EncoderTransformation ignored, no errors produced.

Source:
http://www.tech−archive.net/Archive/Development/microsoft.public.win32.programmer.gdi/2008−03/msg00044.html

- *From*: "jason.cipriani@xxxxxxxxx" <jason.cipriani@xxxxxxxxx>
- *Date*: Thu, 13 Mar 2008 22:58:13 −0700 (PDT)

I have some uncompressed image data in memory that is vertically flipped, and I would like to use GDI+ to save it as a JPEG image, and I would also like to use the EncoderTransformation encoder parameter to vertically flip the saved image.

At the end of this post is the source code for a simple console app I wrote to test this; the entire VS2005 project is also available here (sorry, Google Groups does not allow attachments here):

http://www.sendspace.com/file/300z1r

The code creates a test image in memory −− a 160x160 image with a white pixel at (30,30). It then creates a Gdiplus::Bitmap using that image data, and writes out two jpeg files:

fliptest−identity.jpg with no transformation,
fliptest−vertflip.jpg vertically flipped.

I have two questions:

1) The two output images are identical, and vertical flipping is being ignored. Why?

2) What is the correct way to allocate space for EncoderParameters (in that code I use placement new into a buffer that's probably big enough to hold all the parameters).

Question 1 has been frustrating me; the JPEG is written with no errors but the EncoderTransformation parameter is ignored. It does not matter if I change the order of the parameters.

Thanks,
Jason

−−−−−− fliptest.cpp −−−−−−

```
#include <windows.h>
```

```
#include <gdiplus.h>
#include <stdio.h>
#include <new>

using namespace Gdiplus;


// straight from MSDN example docs
static int GetEncoderClsid(const WCHAR* format, CLSID* pClsid)
{
UINT num = 0; // number of image encoders
UINT size = 0; // size of the image encoder array in bytes

ImageCodecInfo* pImageCodecInfo;

GetImageEncodersSize(&num, &size);
if(size == 0)
return −1; // Failure

pImageCodecInfo = (ImageCodecInfo*)(malloc(size));
if(pImageCodecInfo == NULL)
return −1; // Failure

GetImageEncoders(num, size, pImageCodecInfo);

for(UINT j = 0; j < num; ++j) {
if( wcscmp(pImageCodecInfo[j].MimeType, format) == 0 ) {
*pClsid = pImageCodecInfo[j].Clsid;
free(pImageCodecInfo);
return j; // Success
}
}

free(pImageCodecInfo);
return −1; // Failure
}


// write JPEG; possibly flip vertically.
void SaveJPEG (Bitmap *b, const WCHAR *filename, bool vflip) {

CLSID encoderClsid;
ULONG quality, depth, transform;
Status st;
int index;

// encparams is big enough for 3 encoder parameters i'm assuming.
// question 1: what's the right way to do this?
char encparams[2000];
EncoderParameters *encoderParameters = new (encparams)
EncoderParameters;
```

```
// Get the CLSID of the JPEG encoder.
index = GetEncoderClsid(L"image/jpeg", &encoderClsid);
if (index == −1) {
printf("GetEncoderClsid() failed.\n");
return;
}

// Set encoder options, vertical flip if requested.

encoderParameters−>Count = vflip ? 3 : 2;

quality = 100;
encoderParameters−>Parameter[0].Guid = EncoderQuality;
encoderParameters−>Parameter[0].Type = EncoderParameterValueTypeLong;
encoderParameters−>Parameter[0].NumberOfValues = 1;
encoderParameters−>Parameter[0].Value = &quality;

depth = 24;
encoderParameters−>Parameter[1].Guid = EncoderColorDepth;
encoderParameters−>Parameter[1].Type = EncoderParameterValueTypeLong;
encoderParameters−>Parameter[1].NumberOfValues = 1;
encoderParameters−>Parameter[1].Value = &depth;

if (vflip) {
// question 2: why is this ignored?
transform = EncoderValueTransformFlipVertical;
encoderParameters−>Parameter[2].Guid = EncoderTransformation;
encoderParameters−>Parameter[2].Type =
EncoderParameterValueTypeLong;
encoderParameters−>Parameter[2].NumberOfValues = 1;
encoderParameters−>Parameter[2].Value = &transform;
}

// Write.
st = b−>Save(filename, &encoderClsid, encoderParameters);
printf("%ls: %s\n", filename, (st == Ok) ? "OK" : "FAIL");

// from placement new
encoderParameters−>~EncoderParameters();

}


int main (int, char **) {

GdiplusStartupInput _gdiplusStartupInput;
ULONG_PTR _gdiplusToken;
GdiplusStartup(&_gdiplusToken, &_gdiplusStartupInput, NULL);

// create test Bitmap, black with a white pixel at (30,30).
```

```
unsigned char data[160][160][3];
memset(data, 0, sizeof(data));
data[30][30][0] = 255;
data[30][30][1] = 255;
data[30][30][2] = 255;
Bitmap *img = new Bitmap(160, 160, 3 * 160, PixelFormat24bppRGB,
(BYTE *)data);

// save as jpeg, no transformation
SaveJPEG(img, L"fliptest−identity.jpg", false);

// save as jpeg, vertically flipped.
SaveJPEG(img, L"fliptest−vertflip.jpg", true);

delete img;
GdiplusShutdown(_gdiplusToken);
getchar();
return 0;

}
.
```