Subject: Re: what technique does U++ use to reduce code bloat from too many template instantiations?

Posted by thierry on Sun, 10 Sep 2006 16:32:22 GMT

View Forum Message <> Reply to Message

Could it possible that theIDE can help still a little with encouraging the following practice in case of not so clever linker (relying only on weakreferences) that anyway can't avoid the inline code bloat:

T.h : no inline code for templates
(well just recommand and do it)

T.tpp: template function implementation

T_mytype.cpp: one for each explicit instanciation
#include "T.tpp"
template T<mytype>;

This is really annoying for a programmer to create one cpp per instanciation, but this is where an ide can help a lot to save developer to follow this guideline with generating the T_mytype.cpp for him, with maybe having a repository of all instanciations at least by packages... Then the linker shall be only smart enough not to link twice the same code.

This would not only save code bloat, but also compilation time, so far template would be compile only once, and changes in .tpp won't affect other modules