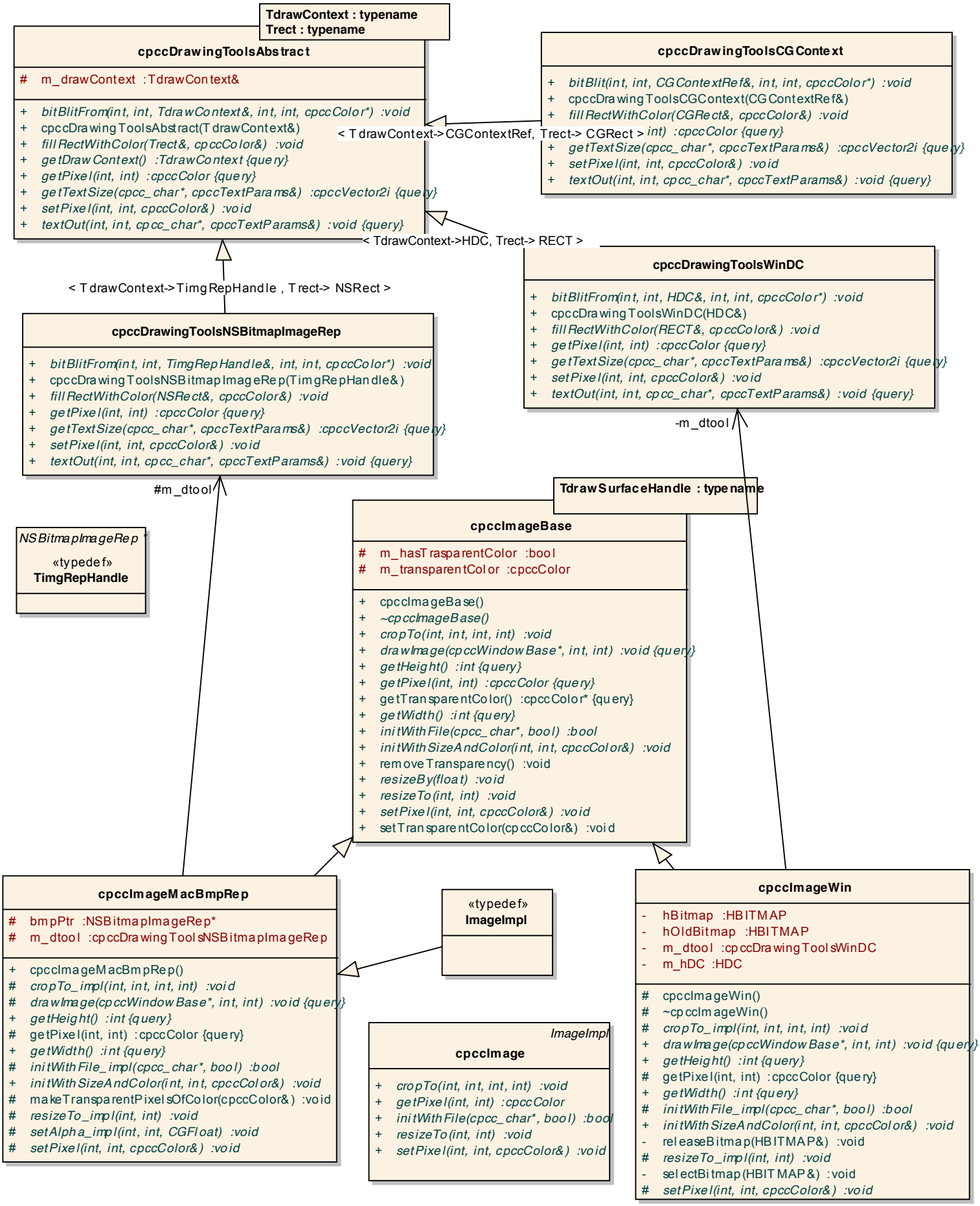
TdrawContext : typename
Trect : typename
cpccDrawingToolsAbstract
m_drawContext :TdrawContext&
+ bitBlitFrom(int, int, TdrawContext&, int, int, cpccColor*) :void
+ cpccDrawingToolsAbstract(TdrawContext&)
+ fillRectWithColor(Trect&, cpccColor&) :void
+ getDrawContext() :TdrawContext {query}
+ getPixel(int, int) :cpccColor {query}
+ getTextSize(cpcc_char*, cpccTextParams&) :cpccVector2i {query}
+ setPixel(int, int, cpccColor&) :void
+ textOut(int, int, cpcc_char*, cpccTextParams&) :void {query}
cpccDrawingToolsCGContext
+ bitBlit(int, int, CGContextRef&, int, int, cpccColor*) :void
+ cpccDrawingToolsCGContext(CGContextRef&)
+ fillRectWithColor(CGRect&, cpccColor&) :void
+ getTextSize(cpcc_char*, cpccTextParams&) :cpccVector2i {query}
+ setPixel(int, int, cpccColor&) :void
+ textOut(int, int, cpcc_char*, cpccTextParams&) :void {query}
< TdrawContext->CGContextRef, Trect-> CGRect >
< TdrawContext->HDC, Trect-> RECT >
cpccDrawingToolsWinDC
+ bitBlitFrom(int, int, HDC&, int, int, cpccColor*) :void
+ cpccDrawingToolsWinDC(HDC&)
+ fillRectWithColor(RECT&, cpccColor&) :void
+ getPixel(int, int) :cpccColor {query}
+ getTextSize(cpcc_char*, cpccTextParams&) :cpccVector2i {query}
+ setPixel(int, int, cpccColor&) :void
+ textOut(int, int, cpcc_char*, cpccTextParams&) :void {query}
-m_dtool
< TdrawContext->TimgRepHandle , Trect-> NSRect >
cpccDrawingToolsNSBitmapImageRep
+ bitBlitFrom(int, int, TimgRepHandle&, int, int, cpccColor*) :void
+ cpccDrawingToolsNSBitmapImageRep(TimgRepHandle&)
+ fillRectWithColor(NSRect&, cpccColor&) :void
+ getPixel(int, int) :cpccColor {query}
+ getTextSize(cpcc_char*, cpccTextParams&) :cpccVector2i {query}
+ setPixel(int, int, cpccColor&) :void
+ textOut(int, int, cpcc_char*, cpccTextParams&) :void {query}
#m_dtool
NSBitmapImageRep *
«typedef»
TimgRepHandle
TdrawSurfaceHandle : typename
