

---

Subject: Re: U++ talk

Posted by [cbpporter](#) on Thu, 04 Sep 2008 02:52:55 GMT

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

---

This could be a possible list of packages to go into the first .so and also some thoughts regarding them:

- Core
- Crypto (also, add description to Crypt and maybe AES - I still have a textbook AES somewhere if you're interested)
- Esc (Esc is a pretty nice minimal scripting language, perfect to embed in your code, so we could include it. OTOH, a plan for the future: make Esc a strict subset of an existing and established scripting language, so that Esc works in that language, but not the other way around necessarily. There is a myriad of scripting languages for any given purpose, and I don't think we or the world needs a new one. Or maybe replace Esc with something small like Lua, but Lua is not my first choice for what Esc should accomplish)
- Geom (also maybe clean-up and decide future of it)
- Geom\Coords (this package depends on TCore package, which is empty; also same clean-up as with Geom)
- plugin/astyle - maybe (add description)
- plugin/bz2
- plugin/dbf - maybe, I don't know exactly what it is and if it is related to the old database format, in which case we may decide move it to DB .so (also add description)
- plugin/ftp
- plugin/pcr - I didn't even know we had regexp, this should be advertised more (add description)
- plugin/z (also, I think it shouldn't be a requirement for basic Core console app)
- Web and friends (add description, and some parts are in need of some extensions)

This would basically be it. I hope I didn't miss any packages.

The question is what to do about image format packages. Some can be included without Draw, while others can't. The best solution would be to make these plugins work with in-memory images and streams, and registering the encoder/decoder, and Draw stuff being able to work with those images and display them as soon as the required plugin is included. This way it would also work for servers (like the way gdlb is used for this purpose).

We could still squeeze in PNG and maybe some other formats which work without GUI packages right now.

So there would be a first .so with maybe some graphical formats. I see 3 more .sos at least: one for GTK GUI, one for NOGTK GUI (but we do need a new default skin for that, because that look is worst than Gtk 1, which was ugly as hell, and doesn't really fit in with any modern look) and at least a SQL .so.

Also, either build packages for MT, or double the number of .sos.

This is just a proposal though. The important thing is getting them out one by one, and not focus on getting all the .so structures at once. We need some time and get it right, because once you

settle on something, it is going to be harder to change it.

Also, some other crazy ideas: make U++ able to run and draw it's own windows when in a framebuffer like environment. This way it could run on the Linux framebuffer, or in a normal WinAPI window, but in DirectX buffers for games. I did some hobby gamedev in the past and there was always the problem of GUI toolkit able to render itself in the DirectX framebuffer. And if this works, why not make it work in a console, for text mode GUI's. We could call it UltimateVision++ .

---