

---

Subject: Re: WinAPI UNICODE question  
Posted by [cbpporter](#) on Sun, 16 Nov 2008 10:33:20 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

luzr wrote on Sun, 16 November 2008 11:19I wonder a little bit, WHY?

Well there are multiple possible answers.

First, it is problem of principle. U++ is my platform of choice for at least the next 2 years, and I want it to be THE platform, not just a platform combined with another platform (WinAPI) and another platform (various STD libs). So following this principle I don't want to wind up with a lot of useless stuff (which I will never ever use and am not able to use if I want cross-platform behavior) just for including a .h from U++.

Second, a practical consideration. Since U++ includes a lot of unnecessary and most importantly platform dependent stuff, and doesn't provide all capabilities under all platforms, and since I use Windows as a primary development platform (and periodically compile under Linux), I end d up using unintentionally a lot of crap that simply does not compile under Linux. Now I have the nice situation of Windows version being OK and close to beta, and Linux one crippled for an unpredictably long time. It is OK to use it if I make a conscious choice (i.e. include a system dependent header), it's not OK if U++ allow me to compile that stuff by default.

Third, it is a principle of giving the precompiler less to work. Under MSC8, just for including Core.h I get about 240.000 lines of code that are precompiled. Time that by the number of packages for BLITZ builds and the number of compilation units for non BLITZ, and you do get a hefty sum. Sure, the by removing all the stuff and winding up with an about 25.000 line precompilation I won't be saving any lives, not even power bill (well maybe a completely unimportant sum), but still...

So in short:

```
#include <Core/Core.h>
CONSOLE_APP_MAIN
{
    HWND hwnd;
}
```

Must not compile under Win32. The same goes if I replace HWND with pthread under Linux in a MT build. Preferably not even strcpy would compile without including <string.h>, but standard lib is stable enough to allow it (even though I can easily break cross-platform compatibility by calling some public functions that don't respect the standards, but I'm not going to clean up standard C library also).

Quote:

To answer your question, we are using "3." - call 'W' version if running >=WinNT, 'A' version otherwise.

Mirek

OK, I think I understand now. And where version without "A" or "W" are called, "A" versions are selected.

---