
Subject: Re: MILESTONE: gtk3 replaces gtk2 as default linux backend

Posted by [Tom1](#) on Mon, 20 Apr 2020 18:59:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

One Win32 question: Is it really necessary to cache the image if it is actually going to be drawn as a colored rect or via SetSurface?

Best regards,

Tom

[EDIT] Did some testing and this is faster than using SetSurface after caching:

```
void SystemDraw::SysDrawImageOp(int x, int y, const Image& img, const Rect& src, Color color)
{
    GuiLock __;
    if(img.GetLength() == 0)
        return;
    LLOG("SysDrawImageOp " << img.GetSerialId() << " " << img.GetSize());

    // Insert this optimization here:
    int kind = img.GetKindNoScan();
    if(kind == IMAGE_OPAQUE && !IsNull(color)) {
        Size sz=img.GetSize();
        DrawRect(x, y, sz.cx, sz.cy, color);
        return;
    }
    if(kind == IMAGE_OPAQUE && (GetDeviceCaps(GetHandle(), RASTERCAPS) &
RC_DIBTODEV)) {
        LTIMING("Image Opaque direct set");
        Size sz=img.GetSize();
        SetSurface(*this, x, y, sz.cx, sz.cy, ~img);
        return;
    }
    // End of insertion

    ImageSysDataMaker m;
    ...
}
```

Or is there some other reason to cache the image before display?
