Subject: Re: Interesting struggle with "Moveable<T>" usage in GCC
Posted by mr_ped on Wed, 30 Jul 2008 10:14:47 GMT
View Forum Message <> Reply to Message

Another funny thing, if you mess up the optional deep copy functions (namely DeepCopyConstruct to call the implicit copy constructor instead of your T(T, int) deep copy constructor) for your class defined to be moveable trough that NTL_MOVEABLE macro:
you can end with Vector source which works with MSC8 (the implicit copy constructor is not called at all), but does break under GCC. (with the wrong pick behavior assert)

I can't provide example because the source is too complex and basically it was my bug in DeepCopyConstruct, so it's not problem of UPP.

It just made me to scratch my head a bit, that two compilers do use quite different way to construct the final Vector related code for my class, MSC avoiding my bugged functions completely.

Of course using "class A : MoveableAndDeepCopyOption<A> {};" fixed that and made my code cleaner (and right now I didn't use direct initialization for that A class anymore), so if you are new to UPP+pick behavior (like me), try to avoid making things by hand, and use rather those Moveable<> and similar things whenever possible, it makes code clean and easy to read, and will prevent you from doing silly mistakes (which may go undetected with one compiler and show just later on another one).