

Update:

I'm done implementing unicode. I don't exactly like the way I did it, but it works. I prefer some global strings manager like the TSTR I suggested, but since Mirek said he doesn't want TCHARs I've updated all API calls manually. I added PLATFORM_UNICODE define, and replaced many `#ifdef PLATFORM_WINCE` with it. Conversion mostly implemented through `ToSystemCharset` and `FromSystemCharset`.

In the process I've also found and eliminated a "security vulnerability" (that's what memory bugs are called nowadays) in `Log.h`:

```
printf(h, "%s %02d.%02d.%04d %02d:%02d:%02d, user: %s\n",  
      (const char*)FromSystemCharset(exe),  
      t.day, t.month, t.year, t.hour, t.minute, t.second, (const char*)FromSystemCharset(user));
```

I added the `(const char*)`, otherwise I got segmentation fault in debug. Probably without the explicit cast `printf` thought that `String` is a char array.

I received another segmentation fault earlier (but maybe it was this one...), and that one was solved by replacing all `To/FromSysChrSet` with `To/FromSystemCharset`. Not a big deal IMHO, they were all used in Win32-specific code anyway...

I've tested unicode filenames - it did open a multilingually-named file and read its content successfully. I've also tested registry - successfully wrote that filename to a `REG_SZ` key, it's fine. Didn't try to create unicode-named keys (don't wanna kill my windows).

Next (final) step - `zlib/lay/iml/images` testing. If that goes well, I'll upload everything.

P.S. I'm working from `Code::Blocks` when editing U++ source, rebuild for console project is half a minute. So I don't want a static lib for debug Though GDB / Code completion don't work that well - GDB reports most stuff in U++ source as incomplete type (so I have to `Cout` whatever I want to see), and Code completion usually can't find definition of things (but it does know function prototypes).

P.S.2 since you know about the MSVC bug, did anyone report to MS? There might be a chance that they fix it...

Edit: I think `UTFBOM` class I posted above, or something else implementing that functionality, should be added to some place in U++. That way unicode support will be complete - unicode filenames + unicode text.