BRIGHT)
                                                                             (4 'BLINK)
                                                                             (5 'UNDERLINE)
                                                                             (7 'INVERSE))]

                                         collect A])
```

(**VTCHAT.DECLFONT**
```
   [LAMBDA (CHAT.STATE VT100.STATE CHAR)                            (* ejs%: "20-Mar-86 14:41")
      (freplace (VT100.STATE CHARSET1) of VT100.STATE with NIL)
      (freplace (VT100.STATE CHARSET0) of VT100.STATE with NIL)
      (freplace (VT100.STATE ESCAPESEQUENCE) of VT100.STATE with NIL])
```

(**VTCHAT.CLEARMODES**
```
   [LAMBDA (CHAT.STATE VT100.STATE)                                ; Edited 18-Dec-86 15:09 by amd
      (COND
         ((OR (ffetch (VT100.STATE BLINKMODE) of (\DTEST VT100.STATE 'VT100.STATE))
              (ffetch (VT100.STATE BRIGHTMODE) of VT100.STATE))
          (DSPFONT (ffetch (CHAT.STATE PLAINFONT) of (\DTEST CHAT.STATE 'CHAT.STATE))
                   (ffetch (CHAT.STATE DSP) of CHAT.STATE))            ; Restore normal font
            (freplace (VT100.STATE BRIGHTMODE) of VT100.STATE with (freplace (VT100.STATE BLINKMODE) of VT100.STATE
                                                                     with NIL])
```

(**VTCHAT.SAVE**
```
   [LAMBDA (CHAT.STATE VT100.STATE)                                ; Edited 18-Dec-86 15:14 by amd

      ;; Function to save current curpos position, graphic rendition, and character set

      (LET [[VT100MEM (OR (ffetch (VT100.STATE VT100MEM) of (\DTEST VT100.STATE 'VT100.STATE))
                          (freplace (VT100.STATE VT100MEM) of VT100.STATE with (create VT100SAVE]
            (DSP (ffetch (CHAT.STATE DSP) of (\DTEST CHAT.STATE 'CHAT.STATE))
           (replace (VT100SAVE CURSORPOS) of VT100MEM with (create POSITION
                                                             XCOORD _ (ffetch (CHAT.STATE XPOS) of CHAT.STATE)
                                                             YCOORD _ (ffetch (CHAT.STATE YPOS) of CHAT.STATE)))
           (replace (VT100SAVE CHARATTR) of VT100MEM with (LIST (DSPFONT NIL DSP)
                                                                (ffetch (CHAT.STATE UNDERLINEMODE) of CHAT.STATE)
                                                                (DSPSOURCETYPE NIL DSP])
```

(**VTCHAT.RESTORE**
```
   [LAMBDA (CHAT.STATE VT100.STATE)                                ; Edited 18-Dec-86 15:13 by amd

      ;; Function to restor cursor, etc from storage

      (LET [[DSP (ffetch (CHAT.STATE DSP) of (\DTEST CHAT.STATE 'CHAT.STATE]
            (VT100MEM (ffetch (VT100.STATE VT100MEM) of (\DTEST VT100.STATE 'VT100.STATE]
           (if VT100MEM
               then (LET ((ATTRS (fetch (VT100SAVE CHARATTR) of VT100MEM))
                          (CURSOR (fetch (VT100SAVE CURSORPOS) of VT100MEM)))
                         (MOVETO (freplace (CHAT.STATE XPOS) of CHAT.STATE with (CAR CURSOR))
                                 (freplace (CHAT.STATE YPOS) of CHAT.STATE with (CDR CURSOR))
                                 DSP)
                         (DSPFONT (CAR ATTRS)
                                  DSP)
```

```
                              (freplace (CHAT.STATE UNDERLINEMODE) of CHAT.STATE with (CADR ATTRS))
                          (DSPSOURCETYPE (CADDR ATTRS)
                                    DSP])
```

(**VTCHAT.SETMODE**
```
  [LAMBDA (CHAT.STATE VT100.STATE MODEARRAY SETCOUNT ON?)          ; Edited  8-Dec-87 12:16 by jrb:
```

;; Does mode setting. Set indicated modes on (on? = T) or off (on? = NIL).

