

---

Subject: Help Propblem work buffer

Posted by [situkhan](#) on Mon, 17 Aug 2015 11:09:01 GMT

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

---

```
void AboutForm::Paint(Draw& w)
{
    if(m_Tick++ > 10)
    {
        m_Tick = 0;
        m_Water.WarpBlob(rand()%320,rand()%512,rand()%320,rand()%302,m_Water.m_iHpage);
    }
    m_Water.Fdensity = 5;
    m_Water.FbDrawWithLight = FALSE;
    m_Water.FbCalcBigFilter = FALSE;

    /// DWORD m_ImageIn[320*512],m_ImageOut[320*512];
    m_Water.Render(m_ImageIn,m_ImageOut);

    memcpy((void*)m_DestImage.Begin(),(void*)m_ImageOut,320*512*4); /// Direct

    //w.DrawImage(0,0,320,512, m_DestImage); /// ---- Ceases to work after the first run ???

    w.DrawImage(1,1,320-1,512-1, m_DestImage); /// ---- Working
    SetTimeCallback(10, THISBACK(Render));
}
```

---

---

Subject: Re: Help Propblem work buffer

Posted by [mirek](#) on Tue, 18 Aug 2015 18:07:53 GMT

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

---

situkhan wrote on Mon, 17 August 2015 13:09

```
void AboutForm::Paint(Draw& w)
{
    if(m_Tick++ > 10)
    {
        m_Tick = 0;
        m_Water.WarpBlob(rand()%320,rand()%512,rand()%320,rand()%302,m_Water.m_iHpage);
    }
    m_Water.Fdensity = 5;
    m_Water.FbDrawWithLight = FALSE;
    m_Water.FbCalcBigFilter = FALSE;

    /// DWORD m_ImageIn[320*512],m_ImageOut[320*512];
    m_Water.Render(m_ImageIn,m_ImageOut);

    memcpy((void*)m_DestImage.Begin(),(void*)m_ImageOut,320*512*4); /// Direct
```

```
//w.DrawImage(0,0,320,512, m_DestImage);  /// ---- Ceases to work after the first run ???  
  
w.DrawImage(1,1,320-1,512-1, m_DestImage);  /// ---- Working  
SetTimeCallback(10, THISBACK(Render));  
}
```

Hard to say without seeing more of code. If `m_DestImage` is `Image`, your code is wrong, because you cannot overwrite `Begin` just as you do. If it is `ImageBuffer`, it is wrong too, because conversion `ImageBuffer -> Image` destroys the source.

I can see that you are rendering everything in `DWORDS` anyway and your problem is just to render it on screen. If that is true, you can bypass `Image` altogether - there is nice small function called `SetSurface`

```
void SetSurface(Draw& w, const Rect& dest, const RGBA *pixels, Size srcsz, Point poff)
```

which should exactly solve your problem IMO. Note that `SetSurface` has "backup path" in case that target `Draw` does not support it...

Mirek

---