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

Posted by [mirek](#) on Mon, 12 Dec 2005 20:44:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

fudadmin wrote on Mon, 12 December 2005 15:02 what technique does U++ use to reduce code bloat from too many template instantiations?

A good question. Actually, some of that is reduced by smart linkers (they are able to merge the same function).

Anyway, the really great reduction is result of U++ containers, esp. two things: Array container has really low footprint (it needs about 200 bytes pre instantiation) and Map containers are composites of Index and Array or Vector, with quite a lot Index hashing code implemented in non-template .cpp.

To explain composition issue: `VectorMap<String, int>` and `VectorMap<String, String>` are actually compositions of `Index<String>` and either `Vector<int>` and `Vector<String>` and "composition" code is really minimal. Compare with STL that generates totally different code for `map<string, int>`, `map<string, string>`, `vector<string>` and `vector<int>`. With more types involved, advantage on U++ side increases....