```
  (\DTEST VT100.STATE 'VT100.STATE)
  (\DTEST CHAT.STATE 'CHAT.STATE)
  (for M from 1 to SETCOUNT do (CASE (ELT MODEARRAY M)
                          (1 (replace (VT100.STATE CURSORMODE) of VT100.STATE with ON?))
                          (4 (freplace (VT100.STATE SMOOTHSCROLL) of VT100.STATE with ON?))
                          (5 [COND
                                ((NEQ (ffetch (VT100.STATE INVERTFLG) of VT100.STATE)
                                      ON?)
                                 (freplace (VT100.STATE INVERTFLG) of VT100.STATE with ON?)
                                 (INVERTW (ffetch (CHAT.STATE WINDOW) of CHAT.STATE))
                                 (LET ((DSP (ffetch (CHAT.STATE DSP) of CHAT.STATE)))
                                      (IF ON?
                                          THEN                  ; White on black display
                                              (DSPSOURCETYPE 'INVERT DSP)
                                              (DSPTEXTURE BLACKSHADE DSP)
                                          ELSE                  ; Normal black on white
                                              (DSPTEXTURE WHITESHADE DSP)
                                              (DSPSOURCETYPE 'INPUT DSP])
                          (6 (freplace (VT100.STATE RELORIGIN) of VT100.STATE with ON?)
                             (VTCHAT.ADDRESS CHAT.STATE VT100.STATE 1 1))
                          (7 (freplace (CHAT.STATE WRAPMODE) of CHAT.STATE with ON?)))])
```

(**VTCHAT.SETMARGINS**
```
  [LAMBDA (CHAT.STATE VT100.STATE TOP BOTTOM)                      ; Edited 18-Dec-86 15:15 by amd
```

;; Function to set top and bottom margins

```
  (LET ([FONTHEIGHT (ffetch (CHAT.STATE FONTHEIGHT) of (\DTEST CHAT.STATE 'CHAT.STATE]
        (HOMEPOS (ffetch (CHAT.STATE HOMEPOS) of CHAT.STATE)))
       [freplace (CHAT.STATE TOPMARGIN) of CHAT.STATE with (IMAX 0 (- HOMEPOS (ITIMES (- TOP 2)
                                                                            FONTHEIGHT]
       [freplace (CHAT.STATE BOTTOMMARGIN) of CHAT.STATE with (IMAX 0 (- HOMEPOS (ITIMES (SUB1 BOTTOM)
                                                                            FONTHEIGHT)
                                                               (ffetch (CHAT.STATE FONTDESCENT)
                                                                   of CHAT.STATE]
       (TERM.HOME CHAT.STATE])
```

(**VTCHAT.REPORT**
```
  [LAMBDA (CHAT.STATE VTCHAT.STATE)                                ; Edited 18-Dec-86 15:12 by amd
```

;; Report terminal parameters -- DECREPTPARM

```
  (LET ((OUTSTREAM (fetch (CHAT.STATE OUTSTREAM) of CHAT.STATE)))
       (PRIN4 "Ÿ[2;1;1;" OUTSTREAM)
       (COND
          [(EQ (fetch (STREAM DEVICE) of OUTSTREAM)
               \RS232C.FDEV)
           (LET [(BAUD (CDR (FASSOC [CDR (FASSOC 'LINE.SPEED (RS232C.GET.PARAMETERS '(LINE.SPEED]
                           '((50 . 0)
                             (75 . 8)
                             (110 . 16)
                             (150 . 32)
                             (200 . 40)
                             (300 . 48)
                             (600 . 56)
                             (1200 . 64)
                             (1800 . 72)
                             (2000 . 80)
                             (2400 . 88)
                             (3600 . 96)
                             (4800 . 104)
                             (9600 . 112)
                             (19200 . 120]
                (COND
                    (BAUD (printout OUTSTREAM BAUD ";" BAUD ";"))
                    (T (printout OUTSTREAM "0;0;"]
          (T (printout OUTSTREAM "0;0;")))
       (PRIN1 "1;0x" OUTSTREAM)
       (FORCEOUTPUT OUTSTREAM])
```

(**VTCHAT.STATUS**
```
  [LAMBDA (CHAT.STATE VT100.STATE TYPE)                            ; Edited 30-Sep-2021 17:30 by briggs
                                                                  ; Edited 18-Dec-86 15:16 by amd
```

;; Returns VT100 status info

