Subject: Re: let's discuss new Draw principles and problems...
Posted by mirek on Mon, 13 Feb 2006 12:41:27 GMT
View Forum Message <> Reply to Message

Just to kickstart this discussion, I would like to list problems that I hope to solve:

- we need to enforce RGBA format as the dominant Image format. Current draw does not support alpha at any level. Host platforms support alpha images for raster bitmaps (starting with Win98).

- new Image will need new Icon designer.

- Text rendering should be redesigned, separating font information from Draw. Implementation should be improved to synthetise characters missing in Linux fonts.

Now of course comes evergreen question about advanced rendering capabilities. IMHO, this is two-headed problem:

- host platform support part - while Cairo and GDI+ are of course very nice, they fail to provide "runs out of box" guarantee

- meanwhile, dominant number of applications does not need advanced rendering at all

Makes me think there are two possibilities of advanced rendering (in fact, they are not mutually exclusive, so both can be sooner or later implemented):

- client-side advanced rendering, using direct access to the new RGBA surfaces. This has disadvantage of being quite a lot of work to do and not using hardware acceleration (OTOH, software can be pretty fast there as well). Advantage is that it would not require anything from the host platform, it would even allow using Draw on GUI-less machines (good for web servers). In fact, using some existing advanced rendering library here looks like quite a good option too (AGG comes to mind).

- using advanced rendering of host platform. That would involve having separate "DrawEx" package that would provide, for applications that need so, interface for host platform advanced rendering ops (GDI+, Cairo).

Mirek