
Subject: Re: Building & using U++ without TheIDE
Posted by [mirek](#) on Wed, 12 Sep 2007 07:20:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

sergei wrote on Tue, 11 September 2007 20:25

I don't know what -D means (doesn't appear in c++ --help), but if these are defines, I need to define: flagGUI, flagGCC, flagDEBUG, flagDEBUG_FULL, flagBLITZ, flagWIN32. Should've done this last time (got overhauled with info due to verbose and missed these defines).

Yes, check...

Quote:

t.h is really weird. I've removed namespace from it, got back to the compiler error.

What is the error? (Important part is from where t.h was included).

Quote:

Writing UPP::LngEntry__ solved the problem, but I still don't understand how INITBLOCK_ is defined there (t.h is included outside UPP namespace). Well, actually I don't understand how INITBLOCK works at all, it looks like some kind of lambda for C++, does it actually execute the code (register) upon program loading, without calling anything?

It creates a special class and single global object of that class; uses constructor of the class to insert initialization code. It is pure macro hackery; the name of class and of object is created based on the line number.

Quote:

I'm getting closer to building U++. It took 43 minutes to get to the first error, which is in RichText (last package to be built). What bothers me is that the build time is so long, and that I get the same warnings over and over again. It's as if Core.h gets re-included for every CPP, and warnings from all its includes repeat (impossible since Core.h has include guards). I might later try to set Core.h to get precompiled and see if it helps.

Yes, it gets included all the time. Anyway, long build times is the thing that theide solves too (with theide, BLITZ and HYDRA - I can completely rebuild UWord, including U++ library, in 24s seconds with mingw and in 14s with MSC...)

Quote:

The RichText error was in Para.cpp. It included an NText.h (non-existent file), and also used Paragraph (non-existent class).

Once again, follow the suggestions. The list of files that really are part of project is displayed in theide and also listed in .upp files inside package directories (also there is the dependency).

(OTOH, thanks, this looks like abandoned file that was forgot in the folder).

Quote:

```
#ifndef USE_MSDOS_MEMMGR /* make sure user got configuration right */  
  You forgot to define USE_MSDOS_MEMMGR in jconfig.h. /* deliberate syntax error */  
#endif
```

That's a funny way to say something's wrong

Defined USE_MSDOS_MEMMGR in jconfig.h, got an error in jmemansi.c:

```
METHODDEF(void)  
read_backing_store (j_common_ptr cinfo, backing_store_ptr info,  
  void FAR * buffer_address,  
  long file_offset, long byte_count)  
{  
  if (fseek(info->temp_file, file_offset, SEEK_SET))  
    ERREXIT(cinfo, JERR_TFILE_SEEK);  
  if (JFREAD(info->temp_file, buffer_address, byte_count)  
    != (size_t) byte_count)  
    ERREXIT(cinfo, JERR_TFILE_READ);  
}
```

Structure has no temp_file member.

Well, these files are not from us, but this is jpeg library. Anyway, I think the problem might be the same as with RichText - redundant file compiled. Please check in theide or in .upp file whether this file is part of package. Same for Local.cpp.

(OTOH, this is really good, as we are now able to remove forgotten files .

Quote:

P.S. I've found: http://en.wikipedia.org/wiki/Single_Compilation_Unit

Pros: only 1 file to compile, probably faster static lib compiling, also icpp issue solution

Cons: INITBLOCK-s will have to be replaced with unique INITBLOCK_(X)-s, possibly other similar changes, every new CPP will have to be added to that file that is compiled (it could be auto-generated, though).

Ah, nice, somebody else noticed the basic principle of BLITZ too. Anyway, the difference is that BLITZ does all things automagically, solving the .icpp problem, creating the "SCU" and managing this so that frequently modified files are compiled separately.

Quote:

P.S.2 Thinking of it now, RichImage.icpp has 2 INITBLOCK-s, how can TheIDE compile these in one source file? It's redefined in Code::Blocks, unless I replace it with INITBLOCK_(X) with different X.

See above. They get different names within single file, which is OK as the global variable is static.

Mirek
