

File created: 26-Feb-86 10:59:11 {ERIS}<LISPCORE>LIBRARY>TEDITCOLOR.;3

changes to: (VARS TEDITCOLORCOMS)

previous date: 26-Feb-86 10:44:36 {ERIS}<LISPCORE>LIBRARY>TEDITCOLOR.;1

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS

(* * Copyright (c) 1986 by Xerox Corporation. All rights reserved.)

(RPAQQ TEDITCOLORCOMS

```
(( * * These function definitions should be added to IMAGEOBJ, TEDITSCREEN, and TEDITWINDOW. They have
been tested to the point of creating, putting, and getting color TEDIT files. \TEDIT.SLOWBLTCHAR is a
new function. *)
(* * NOTE: The MACRO property for SCREENBITMAP was eliminated 25-FEB. Either LOAD the new EXPORTS.ALL or
just (REMPROP (QUOTE SCREENBITMAP)
(QUOTE MACRO))
so that the IMAGEOBJ functions will compile correctly. *)
(COMS (* * Modifications to IMAGEOBJ. *)
(FNS BITMAPOBJ.SNAPW COERCETOBITMAP))
(COMS (* * Modifications to TEDITSCREEN. (\TEDIT.SLOWBLTCHAR is a new function.)
*)
(FNS \TEDIT.BLTCHAR \TEDIT.SLOWBLTCHAR \TEDIT.CREATE.LINECACHE \TEDIT.LINECACHE))
(COMS (* * Modifications to TEDITWINDOW. *)
(FNS TEDIT.MINIMAL.WINDOW.SETUP))))
```

(* * These function definitions should be added to IMAGEOBJ, TEDITSCREEN, and TEDITWINDOW.
They have been tested to the point of creating, putting, and getting color TEDIT files.
\TEDIT.SLOWBLTCHAR is a new function. *)

(* * NOTE: The MACRO property for SCREENBITMAP was eliminated 25-FEB.
Either LOAD the new EXPORTS.ALL or just (REMPROP (QUOTE SCREENBITMAP)
(QUOTE MACRO)) so that the IMAGEOBJ functions will compile correctly.
*)

(* * Modifications to IMAGEOBJ. *)

(DEFINEQ

(BITMAPOBJ.SNAPW

[LAMBDA NIL

(* kbr: "25-Feb-86 17:06")

(* * makes an image object of a prompted for region of the screen.)

```
(PROG (SCREENREGION SCREEN REGION BM)
(SETQ SCREENREGION (GETSCREENREGION))
(SETQ SCREEN (fetch (SCREENREGION SCREEN) of SCREENREGION))
(SETQ REGION (fetch (SCREENREGION REGION) of SCREENREGION))
[SETQ BM (BITMAPCREATE (fetch (REGION WIDTH) of REGION)
(fetch (REGION HEIGHT) of REGION)
(BITSPERPIXEL (SCREENBITMAP SCREEN)
(BITBLT (SCREENBITMAP SCREEN)
(fetch (REGION LEFT) of REGION)
(fetch (REGION BOTTOM) of REGION)
BM 0 0 NIL NIL (QUOTE INPUT)
(QUOTE REPLACE))
(COPYINSERT (BITMAPTEDITOBJ BM 1 0])
```

(COERCETOBITMAP

[LAMBDA (BMSPEC)

(* kbr: "25-Feb-86 17:09")

(* tries to interpret X as a spec for a bitmap.)

```
(PROG (BM CR)
(RETURN (COND
((BITMAPP BMSPEC)
BMSPEC)
[(LITATOM BMSPEC)
(COND
((BITMAPP (EVALV BMSPEC (QUOTE COERCETOBITMAP)
(REGIONP BMSPEC) (* if BMSPEC is a region, treat it as a region of the screen.)
[SETQ BM (BITMAPCREATE (fetch (REGION WIDTH) of BMSPEC)
(fetch (REGION HEIGHT) of BMSPEC)
(BITSPERPIXEL (SCREENBITMAP)
(BITBLT (SCREENBITMAP)
(fetch (REGION LEFT) of BMSPEC)
(fetch (REGION BOTTOM) of BMSPEC)
BM 0 0 NIL NIL (QUOTE INPUT)
(QUOTE REPLACE))
BM)
((type? SCREENREGION BMSPEC) (* if BMSPEC is a screenregion)
[SETQ BM (BITMAPCREATE (fetch (SCREENREGION WIDTH) of BMSPEC)
(fetch (SCREENREGION HEIGHT) of BMSPEC)
```

```

        (BITSPERPIXEL (SCREENBITMAP (fetch (SCREENREGION SCREEN) of BMSPEC)
        (BITBLT (SCREENBITMAP (fetch (SCREENREGION SCREEN) of BMSPEC)
        (fetch (SCREENREGION LEFT) of BMSPEC)
        (fetch (SCREENREGION BOTTOM) of BMSPEC)
        BM 0 0 NIL NIL (QUOTE INPUT)
        (QUOTE REPLACE))
    BM)
    ((WINDOWP BMSPEC)
    (SETQ BM (BITMAPCREATE (WINDOWPROP BMSPEC (QUOTE WIDTH))
        (WINDOWPROP BMSPEC (QUOTE HEIGHT))
        (BITSPERPIXEL BMSPEC))) (* open the window and bring it to the top.)
    (TOTOPW BMSPEC)
    (SETQ CR (DSPCLIPPINGREGION NIL BMSPEC))
    (BITBLT BMSPEC (fetch LEFT of CR)
        (fetch BOTTOM of CR)
        BM 0 0 (fetch WIDTH of CR)
        (fetch HEIGHT of CR))
    BM])

