Subject: Re: How to efficiently update a large Image?
Posted by mirek on Mon, 27 Jul 2009 15:15:15 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Mon, 27 July 2009 10:47Hi Mirek,

SetKind(IMAGE_OPAQUE) gives marginal improvement.

Now I see the visit from Image to ImageBuffer and back to Image does not take more than about 1.5 ms. However, after this visit the DrawImage does something more than usual and becomes slow. It takes about 30 ms to update a small stripe of an image using DrawImage whereas without the ImageBuffer visit, this takes less than a millisecond.

It is because the raster needs to be converted to windows object (HBMP) that then can be painted on the screen.

That said, Image is really optimized for thousands of small images, not single big one...

That said (2), there is SetSurface function (which is sort of semi-public), maybe it can be helpful somehow in this case.

Another option is to divide the big image into many smaller ones.

Quote:
I also noticed that the application allocates and deallocates several megabytes worth of memory when running probably because of what is happening around this image or imagebuffer.

Most like HBMP associated data.

Quote:
Is there any way to directly update the Image pixel contents without ImageBuffer and the associated DrawImage overhead?

No, but I believe it should not really be a problem. Image->ImageBuffer->Image is extremely fast. The problem is that it is necessarry to move those pixels to GDI afterwards.

Mirek