Subject: Re: NTL and C4251
Posted by mirek on Mon, 24 Apr 2006 14:27:02 GMT
View Forum Message <> Reply to Message

jmansion wrote on Mon, 24 April 2006 10:04>TheIDE has specific hacks to compile process that makes those
>ugly declspec issues for importing/exporting stuff unnecessary
>(EXPORTS are automatically generated, just like in linux/.so -
>no explicit export/import declarations are necessary).

Oh - I'd hardly call the situation on Linux ideal, hence the work the gcc team has been doing to make control of visibility much easier and the performance issues seen in OOo and KDE.

I agree that performance sucks with dynamic loading of .so. However, I think this is rather the problem of that specific implementation than the problem of visibility only.

In any case, standalony binary (like one produced by U++) beats them all

Quote:
>Anyway, to tell the truth, U++ is designed to provide
>standalone executables, no .dlls at all.

Ouch. That's really a big disadvantage - it cuts out the use of SWIG to help automate subsystem testing etc or to provide services to JNI, P/Invoke or even to dynamically load extensions. I know I've typically released monolithic in-house binaries built static, but I'd not want to be forced to do this.

Well, the real meaning of standalone here is that you can ship U++ binary without any aditional .dll and it works on everything since Win95 or on most current Linux 386 distros. Anyway, on Linux, the build in fact is shared (against a limited number of system .so - namely glibc, stdc++, xlib and xft).

You can however use 3rd party .dlls, you can build .dlls (even OLE/COM components), and you can even build in "SO" mode where all library subsystems are build as dll.

Quote:
And that's probably answered whether I could move to U++ as my main development env.

Well, as always, you cannot satisfy everybody.... The mission goals are low development / maintainance costs. DLL hell does not contribute to either

Mirek