

---

Subject: Possible solution of "icpp problem"

Posted by [mirek](#) on Sat, 27 Aug 2016 14:09:14 GMT

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

---

Hi,

"icpp issue" is longstanding problem that prevents U++ to be used as 'library' without TheIDE.

Brief summary of issue: if file has .icpp extension, U++ builder treats it differently and always includes corresponding .obj file into executable. Thus it is possible to put initialization code into it (using fake global constructor). That way, e.g. to support gif Image loading, all that has to be done is just add plugin/gif to project. plugin/gif/gif.icpp contains the code that registers gif format in Draw package.

Now if you rename gif.icpp to gif.cpp, build process will produce .obj and it gets into library, but linker will not link it into .exe because there are no references to functions in plugin/gif.

This was for quite a long time a problem when thinking about "U++ library".

Now I got finally a good idea how to resolve this issue:

We can put static constructor instance into "gif.h" calling required initialization code. That way, `#include <plugin/gif/gif.h>` will be required in client code, but that is not that much problem (in most cases, such `#include` will be there anyway) and as library, it is just fitting to include gif header to get gif support.

I can change U++ sources so that .icpp are still supported by U++ builder, but not really required when building outside umk/theide. .icpp will just call the same initialization hook as static constructor in the header (in reality, it will be easier to just call empty function in .cpp that contains INITBLOCK with required initialization...)

What do you think? Should I go that way?

Mirek