```

(* * Modifications to TEDITSCREEN. (TEDIT.SLOWBLTCHAR is a new function. *)

(DEFINEQ

(TEDIT.BLTCCHAR

```

[LAMBDA (CHARCODE DISPLAYSTREAM CURX DISPLAYDATA DDPILOTBBT CLIPRIGHT)
    (* kbr: "25-Feb-86 22:40")

```

(* Version of \BLTCHAR peculiar to TEdit -- relies on \DISPLAYLINE to make sure things keep working right.)

(* puts a character on a guaranteed display stream. Much of the information needed by the BitBlit microcode is prestored by the routines that change it. This is kept in the BitBlitTable.)

(* knows about the representation of display stream image data)
(* MUST NOT POINT AT A WINDOW'S DISPLAYSTREAM!!!)

(* ASSUMES THAT WE NEVER WANT TO PRINT TO THE LEFT OF ORIGIN 0 ON THE LINE CACHE BITMAP, OR THAT IF WE DO, ALL BETS ARE OFF)

(DECLARE (LOCALVARS . T))

```

(PROG (CHAR8CODE NEWX LEFT RIGHT IMAGEWIDTH)
    (SETQ CHAR8CODE (\CHAR8CODE CHARCODE))

```

```

[COND
    ((NOT (EQ (ffetch (\DISPLAYDATA DDCHARSET) of DISPLAYDATA)
        (\CHARSET CHARCODE)))
    (\CHANGECHARSET.DISPLAY DISPLAYDATA (\CHARSET CHARCODE)
[COND
    ((ffetch (\DISPLAYDATA DDSlowPrintingCase) of DISPLAYDATA)
    (RETURN (\TEDIT.SLOWBLTCHAR CHARCODE DISPLAYSTREAM CURX DISPLAYDATA DDPILOTBBT CLIPRIGHT])
(SETQ IMAGEWIDTH (\GETBASE (fetch (\DISPLAYDATA DDCHARIMAGEWIDTHS) of DISPLAYDATA)
    (\CHAR8CODE CHARCODE)))
(SETQ NEWX (IPLUS CURX IMAGEWIDTH))
(SETQ LEFT (IMAX 0 CURX))
(SETQ RIGHT (IMIN CLIPRIGHT NEWX))
(COND
    ((ILESSP LEFT RIGHT)
    (UNINTERRUPTABLY
        (freplace (PILOTBBT PBTDESTBIT) of DDPILOTBBT with LEFT)
        (* Set up the bitblt-table source left)
        (freplace (PILOTBBT PBTWIDTH) of DDPILOTBBT with (IMIN IMAGEWIDTH (IDIFFERENCE RIGHT LEFT)))
        (freplace (PILOTBBT PBTSOURCEBIT) of DDPILOTBBT with (\GETBASE (fetch (\DISPLAYDATA DDOFFSETSACACHE)
            of DISPLAYDATA)
            (\CHAR8CODE CHARCODE)))
        (\PILOTBITBLT DDPILOTBBT 0))
    (RETURN T])

```

(TEDIT.SLOWBLTCHAR

```

[LAMBDA (CHARCODE DISPLAYSTREAM CURX DISPLAYDATA DDPILOTBBT CLIPRIGHT)
    (* kbr: "25-Feb-86 22:40")

```

(* Version of \SLOWBLTCHAR peculiar to TEdit -- relies on \DISPLAYLINE to make sure things keep working right. Does not handle rotated fonts.)

```

(PROG (CHAR8CODE NEWX LEFT RIGHT IMAGEWIDTH DESTBIT WIDTH SOURCEBIT)
    (SETQ CHAR8CODE (\CHAR8CODE CHARCODE))
    (SETQ IMAGEWIDTH (\GETBASE (fetch (\DISPLAYDATA DDCHARIMAGEWIDTHS) of DISPLAYDATA)
        (\CHAR8CODE CHARCODE)))
    (SETQ NEWX (IPLUS CURX IMAGEWIDTH))
    (SETQ LEFT (IMAX 0 CURX))
    (SETQ RIGHT (IMIN CLIPRIGHT NEWX))
    (COND
        ((ILESSP LEFT RIGHT)
        (SETQ DESTBIT LEFT)
        (SETQ WIDTH (IDIFFERENCE RIGHT LEFT))

