Subject: Re: Possible solution of "icpp problem"
Posted by mirek on Sun, 28 Aug 2016 15:49:23 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Sun, 28 August 2016 13:57Hi Mirek,

That would be great! It would also make U++ much easier to use without theide. Using other IDEs or makefiles, cmake, etc. would lower the barrier for new users and could allow many more people to adopt U++.

Mindtraveller wrote on Sun, 28 August 2016 10:07First of all, let's think do we really need to keep icpp-compatibility at all. Or it is possible to forget about them once and for all.I'm definitely for dropping icpp completely, if possible. There are other uses, not only plugin-like stuff (e.g. files with Skylark handlers), but those could be solved in similar way I presume. Could there be a simple utility class for automatic initialization that could be used in all cases, even in user code?

Well, first of all, let us not to forget that the problem we really need to have solved is to force linker to include some .obj files from .lib (or .o from .a ... :) even if none of symbols is referenced from the rest of the code. What I am proposing is basically putting a dummy call from header to such .obj, just to have it 'activated'.

Could be equivalent of

Gif.h

```
struct GifInitializer { GicInitializer() { GifInitializerFunction(); } };

static GifInitializer DummyName;
```

Gif.cpp (former Gif.icpp)

```
void GifInitializerFunction() {}

INITBLOCK {
  ... register GIF in Draw
}
```