Subject: Re: Building & using U++ without TheIDE
Posted by sergei on Sat, 15 Sep 2007 23:43:10 GMT
View Forum Message <> Reply to Message

Yes, UTF-8 can have BOM. Actually it has, and all programs I've used write/recognize it, with the unfortunate exception of GCC (it takes UTF-8 without BOM - such files aren't always shown correctly in text editors).

There is no need to modify existing UTF-8 handling since I doubt BOM is used in strings (it's essential for files). Yet my UTFBOM might be useful since LoadFile/SaveFile aren't encoding-aware, and files that are saved in UTF-16 format (that's what I commonly use for non-English text) would be loaded as ANSI/UTF-8. Or maybe just add autodecode to LoadFile and optional encoding params to SaveFile.

Multiple encodings in one file? Any examples? I don't think any text editor would recognize such a file.

I tried your suggestion about WinCE. Is PocketPC Unicode-only (that would be awesome if MS actually made such good decision)? First of all, cAlternateFileName, from Path.h, is never defined. As such, GetMSDOSName can't be defined either, and can't be used in FileSystemInfo::Find. That's likely a bug - I just commented everything out, or is there a way to implement GetMSDOSName?
Well, it didn't really work. The same "craziness" returned. I replaced:

```
#ifdef PLATFORM_WINCE
```

with:

```
#if defined(PLATFORM_WINCE) || defined(UNICODE)
```

and define UNICODE in main.cpp:

```
#define UNICODE

#include <Core/Core.h>

using namespace Upp;

...
```

Still, in the debugger it insisted to go into the #else. Rebuild all didn't help. My guess is that the solution could've worked (WString should cast into WCHAR*).

BTW, I've found some mistakes in UTFBOM and fixed that. Now I seem to be able to load/save UTF-8/UTF-16 LE/BE with/without BOM. I'm attaching the updated code (class + demo that I tried to use to read unicode-filenamed file). UTF-32 is kinda rare, though I might add it too for sake of completeness. However, that would require a String/WString that can work with embedded nulls - can they?

As for RTL, if I have time, I'll read Unicode specs on how LTR and RTL is mixed in same paragraph. The issue is quite interesting, but I'm afraid a standards-conforming solution might end up being an unusual one since many programs tend to ignore the existance of RTL languages.

P.S. is there a portable way to get a key from console (only key, without Enter)? Like _getch()?

P.S.2 String (UTF-8) seems to be the native U++ representation of text. Is WString used / recommended to be used for anything
besides Unicode OS API calls?

File Attachments

1) UniTest.cpp, downloaded 642 times