

---

Subject: Re: Building & using U++ without TheIDE  
Posted by [mirek](#) on Mon, 10 Sep 2007 19:31:23 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

sergei wrote on Mon, 10 September 2007 14:59OK, this set some things straight. So I can think of BLITZ simply as a build time optimizer.

I made some progress. I "solved" the icpp issue by adding this main.cpp file:

```
#include "Core/Core.h"

#include "RichEdit/RichEdit.icpp"
#include "RichText/RichImage.icpp"
#include "Web/Web_init.icpp"
#include "plugin/bmp/BmpReg.icpp"
#include "plugin/gif/gif.icpp"
#include "plugin/jpg/jpgreg.icpp"
#include "plugin/png/pngreg.icpp"
#include "plugin/tif/tifreg.icpp"
#include "PdfDraw/PdfReport.icpp"
#include "CtrlLib/CtrlLib.icpp"
#include "Geom/Ctrl/GeomCtrl_init.icpp"
#include "Core/Core_init.icpp"
#include "CtrlCore/CtrlCore.icpp"
```

I also set it to high priority, to force it to compile first. Some of these icpp don't include Core.h first, which might be a problem, but since Code::Blocks only compiles C and CPP, it isn't.

I am really not sure what priority is, but I bet it does not matter.

Quote:

This yielded an error of INITBLOCK being defined twice (actually, something inside it). Sounds reasonable, same macro used in both GIF and BMP icpp-s.

This is caused by the fact that INITBLOCK synthetises the name of some static variable based on the line number -> by including them into the single file, you are getting them defined twice...

I think the only correct solution is to add empty Init function to .icpp file (InitPluginPng) and call it from main. And rename .icpp to .cpp...

(Alternatively, you can create helper .cpp that includes .icpp and has this function - see technology section).

Quote:

Resulted in the old error of LngEntry\_\_ in t.h being undefined. But that struct is defined in Core/i18n.h, which is included in Core.h. Main.cpp is the first file compiled, so Core/Core.h

should've been included. Weird.

What is the actual error log?

BTW, you do not have to be so much obsessed about order of including - it does not really matter, as long as you include "highest-level" header for your application.

For GUI only, `#include <CtrlLib/CtrlLib.h>` is all you need. If you are doing SQL GUI app, `#include <SqlCtrl/SqlCtrl>` is all required (will include `CtrlLib.h` and `CtrlCore.h` and `Draw.h` and `Core.h` and `Sql.h...`)

Mirek

---