Subject: Re: Building & using U++ without TheIDE
Posted by sergei on Wed, 12 Sep 2007 00:25:07 GMT
View Forum Message <> Reply to Message

OK, I see that I've sounded a bit too ignorant and stubborn. Sorry for that. To clarify things:

1) I'm not afraid of TheIDE, I just don't like it. I just got myself comfortable with Code::Blocks, so I don't want to move to a new IDE to use a new library. I used it a bit, and found it less comfortable than Code::Blocks.
2) I want to build U++ without TheIDE not only for the reason that I don't want to use TheIDE. I want to ensure the library code is not tied to the development environment, and is pretty much standalone. In the hypotetical case of TheIDE ceasing to function, I want to still be able to build whatever I write with U++.
3) I've built UWord with TheIDE, with verbose. It just seemed that it included too few packages, so I guessed others also might be needed in other applications. Currently, apart from packages that are used in UWord, I added GLCtrl and GridCtrl (I hope none of these can cause trouble). I also added more plugins

Command line in TheIDE:

```
c++ -c  -I"C:\upp\examples" -I"C:\upp\uppsrc" -I"C:\upp\mingw\include" -DflagGUI -DflagGCC
-DflagDEBUG -DflagDEBUG_FULL -DflagBLITZ -DflagW
 IN32 -DbmYEAR=2007 -DbmMONTH=9 -DbmDAY=12 -DbmHOUR=0 -DbmMINUTE=30
-DbmSECOND=10 -g2 -static   -fexceptions  -D_DEBUG -O0 -x c++ "C:\up
 p\uppsrc\CtrlLib\ChWin32.cpp" -o "C:/upp/out/CtrlLib/MINGW.Debug_full.Gui\ChWin32.o"
```

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).

wchar problem is now gone (phew). Got some wrong stuff in Locale.cpp (I'm using dev2b from sourceforge, it might be outdated by now - there was and extra */ and missing :: to call Win32 functions). t.h is really weird. I've removed namespace from it, got back to the compiler error. 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?

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.

The RichText error was in Para.cpp. It included an NText.h (non-existant file), and also used Paragraph (non-existant class). On second look, TheIDE used RichText in UWord, but that particular file wasn't used. I wonder how TheIDE knew it wasn't necessary, especially since

Para.h is used. Removed the file, compilation continued.

```
#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.

OK, I made some progress, time to sleep
Accumulating changes to the original sources package (half of it, actually):

In all icpps using INITBLOCK, replace it with INITBLOCK_(BLK_###) - unique ### for every INITBLOCK.
In t.h, replace LngEntry__ with UPP::LngEntry__
In Locale.cpp, add NAMESPACE_UPP and END_UPP_NAMESPACE, define LOG(x) (what is this?), remove extra */, add :: to some Win32 function calls.

And add main.cpp with :

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

#include "Core/Core_init.icpp"
#include "CtrlCore/CtrlCore.icpp"
#include "RichEdit/RichEdit.icpp"
#include "CtrlLib/CtrlLib.icpp"
#include "RichText/RichImage.icpp"
#include "PdfDraw/PdfReport.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"
```

```
void LinkUppInit() {}
```

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).

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.