```

```
(SETQ SOURCEBIT (IDIFFERENCE (IPLUS (\DSPGETCHAROFFSET CHAR8CODE DISPLAYDATA)
                                LEFT)
                                CURX))
(SELECTQ (ffetch (BITMAP BITMAPBITSPIXEL) of (ffetch (\DISPLAYDATA DDDestination) of DISPLAYDATA))
(1)
(4 (SETQ DESTBIT (LLSH DESTBIT 2))
   (SETQ WIDTH (LLSH WIDTH 2))
   (SETQ SOURCEBIT (LLSH SOURCEBIT 2)))
(8 (SETQ DESTBIT (LLSH DESTBIT 3))
   (SETQ WIDTH (LLSH WIDTH 3))
   (SETQ SOURCEBIT (LLSH SOURCEBIT 3)))
(24 (SETQ DESTBIT (ITIMES 24 DESTBIT))
     (SETQ WIDTH (ITIMES 24 WIDTH))
     (SETQ SOURCEBIT (ITIMES 24 SOURCEBIT)))
(SHOULDNT))
(.WHILE.TOP.DS. DISPLAYSTREAM (freplace (PILOTBBT PBTDESTBIT) of DDPILOTBBT with DESTBIT)
 (freplace (PILOTBBT PBTWIDTH) of DDPILOTBBT with WIDTH)
 (freplace (PILOTBBT PBTSOURCEBIT) of DDPILOTBBT with SOURCEBIT)
 (\PILOTBITBLT DDPILOTBBT 0))
(RETURN T])
```

(\TEDIT.CREATE.LINECACHE

```
[LAMBDA (#CACHES BITSPIXEL)
(* kbr: "25-Feb-86 18:47")
(* Create a linked-together set of LINECACHES, for saving line images.)
(PROG [(CACHES (for I from 1 to #CACHES collect (create LINECACHE
LCBITMAP (BITMAPCREATE 100 15 BITSPIXEL)
[for CACHE on CACHES do
(replace LCNEXTCACHE of (CAR CACHE) with (OR (CADR CACHE)
(CAR CACHES])
(RETURN CACHES])
```

(\TEDIT.LINECACHE

```
[LAMBDA (CACHE WIDTH HEIGHT)
(* kbr: "25-Feb-86 18:48")
(* Given a candidate line cache, return the bitmap, making sure it's at least WIDTH by HEIGHT big.)
(PROG ((BITMAP (fetch LCBITMAP of CACHE))
CW CH)
(SETQ CW (fetch BITMAPWIDTH of BITMAP))
(SETQ CH (fetch BITMAPHEIGHT of BITMAP))
(COND
((AND (IGEQ CW WIDTH)
(IGEQ CH HEIGHT))
(RETURN BITMAP))
(T (RETURN (replace LCBITMAP of CACHE with (BITMAPCREATE (IMAX CW WIDTH)
(IMAX CH HEIGHT)
(BITSPERPIXEL BITMAP))
```

(* * Modifications to TEDITWINDOW. *)

