Subject: Re: WinAPI UNICODE question
Posted by cbpporter on Mon, 24 Nov 2008 00:04:57 GMT
View Forum Message <> Reply to Message

bytefield wrote on Sun, 23 November 2008 19:46Is your work related to "cleaning up" the Upp from unwanted dependencies, functions conflict, "old" not so good design, etc. or it is a derivation of Upp? If not, when your modifications will be available in Upp sources?
It is related to exposing as little as possible as possible from the target platform API at an interface level. It only allows platform specific code (right now just WinAPI, but later can be done for Linux also, I think even easier) to not compile or be useful for client of U++ API. Also, I removed some unwanted external headers. All WinAPI functions are hidden, and only some types are available, but these types are available independent of platform. You can use a HWND under Linux, and both under Linux and Windows it will still be completely useless for you. It will just be a pointer type, without any way to use it in a platform dependent way (unless you go out of your way to cast it to something evil, but like C++ in general, my approach protects you from accident, not from intention). So this raises the question: how to deal with the parts of U++ that need Windows API. Well I made Core.h platform independent and added a CorePS.h that must be included in platform dependent compilation units. The number of compilation units that need platform specific code is very small, thus the 15% compilation time decrease. Except the new header that is included, and some small API differences that are applied here and there, the rest of the U++ code base remains unmodified. I'm not expecting this to be used in the official release, but to get it to a shape where 99% of it is identical to official U++, the rest of 1% is handled by auto merge, and the generated binary from compilation is the same.