```
  (LET [(OUTSTREAM (ffetch (CHAT.STATE OUTSTREAM) of (\DTEST CHAT.STATE 'CHAT.STATE]
       (SELECTQ TYPE
```

```
                    (5                                              ; Host wants device status
                        (PRIN1 "Ÿ[0n" OUTSTREAM))
                    (6                                              ; Host wants cursor coords
                        (BOUT OUTSTREAM (CHARCODE ESC))
                        (BOUT OUTSTREAM (CHARCODE %[))
                        (PRIN1 (ADD1 (IQUOTIENT (ffetch (CHAT.STATE YPOS) of CHAT.STATE)
                                                (ffetch (CHAT.STATE FONTHEIGHT) of CHAT.STATE)))
                                OUTSTREAM)
                        (BOUT OUTSTREAM (CHARCODE ;))
                        (PRIN1 (ADD1 (IQUOTIENT (ffetch (CHAT.STATE XPOS) of CHAT.STATE)
                                                (ffetch (CHAT.STATE FONTWIDTH) of CHAT.STATE)))
                                OUTSTREAM)
                        (BOUT OUTSTREAM (CHARCODE R)))
                 NIL)
            (FORCEOUTPUT OUTSTREAM])
)

(RPAQ? VTCHAT.DEBUGGING.FLG )

(RPAQ? VTCHAT.TERM.IDENTITY.STRING "Ÿ[?1;0c")

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS VTCHAT.DEBUGGING.FLG CHATFONT GRAPHICSFONT VTCHAT.TERM.IDENTITY.STRING)
)

(ADDTOVAR CHAT.DRIVERTYPES (VT100 VTCHAT.HANDLECHARACTER VTCHAT.STATE))

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)

(FILESLOAD (LOADCOMP)
        CHATDECLS)

(DECLARE%: EVAL@COMPILE

(RECORD VT100SAVE (CURSORPOS CHARATTR VTSCHARSET))

(DATATYPE VT100.STATE ((DINGED FLAG)
                       (EATLF FLAG)
                       (EATCRLF FLAG)
                       (EATTOCRLF FLAG)
                       (AUTOLF FLAG)
                       (BLINKMODE FLAG)
                       (BRIGHTMODE FLAG)
                       (SMOOTHSCROLL FLAG)
                       (SIFONT POINTER)
                       (KEYPADMODE FLAG)
                       (CURSORMODE FLAG)
                       (CHARSET0 FLAG)
                       (CHARSET1 FLAG)
                       (SOFONT POINTER)
                       (PARAMCOUNT WORD)
                       (ADDRESSING WORD)
                       ESCAPESEQUENCE VT100MEM PARAMARRAY RELORIGIN INVERTFLG CSTERM)
        VT100MEM _ (create VT100SAVE
                          CURSORPOS _ (create POSITION
                                              XCOORD _ 1
                                              YCOORD _ 1))
        PARAMARRAY _ (ARRAY 12 'SMALLP 0 1))
)

(/DECLAREDATATYPE 'VT100.STATE
        '(FLAG FLAG FLAG FLAG FLAG FLAG FLAG FLAG POINTER FLAG FLAG FLAG FLAG POINTER WORD WORD POINTER POINTER
                POINTER POINTER POINTER POINTER)
        ;; ---field descriptor list elided by lister---
        '20)
)

(/DECLAREDATATYPE 'VT100.STATE
        '(FLAG FLAG FLAG FLAG FLAG FLAG FLAG FLAG POINTER FLAG FLAG FLAG FLAG POINTER WORD WORD POINTER POINTER
                POINTER POINTER POINTER POINTER)
        ;; ---field descriptor list elided by lister---
        '20)
)

(ADDTOVAR SYSTEMRECLST
        (DATATYPE VT100.STATE ((DINGED FLAG)
                               (EATLF FLAG)
                               (EATCRLF FLAG)
                               (EATTOCRLF FLAG)
```

```
                                     (AUTOLF FLAG)
                                     (BLINKMODE FLAG)
                                     (BRIGHTMODE FLAG)
                                     (SMOOTHSCROLL FLAG)
                                     (SIFONT POINTER)
                                     (KEYPADMODE FLAG)
                                     (CURSORMODE FLAG)
                                     (CHARSET0 FLAG)
                                     (CHARSET1 FLAG)
                                     (SOFONT POINTER)
                                     (PARAMCOUNT WORD)
                                     (ADDRESSING WORD)
                     ESCAPESEQUENCE VT100MEM PARAMARRAY RELORIGIN INVERTFLG CSTERM)))
```

(DECLARE%: DONTEVAL@LOAD DOCOPY

(FILESLOAD (SYSLOAD)
        VT100KP)

(ADDTOVAR **CHAT.DISPLAYTYPES** (NIL NIL VT100))
)

(PUTPROPS **VTCHAT COPYRIGHT** ("Venue & Xerox Corporation" 1983 1984 1985 1986 1987 1988 1990 1993 2022))

FUNCTION INDEX

VARIABLE INDEX

RECORD INDEX