
Subject: Re: Building & using U++ without TheIDE
Posted by [mirek](#) on Wed, 19 Sep 2007 09:39:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

sergei wrote on Wed, 19 September 2007 05:18 Why need dli? You already have all functions from `#include <windows.h>`.

Yes, but it would not start on Win98 ("missing dll call").

Quote:

UTF-8 -> UTF-16 -> UTF-8 won't happen. `FromFileCharset` returns `String` if it's ASCII/UTF-8 and `WString` if it's UTF-16.

How? Sure, you can make it return something more complex, but I would prefer two functions: one returning `String` and other `WString`.

Quote:

I wanted to compile `UWord` (now in ANSI, GUI Unicode isn't complete yet) to see if `zlib` work (`UWord.iml`), and found an interesting problem in `PdfDraw`:

`ScreenDraw sd;`

That causes a warning of statement is a reference not a function call. + error about `sd` definition. In `Draw/DrawWin32`, `ScreenDraw` is a class, but also:

```
ScreenDraw& ScreenDraw()  
{  
    return Single<ScreenInfoClass>();  
}
```

That's a singleton? Whatever it is, it doesn't work - `ScreenDraw sd;` is recognized as a function name, not class type. Any suggestions how to fix?

Now this is really interesting. It is obvious bug (minor, this is just something forgotten), just tell why in `theide` it compiles without a error, in both `MSC` and all versions of `mingw`?

Maybe different way of `SCU`? (Your `SCU` is bigger than mine?)

Quote:

P.S. Why does U++ use so many global functions? I prefer .Net-style - tree-like organization using namespaces/classes. After all, gathering functions into static classes should be relatively easy, and at the cost of some extra typing you (potentially) resolve naming conflicts, and make stuff easier to find.

Actually, if you look a bit more carefully, then either these are really very globally used names (e.g. `AsString`) - there is a limited number of such cases, or they are carefully bound to parameters so that C++ overloading resolves the conflicts.

Believe or not, function name conflicts are not an issue. (Moreover, everything is still in U++ namespace).

Quote:

Plus that would be an OOP approach

I like multi-paradigm approaches more

Mirek
