Subject: Re: Building & using U++ without TheIDE
Posted by sergei on Wed, 19 Sep 2007 08:12:23 GMT
View Forum Message <> Reply to Message

I see now... I knew there should've been a good reason for a second set of charset functions. OK, I'll reverse the changes and see if that works.

About registry:

```
bool SetWinRegString(const String& string, const char *value, const char *path, HKEY base_key)
{
 HKEY key = 0;
 if(RegCreateKeyEx(base_key, ToSystemCharset(path), 0, NULL,
REG_OPTION_NON_VOLATILE,
  KEY_ALL_ACCESS, NULL, &key, NULL) != ERROR_SUCCESS)
  return false;
#ifdef PLATFORM_UNICODE
 WString wstring = string.ToWString(); wstring.Cat(0, 1);
 bool ok = (RegSetValueEx(key, ToSystemCharset(value), 0, REG_SZ, (const byte*)(const
wchar*)wstring, (wstring.GetLength() + 1)*2) == ERROR_SUCCESS);
#else
 bool ok = (RegSetValueEx(key, value, 0, REG_SZ, (const byte*)(const char*)string,
string.GetLength() + 1) == ERROR_SUCCESS);
#endif
 RegCloseKey(key);
 return ok;
}
```

The #else part is what was prevoiusly the code (I added the casts, though). Linking that to W version wouldn't be possible - defining UNICODE without #ifdef would cause an error of cast of char* to WCHAR*.

I found that I didn't modify everything, since console apps don't include much Then I found that already exists a macro L_(). Thus using TCHAR instead of char/wchar, + L_() and To/FromSystemCharset might indeed remove these #ifdefs. But that would be later, first to ensure everything works.

I'm not sure how you want to use dynamic dll loading. Change all #ifdefs into if/elses, and explicitely call W and A versions, to enable runtime switching between ANSI/Unicode?

UTFBOM: Skip BOM of UTF-8 / UTF-16 LE / UTF-16 BE files (not only UTF-8), and read ASCII/UTF-8 (if there's no BOM, it's considered ASCII) into String, UTF-16 LE/BE into WString. Convert UTF-8 String into UTF-8 / UTF-16 LE/BE with/without BOM. I guess it should be:

```
int FromFileCharset(const String& s, String* os, WString* ows);
```

String ToFileCharset(const String& s, int bytes, bool BOM = true, bool LE = true);
String ToFileCharset(const WString& s, int bytes, bool BOM = true, bool LE = true);

(maybe should add ASCII -> UTF-8 conversion if there's no BOM, since chars > 127 could cause invalid UTF-8, being just system-charset chars)