cpccImageBase
m_hasTrasparentColor :bool
m_transparentColor :cpccColor
+ cpccImageBase()
+ ~cpccImageBase()
+ cropTo(int, int, int, int) :void
+ drawImage(cpccWindowBase*, int, int) :void {query}
+ getHeight() :int {query}
+ getPixel(int, int) :cpccColor {query}
+ getTransparentColor() :cpccColor* {query}
+ getWidth() :int {query}
+ initWithFile(cpcc_char*, bool) :bool
+ initWithSizeAndColor(int, int, cpccColor&) :void
+ removeTransparency() :void
+ resizeBy(float) :void
+ resizeTo(int, int) :void
+ setPixel(int, int, cpccColor&) :void
+ setTransparentColor(cpccColor&) :void
cpccImageMacBmpRep
bmpPtr :NSBitmapImageRep*
m_dtool :cpccDrawingToolsNSBitmapImageRep
+ cpccImageMacBmpRep()
# cropTo_impl(int, int, int, int) :void
# drawImage(cpccWindowBase*, int, int) :void {query}
+ getHeight() :int {query}
# getPixel(int, int) :cpccColor {query}
+ getWidth() :int {query}
# initWithFile_impl(cpcc_char*, bool) :bool
+ initWithSizeAndColor(int, int, cpccColor&) :void
# makeTransparentPixelsOfColor(cpccColor&) :void
# resizeTo_impl(int, int) :void
# setAlpha_impl(int, int, CGFloat) :void
# setPixel(int, int, cpccColor&) :void
«typedef»
ImageImpl
cpccImageWin
- hBitmap :HBITMAP
- hOldBitmap :HBITMAP
- m_dtool :cpccDrawingToolsWinDC
- m_hDC :HDC
cpccImageWin()
~cpccImageWin()
cropTo_impl(int, int, int, int) :void
+ drawImage(cpccWindowBase*, int, int) :void {query}
+ getHeight() :int {query}
# getPixel(int, int) :cpccColor {query}
+ getWidth() :int {query}
# initWithFile_impl(cpcc_char*, bool) :bool
+ initWithSizeAndColor(int, int, cpccColor&) :void
- releaseBitmap(HBITMAP&) :void
resizeTo_impl(int, int) :void
- selectBitmap(HBITMAP&) :void
setPixel(int, int, cpccColor&) :void
ImageImpl
cpccImage
+ cropTo(int, int, int, int) :void
+ getPixel(int, int) :cpccColor
+ initWithFile(cpcc_char*, bool) :bool
+ resizeTo(int, int) :void
+ setPixel(int, int, cpccColor&) :void