Subject: Finally there... new Image "commited"...
Posted by mirek on Mon, 15 May 2006 12:47:35 GMT
View Forum Message <> Reply to Message

Recent tree is now freezed as "upp605" repository.

Short overview of new architecture:

Image has ALWAYS RGBA format. Alpha channel is now supported.

ImageBuffer - similar to StringBuffer, "writable" Image

Image - same as now, raster image that can be displayed by Draw

Raster - represents source of raster data, e.g. PNG file or PNG file in memory

ImageRaster - Raster of Image

StreamRaster - stream based raster, usually image file

RasterEncoder - "writer" of raster data - output stream of scanlines

ImageEncoder - RasterEncoder that produces Image

RasterStreamEncoder - base of file format encoders

Various image operations are available (sharpen/blur, color adjustments etc...), the often used basic form is "Raster->RasterEncoder".

MakeImage: Image operations can be registered by name, MakeImage creates image based on some input parameters (usually Image) and "cooking receipt", e.g.:

MakeImage(CtrlImg::exclamation, "rescale 100, 100; grayscale");

The thing is that MakeImage contains cache, so results can be cached (ideal for future chamemelon issues).

Another significant improvement of Draw:

void DrawData(int x, int y, int cx, int cy, const String& data, const char *type);

- this draws binary (or whatever) data with specified type. E.g.

w.DrawData(0, 0, 100, 100, LoadFile("d://file.png"), "image");

The most important part of this is that it works with Drawing (gets stored and rescaled as needed) and even has optimization support for printing (paints by bands, detects areas of uniform colors). In other words, it is a tool necessary to get AGG and other software renderers working.

To add your own DrawData format, you need to implement and register "DataDrawer" class.

Various other improvements were done too; e.g. Linux now supports images on clipboard and even RTF on clipboard.

Mirek
