
Subject: Request: completion 64 bit support in Draw and Stream

Posted by [nixnixnix](#) on Wed, 06 Feb 2013 01:36:36 GMT

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

Hi,

I recently had a user complain that they couldn't save a TIF that they had opened in my software. The TIF is only 120MB but once as an Image it is approx 3000x3000 pixels and therefore around 3.6GB.

This means that the following function

```
void Image::Serialize(Stream& s)
{
    int version = 0;
    s / version;
    Size sz = GetSize();
    Point p = GetHotSpot();
    Size dots = GetDots();
    s % sz % p % dots;
    int len = sz.cx * sz.cy;
    if(s.IsLoading())
        if(len) {
            ImageBuffer b(sz);
            if(!s.GetAll(~b, len * sizeof(RGBA)))
                s.SetError();
            b.SetDots(dots);
            b.SetHotSpot(p);
            *this = b;
        }
    else
        Clear();
    else
        s.Put(~*this, len * sizeof(RGBA));
}
```

needs len to be declared as int64 leading to

```
int64 len = int64(sz.cx) * int64(sz.cy);
```

or something like that. Unfortunately, Stream::Put takes an int rather than an int64 and so the size is reported as less than zero and fails the ASSERT(size>=0) on Stream.h line 89.

I realise that I could re-write Image::Serialize to save Image in chunks but then you guys are so much better than me at this and it needs to be a permanent solution as we are all moving eventually to 64 bit and large memory items. Some of the code in Stream looks like it is intended

for 64 bit.

I hope my detailed diagnostic helps.

Stream::GetAll will need upgrading as well to use int64. If no-one has time for this I can try to suggest a patch but I figure my code is unlikely to be the long-term solution.

Cheers,

Nick

Subject: Re: Request: completion 64 bit support in Draw and Stream

Posted by [nixnixnix](#) on Wed, 06 Feb 2013 14:13:19 GMT

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

Here is my patch.

```
void Image::Serialize(Stream& s)
{
    int version = 0;
    s / version;
    Size sz = GetSize();
    Point p = GetHotSpot();
    Size dots = GetDots();
    s % sz % p % dots;
    int64 len = (int64)sz.cx * (int64)sz.cy * (int64)sizeof(RGBA);
    if(s.IsLoading())
        if(len) {
            ImageBuffer b(sz);

            int64 offset = 0;
            const byte* ptr = (byte*)~b;

            while(len>INT_MAX)
            {
                if(!s.GetAll((void*)(ptr+offset), INT_MAX))
                {
                    s.SetError();
                    return;
                }
                len -= INT_MAX;
            }
        }
}
```

```

    offset += INT_MAX;
}
if(!s.GetAll((void*)(ptr+offset), len))
    s.SetError();

b.SetDots(dots);
b.SetHotSpot(p);
*this = b;
}
else
    Clear();
else
{
    int64 offset = 0;
    const byte* ptr = (byte*)~*this;

    while(len>INT_MAX)
    {
        s.Put(ptr+offset, INT_MAX);
        len -= INT_MAX;
        offset += INT_MAX;
    }
    s.Put(ptr+offset, len);
}
}

```

I realised that what I was asking would necessitate a lot of rewriting of other code in UPP. I have tested this and it works with an image of 3.45GB.

Can we make this change in the SVN please?

Cheers,

Nick

Subject: Re: Request: completion 64 bit support in Draw and Stream

Posted by [mirek](#) on Thu, 07 Feb 2013 16:35:22 GMT

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

nixnix wrote on Wed, 06 February 2013 09:13 Here is my patch.

```

void Image::Serialize(Stream& s)
{
    int version = 0;
    s / version;
}

```

```

Size sz = GetSize();
Point p = GetHotSpot();
Size dots = GetDots();
s % sz % p % dots;
int64 len = (int64)sz.cx * (int64)sz.cy * (int64)sizeof(RGBA);
if(s.IsLoading())
if(len) {
    ImageBuffer b(sz);

    int64 offset = 0;
    const byte* ptr = (byte*)~b;

    while(len>INT_MAX)
    {
        if(!s.GetAll((void*)(ptr+offset), INT_MAX))
        {
            s.SetError();
            return;
        }
        len -= INT_MAX;
        offset += INT_MAX;
    }
    if(!s.GetAll((void*)(ptr+offset), len))
        s.SetError();

    b.SetDots(dots);
    b.SetHotSpot(p);
    *this = b;
}
else
    Clear();
else
{
    int64 offset = 0;
    const byte* ptr = (byte*)~*this;

    while(len>INT_MAX)
    {
        s.Put(ptr+offset, INT_MAX);
        len -= INT_MAX;
        offset += INT_MAX;
    }
    s.Put(ptr+offset, len);
}
}

```

I realised that what I was asking would necessitate a lot of rewriting of other code in UPP. I have

tested this and it works with an image of 3.45GB.

Can we make this change in the SVN please?

Cheers,

Nick

It is in svn now. I think I will add direct support to Stream soon.

I also think that you might consider whether you really need to store the image expanded. Note that you can e.g. draw compressed TIFF in Paint with some effort. Or you can rescale it to something smaller without decompressing original.

Mirek

Subject: Re: Request: completion 64 bit support in Draw and Stream

Posted by [nixnixnix](#) on Thu, 07 Feb 2013 20:33:31 GMT

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

Thanks Mirek,

I was wondering about that. I deliberately split file format from data objects in openWind and so a TIFF specific solution would not be desirable. My users tend to have lots of RAM so speed is more important.

However, it would be good if I could compress on saving. Then again the best solution would be to compress the entire workbook as it is saved and decompress on reload.

I should be back home in Canada next week and will email you and Koldo about this and other issues then.

Thanks again (I think this is might be my first direct contribution to UPP code - it's taken me a while)

Nick

Subject: Re: Request: completion 64 bit support in Draw and Stream

Posted by [koldo](#) on Thu, 07 Feb 2013 22:03:52 GMT

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