Subject: Re: Some small issues with TheIDE and DLLs
Posted by mirek on Wed, 01 Jul 2009 08:58:44 GMT

View Forum Message <> Reply to Message

cbpporter wrote on Tue, 30 June 2009 22:58I need to generate a DLL from TheIDE which exports some functions with implementations that use U++ classes.

First I tried setting the "All shared" option. This was a long shot and I was sure it wouldn't work, but the build process crashed. It shouldn't crash.

With MSC?

Quote:
After some searching on the site I found that you are supposed to set the "DLL" configuration flag. This seems to have worked. I get a rather large DLL (probably the linker doesn't know what to exclude and includes all), an ".exp" file and a ".lib" file. Everything seems to work. Except when I hit run, I get a nice error message that the DLL can't be run (which is true of course) and a nice crash. It shouldn't crash.

Definitely. Will look into it ASAP.

Quote:
But when building with MINGW I don't get the ".lib". This is needed if I don't do manual loading of the DLL. (Also the DLL is more than twice as big when compared to the MSC version, but I guess this is to be expected).

Not sure we even want to support mingw there....

Quote:
I have a few questions:
1. Should I compile with USEMALLOC option. Will U++'s malloc implementation cause problems if I free in the client application something allocated by the host? Will it be okay if all allocation and freeing is done by calls to the DLL?

Depends.If all allocations / freeing is done by calls to the DLL, then AFAIK it should be ok.

If not, then AFAIK you can have problems with USEMALLOC as well...

Generally, DLL interfaces are designed so that either memory is handled by Win32 API, or handled by calls to DLL (no explicit frees).

Quote:
2. Will there be issues with "icpp"s and other initialization tricks that U++ uses?

I believe not, or not really big, as long as compiler/linker supports global constructors/destructors.

Quote:
3. Can I use the the "DLL" flag under Linux or must it be changed to "SO". Same issue, I don't want a .so for every package, I want one big one.

It is still DLL (although it produces SO).

Quote:
4. Is there some macro provided by U++ as a platform independent equivalent of __declspec(dllexport) for writing DLLs?

No.

Quote:
5. What happens to global variables. What if my functions alter such variables. Will there be multiple such instances for different clients or will they override each other. Can this cause problems with all the global variables U++ uses internally? And is this behavior consistent across platforms?

In POSIX, everything behaves exactly as normal.

In Win32, global variables exist, but are not accessible from outside of DLL.

One special consideration is TLS variables in Win32 - they are mutually exclusive with run-time loading of .dll (if that is the right term, I mean loading .dll explicitly, not by linker).

Mirek