(DEFINEQ

(\TEDIT.MINIMAL.WINDOW.SETUP

```
[LAMBDA (WINDOW TEXTOBJ TEXTSTREAM PROPS AFTERWINDOW)
(* kbr: "25-Feb-86 18:46")
(* Do the absolute minimum setup so that TEXTOBJ and WINDOW know about each other.
Does NOT include mouse interface or scrolling.)
(* If AFTERWINDOW is non-NIL, the new window will be placed after AFTERWINDOW in the TEXTOBJ's list.
This lists us maintain an ordering of windows, for splitting and unsplitting.)
(PROG ((SEL (fetch SEL of TEXTOBJ))
TEDITPROMPTWINDOW DS PROP TWIDTH THEIGHT LINES OLDWINDOWS)
(OR WINDOW (\ILLEGAL.ARG WINDOW))
(replace (TEDITCARET TCCARETDS) of (COND
[(LISTP (fetch CARET of TEXTOBJ))
(CAR (FLAST (fetch CARET of TEXTOBJ)
(T (fetch CARET of TEXTOBJ))
with (WINDOWPROP WINDOW (QUOTE DSP)))
(replace SELWINDOW of TEXTOBJ with WINDOW)
(WINDOWPROP WINDOW (QUOTE PROCESS)
NIL)
(WINDOWPROP WINDOW (QUOTE TEDIT.PROPS)
PROPS)
(WINDOWPROP WINDOW (QUOTE TEXTSTREAM)
TEXTSTREAM)
(WINDOWPROP WINDOW (QUOTE TEXTOBJ)
TEXTOBJ)
(WINDOWPROP WINDOW (QUOTE TEDIT.CURSORREGION)
(LIST 0 0 0 0))
(* The displaystream for flashing the caret)
(* For the moment, this window has no process)
(* Put the props on the window for others ...
**this should go**)
(* Save the text stream for the user to get at via the window.)
(* Give a handle on the TEXTOBJ for the text being edited.)
(* Used by CursorMovedFn)
```

```

(WINDOWPROP WINDOW (QUOTE CURSORMOVEDFN)
 (FUNCTION TEDIT.CURSORMOVEDFN))
(WINDOWPROP WINDOW (QUOTE CURSOROUTFN)
 (FUNCTION TEDIT.CURSORMOVEDFN))
(SETQ DS (WINDOWPROP WINDOW (QUOTE DSP)))
(DSPRIGHTMARGIN 32767 DS)
(SETQ OLDWINDOWS (fetch \WINDOW of TEXTOBJ))
[replace \WINDOW of TEXTOBJ with (COND
  [(LISTP OLDWINDOWS) (* There are windows already. Add this to the list.)
  (COND
    [AFTERWINDOW (* We know which window to put it after.
      Put it there)
      (RPLACD (FMEMB AFTERWINDOW OLDWINDOWS)
        (CONS WINDOW (CDR (FMEMB AFTERWINDOW OLDWINDOWS))
          (* Otherwise, just add it at the end of the list)
        )
      ]
    (T
      (NCONC1 OLDWINDOWS WINDOW)
      (WINDOW (LIST WINDOW)
        ]
  ]
[replace DISPLAYCACHE of TEXTOBJ with (CAR (\TEDIT.CREATE.LINECACHE 1 (BITSPERPIXEL WINDOW)
 (* and a CACHE for creating line images for display)
[replace DISPLAYCACHEDS of TEXTOBJ with (DSPCREATE (fetch LCBITMAP of (fetch DISPLAYCACHE of TEXTOBJ)
 (* A displaystream for changinge the image caches)
(DSPOPERATION (QUOTE PAINT)
 (fetch DISPLAYCACHEDS of TEXTOBJ))
(DSPCLIPPINGREGION (create REGION
  LEFT _ 0
  BOTTOM _ 0
  WIDTH _ 100
  HEIGHT _ 15)
 (fetch DISPLAYCACHEDS of TEXTOBJ))
 (* Remember its size, too.)
(COND
 ((SETQ PROP (LISTGET PROPS (QUOTE REGION)))
 (* The caller wants to set a region.
  Use his)
  (replace WTOP of TEXTOBJ with (fetch PTOP of PROP))
  (replace WRIGHT of TEXTOBJ with (fetch RIGHT of PROP))
  (replace WBOTTOM of TEXTOBJ with (fetch BOTTOM of PROP))
  (replace WLEFT of TEXTOBJ with (fetch LEFT of PROP)))
 (T
  (* Otherwise, default to the whole window)
  (replace WLEFT of TEXTOBJ with 0)
  (replace WBOTTOM of TEXTOBJ with 0)
  (replace WTOP of TEXTOBJ with (fetch HEIGHT of (DSPCLIPPINGREGION NIL DS)))
  (replace WRIGHT of TEXTOBJ with (fetch WIDTH of (DSPCLIPPINGREGION NIL DS)))
(SETQ LINES (\SHOWTEXT TEXTOBJ NIL WINDOW))
(WINDOWPROP WINDOW (QUOTE LINES)
 LINES)
 (* Display the text in the window, for later use.)
[replace LINES of TEXTOBJ with (COND
 [AFTERWINDOW (for LINE in (fetch LINES of TEXTOBJ) as WINDOW in OLDWINDOWS
  join (COND
    ((EQ WINDOW AFTERWINDOW)
      (LIST LINE LINES))
    (T (LIST LINE)
      ]
    ((LISTP (fetch LINES of TEXTOBJ))
      (NCONC1 (fetch LINES of TEXTOBJ)
        LINES))
    (LINES (LIST LINES)
      ]
  ]
(\FIXSEL SEL TEXTOBJ)
(TEDIT.NORMALIZECARET TEXTOBJ SEL)
(\SHOWSEL SEL NIL T)
(\TEDIT.SET.WINDOW.EXTENT TEXTOBJ WINDOW)
(\COPYSEL SEL TEDIT.SELECTION])
)

```

FUNCTION INDEX

BITMAPOBJ.SNAPW	1	\TEDIT.BLTCHAR	2	\TEDIT.SLOWBLTCHAR	2
COERCETOBITMAP	1	\TEDIT.CREATE.LINECACHE	3		
TEDIT.MINIMAL.WINDOW.SETUP	3	\TEDIT.LINECACHE	3		
