## renderBufferWinGDI cpccCSS hBitmapMemory :HBITMAP cpccTextStylesMac hDCmemoryBuffer :HDC bgColor :cpccProperty<cpccColor> hOldBitmap :HBITMAP color :cpccProperty<cpccColor> setColor(NSM utableDictionary\*, cpccColor\*) :void fontName :cpccProperty<cpcc\_string> setFont(NSMutableDictionary\*, cpcc\_char\*, int) :void dc() :HDC fontSize :cpccProperty<int> setParagraph (NSM utableDictionary\*, int):void renderBufferWinGDI(HDC, int, int textAlign :cpccProperty<TcssTextAlignValue ~renderBufferWinGDI() -m\_style Helper -m\_renderBuffer cpccWindow Base bgColor :cpccStackWithDefault<cpccColor> drawColor :cpccStackWithDefault<cpccColor> fontName :cpccStackWithDefault<cpcc\_string> fontQuality :cpccStackWithDefault<eFontQuality> fontSize :cpccStackWithDefault<int> fontWeight :cpccStackWithDefault<eFontWeight> $\verb"m_windowHandle": \verb"cpccNativeWindowHandle"$ textAlign :cpccStackWithDefault<int> textSpacing :cpccStackWithDefault<int> BitBilt(int, int, cpccNativeDrawSurfaceHandle, int, int, cpccColor\*) :void clear():void cpccWindowBase(cpccNativeWindowHandle) ~cp ccWindow Base () fill WithColor(cpccColor&) :void flush() :void getHeight():int getNativeWindowHandle() :cpccNativeWindowHandle getPixeI(int, int) :cpccColor getTextSize(cpcc\_char\*, int\*, int\*) :void getWidth() :int popCss(cpccCSS\*) :void pushCss(cpccCSS\*) :void setPixel(int, int, cpccColor&) :void textOut(int, int, cpcc\_char\*, cpccCSS\*) :void cpccWindowWin cpccWindow Mac m\_DrawDC :HDC m\_drawSurface :CGContextRef m\_rect :RECT m\_skewX :float {readOnly} m\_renderBuffer :renderBufferWinGDI\* m\_skewY :float {readOnly} m\_WindowDC :HDC m\_windowRect:NSRect BitBilt(int, int, HDC, int, int, cpccColor\*) :void BitBilt(int, int, NSImage\*, int, int, cpccColor\*) :void cpccWindowWin(HWND) cpccWindowMac(NSView\*) ~cpccWindowWin() fillWithColor\_impl(cpccColor&) :void fillWithColor\_impl(cpccColor&) :void flush():void flush():void getHeight():int «typedef» getHeight():int getPixel\_impl(int, int) :cpccColor window Impl getNativeWindowHandle():HWND getTextSize(cpcc\_char\*, int\*, int\*) :void getPixel\_impl(int, int) :cpccColor getWidth():int getTextSize(cpcc\_char\*, int\*, int\*) :void setPixel\_impl(int, int, cpccColor&) :void getWidth():int textOut\_impl(int, int, cpcc\_char\*) :void setPixel\_impl(int, int, cpccColor&) :void textOut\_impl(int, int, cpcc\_char\*) :void cpccWindow fontGDI clear():void cpccWindow(cpccNativeWindowHandle) m\_hDC :HDC fillWithColor(cpccColor&) :void m\_hFont :HFONT getPixel(int, int) :cpccColor m\_hFontOld :HFONT popCss(cpccCSS\*) :void pushCss(cpccCSS\*) :void fontGDI(HDC, cpcc\_char\*, int, eFontWeight, eFontQuality) setPixel(int, int, cpccColor&) :void ~fontGDI() textOut(int, int, cpcc\_char\*, cpccCSS\*) :voi