
Subject: Re: Building & using U++ without TheIDE
Posted by [sergei](#) on Sun, 09 Sep 2007 23:48:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks for the quick reply.

I did search the forum previously, all I found was outdated MSVC solution files. Now I've seen your new post about icpp, previously I didn't even understand what they were.

I renamed icpp to cpp, inserted dummy functions, but then I don't think these were the cause of my problems (yet?). Static libraries build every cpp, and include every h included, right? (that would explain locale.cpp with no locale.h).

I might be missing something here. Here's what I'm doing:

- 1) In Code::Blocks (recent nightly build) using MinGW (GCC 3.4.5), I created a static library project.
- 2) Copied everything from uppsrc folder to project folder, added all cpp (including icpp, renamed to cpp so they get compiled), h and rc.
- 3) Defined PLATFORM_WIN32 in project options.
- 4) Ran build all. Got a bunch of warnings and an error.

Warning are mostly signed/unsigned. One was more serious:

Core\Util.h:283: warning: `class Upp::CharFilterTextTest' has virtual functions but non-virtual destructor

Error:

CbGen\CppGen.cpp:: In function `void CallbackGen(Upp::String, Upp::String, int, Upp::String, Upp::String)':
CbGen\CppGen.cpp:27: error: no match for 'operator<<' in 'String(((const char*)"template <class OBJECT, class METHOD") << If(Upp::String, Upp::String)(Upp::operator+(const char*, const Upp::String&)(((const Upp::String&)((const Upp::String*)(&classdef))))))'

That was on:

```
String cl_temp = String("template <class OBJECT, class METHOD") << If(classdef, " " + classdef) << ">";
```

I changed that to (not sure if correct, just wanted to continue):

```
String cl_temp = String("template <class OBJECT, class METHOD");  
cl_temp <<= If(classdef, " " + classdef);  
cl_temp <<= ">";
```

Tried to rebuild. Even more warnings - signed/unsigned, and virtual for PrintPageDraw. It also found that:

```
Item& AddMenu(const String& t, const UPP::Image& m, Callback c) { AddMenu(t, m, c); }
```

Didn't return a value though it should.

And again an error. This time in Core\t.h:

```
INITBLOCK_(COMBINE3(LNG_MODULE, LNG_VERB, LNG_VERA))
{
    static LngEntry__ langset[] = {
        #include TFILE
        { 0, NULL }
    };
    #ifdef LNGMODULE
        AddModule(langset, #LNGMODULE);
    #else
        AddModule(langset);
    #endif
}
```

LngEntry__ and AddModule are undefined.

Included <Core/i18n.h> to this file. Looks like it helped.

Next error: String undefined in i18n, including <Core/String.h> helped.

Next - seems like due to my modifications String got included more than once. And looks like it doesn't have include guards.

OK, I might be able to go on "fixing" the source, but I guess something is wrong if I'm getting all these errors. Or maybe it's due to BLITZ putting all files together and thus solving dependancy problems...

I'd be glad to hear if there is an easy way to solve this. I didn't find a makefile to build U++. I guess taking *.o files from TheIDE could be a solution, but I'd prefer to be able to build a library without TheIDE.