
Subject: Re: Building & using U++ without TheIDE
Posted by [mirek](#) on Sat, 15 Sep 2007 18:11:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

sergei wrote on Thu, 13 September 2007 20:21

That would be one pretty serious drawback IMHO. I'm using Russian and Hebrew, and I've used so many apps with bad unicode support that I really want to avoid making another one. Who uses Win98 nowadays (and there's unicode layer for these)

Well, I agree. Believe me, I would be more than happy to be finally able to cancel Win98 support and make everything UNICODE.

But even year ago there were people asking for *TheIDE* to run 98.... I can understand that people want apps to be Win98 compatible, but developing on it?...

Quote:

I've tried entering non-English chars in TheIDE source editor, it didn't let me.

Actually, this is different reason. See FAQ. (In short, theide keeps track of encoding of file used and if keystroke does not match encoding, it is not inserted-> switch file encoding to UTF-8.)

Quote:

Could this be fixed by some creative editing of Util.h?

Yes. In fact, the code is there for PocketPC version.

Quote:

And what about window title, and controls?

Window title already works both in Win98 and XP UNICODE (it is quite easy to provide both way).

All widgets are UNICODE capable, there really is no trouble.

Quote:

I mean, any chance to bring this to the TCHAR style, so the program can be compiled in ANSI, and in Unicode (supporting several languages simultaneously)?

No. Forget about TCHAR style. But it can be easily done just right:)

Which makes me think, this can even be done right while still supporting Win98, it is just much more work (need to test platform and dynamically link A or W version). .dli would help.

Quote:

This should be a matter of some defines and a function to convert UTF8 to Win32 unicode.

Well, in fact, these functions are already there and they work. In fact, U++ has no problem to handle chinesse filenames (because chinesse windows are using multibyte 8bit encoding of characters, something similar to UTF-8, and it is trivial to convert this to UTF-8 and back...).

Quote:

Or are there other parts of U++ that rely on a specific String/WString encoding?

Well, once you have String<->WString conversion and default encoding (SetDefaultCharset), the rest of code is pretty encoding ambivalent. The only real problem is the one you have encountered - the outside world interface. This is basically filenames and fonts; fonts are working UNICODE in Win98 out of the box, so really the remaining problem is filenames.

But once again, current pragmatic solution is just a result of fact that 2 years ago we had customers that required Win98 support...

Mirek

PS.: BTW, Hebrew... Is not it RTL? (Another not yet resolved problem, this time because we just do not know how RTL is supposed to work...).