
Subject: Re: Possible solution of "icpp problem"
Posted by [mirek](#) on Mon, 26 Sep 2016 15:43:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

kov_serg wrote on Mon, 26 September 2016 17:05How about another explicit method

Generate in project dir file like this

```
// static_init.cpp
#define INIT(lib) extern int lib##_init(); lib##_init();
static int init() {
    INIT(lib1)
    INIT(lib2)
    INIT(lib3)
    // ....
    return 0;
}
static int init_result=init();
```

and link it to project. Just like .def file.

This is simple and you can control order of initialization. And no need of inline variables support.

We do not need to control order of initialization. That is pointless as it is easy to write order neutral initialization code.

Other than that, 'link it to the project, just like .def' creates about the same problem as original .icpp - works great if you have control about build process. .icpp is perfect solution to the problem - the only problem is that it is unusual and no other build system supports it. That is something I am trying to solve. People just expect .cpp and .h (or .h and .lib), any other 'magic' is not welcome.

Mirek

P.S.: I have thought about inline variables and I think they are not much advantage over good old global initialization.
